IMPROVED POWER SYSTEM REALIZATION AND INTEGRATION

by

Ali Ridha Al-Roomi

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
April 2020

This dissertation is lovingly dedicated to my family for their support, encouragement, understanding, and constant love.

I would also like to dedicate this dissertation to the memory of Prof. Mohamed E. El-Hawary. Although he was my Ph.D. supervisor for almost three years and eight months of encouragement and inspiration, he was unable to see my graduation. This is for him.

Table of Contents

List of Tables

List of Figures

Abstract

The main goal of this research is to pave a new path to solve electric power system problems from a realistic perspective. The problems covered in this dissertation are power transmission lines, power flow (PF) analysis, economic load dispatch (ELD), power and energy forecasting, optimal relay coordination (ORC), fault location, and state estimation. In this journey, we try to push the borders by digging everywhere to find other alternatives to precisely solve many existing real-world power system problems. This study considers many hidden facts and phenomena.

To ensure accomplishing this ambitious task with some sorts of intelligence, advanced tools are involved; including traditional and meta-heuristic optimization algorithms and machine learning (ML) computing systems. Some superior hybrid optimization algorithms and ML computing systems are developed.

The mission starts from the Telegrapher's equations where the distributed- and lumped-parameter transmission line models are built on. The realization phase is done by considering the effects of surrounding weather, system frequency, load current, and cable design/status/age. Some innovative techniques are proposed to solve the inherent weaknesses in explaining the deviation in distributed series and shunt parameters of lines with sag. This realization is applied to enhance the solutions of PF, ELD, short-circuit analysis, power system stability and ORC problems.

To avoid tedious and highly time-consuming computational methods, a new set of optimization-free/modeling-free techniques are designed to solve ELD problems.

Through the realization and integration phases, many new innovative ideas are presented. Also, because many problems heavily depend on ML tools, so a new computing system is designed to achieve the accuracy and precision criteria without losing the explainability and interpretability criteria. That is, compromising between the strengths of classical linear regression (LR) and nonlinear regression (NLR) analysis and modern artificial neural networks (ANNs) and support vector machines (SVMs).

To judge the performance of each technique, many theoretical and real-world test systems and datasets are used with considering different scenarios and conditions.

List of Abbreviations and Symbols Used

A	Conductor cross-sectional area
$A_{\mathbf{dead}}$	Eliminated area, above sag line, of the line wings problem
$A_{\mathbf{total}_0}$	Nominal area, before sag, of the line wings problem
$A_{\mathbf{total}_s}$	Effective area, after sag, of the line wings problem
B	Line lumped shunt charging susceptance
B_m	Transformer magnetizing susceptance
C	Line capacitance
$CTI_i^{y_i}$	Coordination time interval of the ith relay manufactured with the y_ith technology
CTI_i	Coordination time interval of the ith primary relay
CTR_i	Current transformer ratio of the ith protective relay
$C_i^a\left(P_i\right)$	Operating cost of the ith unit to supply P_i
$C_i^r\left(Q_i\right)$	Operating cost of the ith unit to supply Q_i
$C_i^*\left(P_i^*\right)$	Operating cost of the ith multi-fuel-based unit when it is operated by the best fuel mixture
$C_{i,k_i}^*\left(P_{i,k_i}^*\right)$	Operating cost of the k_ith unit located in the ith power station when the best fuel mixture is used
C_0	Conductor nominal capacitance
C_T	Total fuel cost to operate n units
C_p	Mallows' factor
C_r	crossover rate of DE
$C_{\mathbf{AN}}$	Capacitance of different overhead line systems
$C_{i,k_i}\left(P_{i,k_i}\right)$	Operating cost of the k_ith unit located in the ith power station
D	Distance between two single phase wires
D	Generator damping
$D_{i,k_i,h}^*$	Penalty function of the hth planned maintenance applied to the k_ith unit of the ith power station when the best fuel is used

$D_{\mathrm{xy}'}$	Distance between the overhead conductor x and the imaginary conductor y$'$, where x $\in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and y$'$ $\in \{\mathrm{A}', \mathrm{B}', \mathrm{C}'\}$
$D_{\mathbf{xy}}$	Distance between phase x and phase y, where x and y $\in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$
E	Energy
E_G	Generator internal voltage
E_∞	Infinite bus voltage
F	step-size parameter of DE
$F_i\,(P_i)$	Billing of the ith unit as a function of the ith active power
F_x	Fault at the xth location
F_{SVR}	SVR kernel hyperparameter, which represents one element of Υ
G	Line conductance
G	Number of iterations or generations
G_c	Transformer magnetizing conductance
G_{RSA}	Number of iterations assigned to RSA
G_{SVR}	Maximum number of iterations that the SVR algorithm can use
$G_{\mathbf{AN}}$	Conductance of different overhead line systems
$G_{\mathbf{L\text{-}L}}$	Conductance of a two-wire system operated under low frequency
H	Generator inertia constant
H	Temperature difference vector of branches
H_0	Null Hypothesis
H_1	Alternative Hypothesis
$I_L^{\max}$	Maximum rated current
$I_{OL}^{\max}$	Maximum overload current
$I_{f,i}^{\min}$	Minimum fault current that must be detected by the ith relay
I_G	Current delivered to an infinite bus

I_R	Receiving-end current
I_S	Sending-end current
I_{R_i,F_x}	Short-circuit current seen by the ith relay for the xth fault location
$I_{R_i}(x)$	Short-circuit current seen by the ith relay, which is expressed as a function of the xth fault location
$I_{i,k}$	Fault current occurred at the kth location and seen by the ith primary relay
I_p	Peak current
I_{sc}	Short-circuit current
$J_0(\upsilon_r)$	The zero-order Bessel function of the first kind
$J_1(\upsilon_r)$	The first-order Bessel function of the first kind
L	Line inductance
L_0	Conductor nominal inductance
$L_{\mathbf{ext}_0}$	Conductor nominal external inductance
$L_{\mathbf{ext}}$	External inductance
$L_{\mathbf{int}_0}$	Conductor nominal internal inductance
L_{ij}	Line located between the ith and jth busbars
N	Population size
$P(X)$	Penalty function
PSM_{ix}	Plug setting multiplier of the ith primary protective set where the subscript x stands for either the main-1 or main-2 relay
$P_i^{'}$	Transformed active power of the ith unit
$P^*_{\mathbf{PS}_x}$	Net active power generated by the xth power station when its units are operated by the best fuel mixtures
$P^*_{i,\check{k}}$	Active power generated by the slack unit of the ith power station when the best fuel mixture is used
$P^*_{i,\text{aux}}$	Total power consumed by the auxiliary equipment of the ith power station when all the units are operated by the best fuel mixtures

P^{*}_{i,k_i}	Active power generated by the k_ith unit of the ith power station when the best fuel mixture is used
$P_i^{\mathbf{net}}$	The net active power injected to the ith busbar
P_D	Total active load demand
P_L	Total active power loss in the network
P_R	Receiving-end active power
P_S	Sending-end active power
P_T	Total power generated by n units
P_e	Full-load electric active power output of a combined-cycle power plant
P_i	Active power of the ith unit
P_x	The xth power demand
$P_{D,i}$	The load active power consumed from the ith busbar
$P_{G,i}$	The source active power injected to the ith busbar
P_G	Electric active power delivered to an infinite bus
$P_{L,ij}$	Active power loss in the branch between the ith and jth busbars
P_{COST}	Array of daily total fuel cost of all power stations
P_{GT_Λ}	Active power generated by the Λth gas turbine
P_{NET}	Array of daily net active power produced by all power stations
P_{ST_j}	Active power generated by the jth steam turbine
P_{buses}	Loads connected to all busbars
$P_{\mathbf{in}}$	Total active power injected to a power network
$P_{\mathbf{out}}$	Total active power consumed by power loads
$P_{i,\mathrm{aux}}$	Total power consumed by the auxiliary equipment of the ith power station
P_{i,k_i}	Active power generated by the k_ith unit of the ith power station
Pr	Species count probability
$Pr_{\max}$	The largest element of Pr

$Q_L^{\breve{T},f}$	Reactive power losses at $\breve{T}$ and f
$Q_i^{\mathbf{net}}$	The net reactive power injected to the ith busbar
Q_L	Total reactive power loss in the network
Q_R	Receiving-end reactive power
Q_S	Sending-end reactive power
$Q_{D,i}$	The load reactive power consumed from the ith busbar
$Q_{G,i}$	The source reactive power injected to the ith busbar
$Q_{L,ij}$	Reactive power loss in the branch between the ith and jth busbars
$Q_{\mathbf{in}}$	Total reactive power injected to a power network
$Q_{\mathbf{out}}$	Total reactive power consumed by power loads
Q_i	Reactive power of the ith unit
$R^{\mathbf{bc}}$	Backup relay
$R_{\mathbf{tr}}^{\mathbf{eq}}$	Transformer series resistance
$R^{\mathbf{pr}}$	Primary relay
R_a	Generator stator resistance
R_m	User-defined mutation range
R_{DC}	DC resistance of a cylinderical conductor
$R_{\mathbf{cond}}$	Wound conductor resistance
$R_{\mathbf{layer}_i}$	Total resistance of the ith layer having multiple conductors
S_L	Total complex power loss in the network
S_i	Total number of units operated in the ith power station
$S_{\max}$	Maximum number of species on an island
$T_{R_i}^{F_x}$	Operating time of the ith relay for the xth fault location
T_r	Number of trials or runs
T_{CB}	Circuit breaker time delay
T_{OS}	Over-shoot time, which is also called the over-travel delay and the coasting time

$T_{R_i}\left(I_{R_i}\left(x\right)\right)$	Operating time of the ith relay for the xth fault location, which is expressed as a function of $I_{R_i}\left(x\right)$
$T_{R_i}\left(x\right)$	Operating time of the ith relay, which is expressed as a function of the xth fault location
T_{SM}	Safety margin
$T_{\mathbf{clear}}$	Clearing time
$T_{\mathbf{cr}}$	Critical time
U	Potential
$U^*_{i,k_i,\mathrm{EOH}}\left(P^*_{i,k_i,\mathrm{EOH}}\right)$	The degradation rate of the k_ith unit located in the ith power station based on the equivalent operating hours accumulated for that unit when the best fuel mixture is used
$U_{ij,k}$	Actual complex power magnitude flowing in the i-jth branch, where the subscript k denotes the observation number
V^{rated}	Bus rated voltage
V_R	Receiving-end voltage
V_S	Sending-end voltage
V_t	Generator terminal voltage
V_p	Peak voltage
W	Number of towers
X	Set of design variables, which is also called the design vector
X'_d	Generator direct axis transient reactance
$X^{\mathbf{eq}}_{\mathbf{tr}}$	Transformer series reactance
X_C	Conductor capacitive reactance
X_{C_0}	Conductor nominal capacitive reactance
X_{L_0}	Conductor nominal inductive reactance
$X_{\mathbf{ext}}$	External inductive reactance
$X_{\mathbf{tot}}$	Total reactance between generator and infinite bus
Y	Feasible individual
Y	Line admittance

Y_0	Nominal shunt admittance of the M-model
Y_C	Center shunt admittance of the M-model
Y_S	Left and right shunt admittance of the M-model
Y_c	Characteristic admittance
$Y_{\mathbf{BUS}}$	Nodal admittance matrix
Z	Line impedance
Z_0	Nominal series impedance of the M-model
Z_L	Left series impedance of the M-model
Z_R	Right series impedance of the M-model
Z_V	Left or right series impedance of the M-model under leveled-spans scenario
Z_c	Characteristic impedance
$Z_{\mathbf{ext}}$	External impedance
$Z_{\mathbf{int}}$	Internal impedance
$[G]_m$	Set of m inequality constraints
$[H]_l$	Set of l equality constraints
$\Delta\breve{T}$	Temperature change
$\Delta\mathfrak{L}$	Length change of an overhead line due to $\Delta\breve{T}$
Δx	Length section of long-length transmission lines
Γ-circuit	A gamma-shaped circuit to represent the lumped series and shunt parameters of transmission lines
$\Phi(x,t)$	External source to make wave PDE inhomogeneous
$\Phi_{i,j}$	Penalty function of the jth emission rate applied to the ith power station
Υ	Randomly generated SVR hyperparameters
α	Conductor resistance temperature coefficient
$\alpha(\breve{T})$	Cooling function of SA
$\alpha_{\mathbf{aluminum}}$	Resistance temperature coefficient of an aluminum conductor
$\beth$-circuit	An opposite-gamma-shaped circuit to represent the lumped series and shunt parameters of transmission lines

$\bar{C}_{i,J}\left(P_{i,J}\right)$	Operating cost of the ith multi-fuel-based unit when the Jth fuel type is used without considering the valve-point loading effect
$\bar{I}_p$	Complex peak current
$\bar{V}_p$	Complex peak voltage
$\bar{\alpha}$	Attenuation constant
$\bar{\beta}$	Phase constant
$\bar{x}_{\textbf{bus i}}$	Distance estimated from the ith busbar to the fault location F_x
$\beta_{\textbf{ext}}$	Conductor external inductance temperature coefficient
$\beta_{\textbf{int}}$	Conductor internal inductance temperature coefficient
$\breve{E}$	Energy of a system maintained in a thermal equilibrium
$\breve{P}$	Boltzmann's energy probabilistic distribution
$\breve{T}$	Temperature
$\breve{T}_0$	Initial temperature
$\breve{T}_0$	Reference temperature
$\breve{T}_F$	Temperature constant
$\breve{T}_a$	Ambient temperature
$\breve{T}_k$	Temperature at the kth zone
$\breve{T}_s$	Surface temperature
$\breve{T}_\infty$	Average surrounding temperature
$\breve{T}_\omega$	Wind chill
$\breve{T}_{\textbf{abs}}$	Absolute temperature
χ_j	Number of iterations assigned to the jth stage of RSA
$\circledcirc_{\substack{j \\ j<v}}$	The jth external universal arithmetic operator assigned to the jth and $(j+1)$th blocks of UFO; it could be $+$, $-$, $\times$, $\div$, etc
$\lvert S_R \rvert$	Receiving-end apparent power
δ	Bus voltage angle

δ_G	Generator internal voltage angle
$\dot{V}(\varpi)$	Gas turbine fuel volumetric flow-rate, which is expressed as a function of the prime-mover angular speed
$\dot{V}_i^{M_i}$	Total volumetric flow-rate of the M_i fuels assigned to the ith unit
$\dot{V}_{i,\check{k}}$	Total volumetric flow-rate consumed of multiple fuels consumed by the slack unit of the ith power station
$\dot{V}_{i,k_i}$	Total volumetric flow-rate of multiple fuels consumed by the k_ith unit of the ith power station
$\dot{v}_{i,W}$	Volumetric flow-rate of the Wth fuel assigned to the ith unit
$\dot{v}_{i,k_i,J}$	Volumetric flow-rate of the Jth fuel assigned to the k_ith unit of the ith power station
$\downarrow$	Decrease
$\emptyset$	Empty set
η	Line efficiency
η	Thermal expansion coefficient
$\eta_{\text{overall}}^{\text{lossless}}$	Overall efficiency of a heating system without considering losses
$\eta_{\text{overall}}^{\text{lossy}}$	Overall efficiency of a heating system with considering losses
$\eta_{c\to m}$	Efficiency of a chemical to mechanical energy transformation
$\eta_{c\to t}$	Efficiency of a chemical to thermal energy transformation
$\eta_{e\to t}$	Efficiency of an electrical to thermal energy transformation
$\eta_{m\to e}$	Efficiency of a mechanical to electrical energy transformation
$\forall$	For all
γ	Propagation constant
γ_i	Propagation constant with respect to current

γ_v	Propagation constant with respect to voltage
$\hat{B}$	B-coefficients
$\hat{C}$	C-coefficients
$\hat{C}_i(P_i^{'})$	Estimated operating cost of the ith unit as a function of the ith transformed active power
$\hat{C}_i(P_i)$	Estimated operating cost of the ith unit as a function of the ith active power
$\hat{P}_i\left(C_i\right)$	Estimated active power of the ith unit as a function of the ith operating cost
$\hat{T}$	Species turnover rate at equilibrium
$\hat{Y}$	Equivalent medium line lumped shunt admittance obtained from the ABCD parameters of a lone line
$\hat{Z}$	Equivalent medium line lumped series impedance obtained from the ABCD parameters of a lone line
$\hat{g}_q^{\max}(X)$	The biggest functional constraint
$\hat{g}_q(X)$	The qth inequality or relaxed equality constraint
$\hat{u}_{ij,k}$	Estimated complex power magnitude flowing in the i-jth branch, where the subscript k denotes the observation number
$\hat{y}$	Regressed response
$\hbar$	Number of hidden layers
∞**-model**	A transmission line model designed based on an infinity-shaped circuit of lumped series and shunt parameters
κ	Conductor shunt capacitance temperature coefficient
λ	Conductor shunt resistance temperature coefficient
λ	Immigration rate
$\lvert V\rvert$	Bus voltage magnitude
$\breve{T}_{\mathrm{dp}}$	Dew point temperature
$\hat{S}$	Equilibrium number of species
$\mathcal{J}$	Jacobian matrix
$\mathcal{F}$	Feasible

$\mathcal{N}$	Objective function used to find the optimum polynomial order of regression analysis
$\mathcal{O}$	Number of observations
$\mathcal{P}_G$	Gas pressure
$\mathcal{P}_a$	Atmospheric pressure
$\mathcal{P}_s$	Station pressure
$\mathcal{P}_v$	Exhaust steam vacuum pressure
$\mathcal{P}_{\mathbf{ma}}$	Pressure of moist air
$\mathcal{P}_{\mathbf{sw}}$	Pressure of saturated water vapour
$\mathcal{R}$	Death penalty multiplier
$\mathfrak{L}$	Total length of a transmission line
$\mathfrak{L}_0$	Total length of a nominal transmission line with no sag, which is also the total spans between all the towers of overhead lines
$\mathfrak{L}_{\mathfrak{s}}$	Total length of a sag transmission line
$\mathfrak{L}_{ij}$	Status of the branch located between the ith and jth busbars
$\mathcal{E}$	Total height of an overhead line tower
$\mathfrak{e}$	Clearance of an overhead line
$\mathfrak{f}$	Pressure of water vapour
$\mathfrak{h}$	Number of temperature-dependent branches
$\mathfrak{m}$	Number of load buses
$\mathfrak{n}$	Number of buses
$\mathfrak{p}$	Recommended population size for the building stage of UFO when a population-based optimization algorithm is used
$\mathfrak{p}_{\mathbf{cond}}$	Wound conductor relative pitch
$\mathfrak{s}$	Sag of an overhead line
pf_R	Receiving-end power factor
pf_S	Sending-end power factor
$\mathrm{Err}_{\mathrm{abs},i}$	Absolute error of the fitness at the ith trial
$\mathrm{Err}_{\mathrm{largest\_ever}}$	The largest absolute error among all the trials

$\mathrm{Err}_{\mathrm{mean}}$	Mean absolute error
$\mathrm{Err}_{\mathrm{smallest\_ever}}$	The smallest absolute error among all the trials
$\mathrm{Err}_{\mathrm{std\_dev}}$	Absolute error standard deviation
$\mathrm{E}_{\mathrm{SVR}}$	Acceptable tolerance to stop the internal optimizer of the SVR algorithm
$\mathrm{GER}_j\left(P_T\right)$	Rate of the jth gas emission produced by all n units to generate P_T
$\mathrm{GER}_{k_i,j}\left(P^*_{i,k_i}\right)$	Rate of the jth gas emission produced by the k_ith unit of the ith power station when the best fuel mixture is used
MAL_j	Maximum allowable limit of the jth gas exhausted to the atmosphere from all n units
$\mathrm{MAL}_{i,j}$	Maximum allowable limit of the jth gas exhausted to the atmosphere from all the ith power station
$\mathrm{PS}_{\mathrm{aux}}$	Array of daily total power consumed by a power station auxiliary equipment
$\mathrm{PS}_{\mathrm{conf}}$	Array of daily configuration dates of all power station units
$\mathrm{PS}_{\mathrm{cost}}$	Array of daily total fuel cost of a power station
$\mathrm{PS}_{\mathrm{net}}$	Array of daily net power delivered by a power station
$\mathrm{PS}_{\mathrm{prod}}$	Array of daily total power produced by a power station
$\mathrm{RES}_{\varpi\ \textbf{forecast}}$	Resolution or updating manner of a forecast where ϖ could be: hourly, daily, monthly, annually, etc
$\mathrm{SS}_{\mathrm{res}}(d)$	Residual sum of squares obtained from the dth order polynomial equation
$\mathrm{Sol}^G_{\mathrm{tab}}$	Global solutions table
$\mathrm{Sol}^L_{\mathrm{tab}}$	Local solutions table
Π-circuit	A pi-shaped circuit to represent the lumped series and shunt parameters of transmission lines
Π-model	An equivalent model to the long-length transmission line model using the Π-circuit
coef0	Independent term, used in SVR for polynomial and sigmoid kernels, which represents one element of Υ

rand	Uniformly distributed random number
μ**GA**	Micro Genetic Algorithm
μ	Emigration rate
μ_0	Permeability of free space
μ_r	Conductor relative magnetic permeability
$\mu_{\mathbf{air}}$	Permeability of air
ν	Vacuum wave-number
$\odot_{k,j}$	The kth internal universal arithmetic operator assigned to the kth predictor of the jth block of UFO; it could be $+$, $-$, $\times$, $\div$, etc
$\mathfrak{g}$	Number of local backup or main-2 DOCRs
ω	Angular frequency
ω_0	Nominal angular frequency
ω_d	Wind direction
ω_s	Wind speed
$\phi(X)$	Penalized cost function
$\propto$	Proportional to
ρ	Conductor resistivity
ρ_0	Conductor resistivity at $\breve{T}_0$
σ_0	Nominal conductivity of a medium between two conductors
σ	Stefan-Boltzmann's constant
$\subset$**-model**	A transmission line model designed based on an subset-shaped circuit of lumped series and shunt parameters
$\supset$**-model**	A transmission line model designed based on a superset-shaped circuit of lumped series and shunt parameters
τ	Reciprocal of square root of time
$\mathbf{ł}_0$	Span between two towers, which is also the ideal cable length or the tower-to-tower length
$\mathbf{ł}_{\mathbf{s}}$	Length of a sag cable placed between two towers

θ_{I_R}	Receiving-end current angle
θ_{V_R}	Receiving-end voltage angle
$\tilde{C}_{i,0}$	Operating cost of the ith unit at $P_{i,0}$ estimated by interpolation techniques
$\tilde{P}^{*}_{i,k_i,\mathrm{EOH}}$	Active power generated by the k_ith degraded/weary unit located in the ith power station when the best fuel mixture is used
$\tilde{P}^{*}_{i,k_i}$	Active power generated by the k_ith unit of the ith power station when that unit is operated by the best fuel mixture and its efficiency is degraded
$\tilde{P}_D$	Forecasted power demand
$\tilde{R}_0$	Shunt resistance at $\breve{T}_0$ and f_0
$\tilde{e}_{o/p}$	Net electrical energy output after subtracting the losses e_L from $e_{o/p}$
$\tilde{x}_2$	Distance estimated from the 1$^{\mathrm{st}}$ terminal of the i-jth branch to the fault location, which is obtained by subtracting x_2 (i.e., the distance from the 2$^{\mathrm{nd}}$ terminal to the fault location) from the total length of the i-jth branch
$\tilde{y}$	Forecasted response
$\uparrow$	Increase
ε	Acceptable error or tolerance
ε	Grey body emissivity
ε_0	Permittivity of free space
$\varepsilon_{\mathbf{air}}$	Permittivity of air
$\varkappa$	Total number of prohibited operating zones
ϖ_i	Prime-mover angular speed of the ith gas turbine
ϱ	Number of DOCRs mounted in a mesh power network
ς	Number of stages needed to run RSA
ζ	Skin-depth
$\{B_{ij}, B_{0i}, B_{00}\}$	Loss-coefficients or B-coefficients

$\{CTI_{ia}, CTI_{ib}\}$	These coordination time intervals are similar to CTI_i, except that the subscripts a and b are used to denote main-1 and main-2 relays
$\{I^*_{ij}, I^*_{ji}\}$	Conjugate values of $\{I_{ij}, I_{ji}\}$
$\{I_{ij}, I_{ji}\}$	Complex current flowing between the ith and jth busbars
$\{M_1, M_2\}$	Population before and after the migration stage
$\{PS_i^{y_i}, PS_j^{u_j}\}$	These plug settings are similar to $\{PS_i, PS_j\}$, except that the superscripts y_i and u_j are used to denote the technologies of the ith primary and the jth backup relays
$\{PS_{ix}, PS_{jx}\}$	These plug settings are similar to $\{PS_i, PS_j\}$, except that the subscript x is either a or b where a means main-1 relay and b means main-2 relay
$\{PS_i, PS_j\}$	Plug settings of the ith primary and the jth backup relays
$\{P_{ij}, P_{ji}\}$	Active power flowing between the ith and jth busbars
$\{Q_{ij}, Q_{ji}\}$	Reactive power flowing between the ith and jth busbars
$\{R_i, R_j\}$	The ith primary and the jth backup relays
$\{R_i^{\text{down}}, R_i^{\text{up}}\}$	Downward and upward ramp rate limits
$\{R_{i,k_i}^{*,\mathbf{down}}, R_{i,k_i}^{*,\mathbf{up}}\}$	Downward and upward ramp rate limits of the k_ith unit located in the ith power station when the best fuel mixture is used
$\{R_{ij}, X_{ij}\}$	Resistance and reactance of the i-jth line
$\{S_{ij}, S_{ji}\}$	Complex power flowing between the ith and jth busbars
$\{TMS_i^{y_i}, TMS_j^{u_j}\}$	These time multiplier settings are similar to $\{TMS_i, TMS_j\}$, except that the superscripts y_i and u_j are used to denote the technologies of the ith primary and the jth backup relays
$\{TMS_{ix}, TMS_{jx}\}$	These time multiplier settings are similar to $\{TMS_i, TMS_j\}$, except that the subscript x is either a or b where a means main-1 relay and b means main-2 relay

$\{TMS_i, TMS_j\}$	Time multiplier settings of the ith primary and the jth backup relays
$\{T'_{i,k}, T'_{j,k}\}$	Operating time of the ith primary and the jth backup relays for a fault occurring at the kth location and during a transient network topology
$\{T_{R_i}^{\mathbf{clb}}, T_{R_i}^{\mathbf{cub}}\}$	Closest known lower and upper operating times of the ith relay measured near the actual fault location
$\{T_{i,k}^{y_i}, T_{j,k}^{u_j}\}$	These relay operating times are similar to $\{T_{i,k}, T_{j,k}\}$, except that the superscripts y_i and u_j are used to denote the technologies of the ith primary and the jth backup relays
$\{T_{i,k}, T_{j,k}\}$	Operating time of the ith primary and the jth backup relays for a fault occurring at the kth location
$\{T_{ia,k}, T_{ib,k}\}$	These relay operating times are similar to $T_{i,k}$, except that the subscripts a and b are used to denote main-1 and main-2 relays
$\{V_i, V_j\}$	Complex voltages at the ith and jth busbars
$\{\delta_i, \delta_j\}$	Voltage angles at the ith and jth buses
$\{\hat{S}_{ij,k}, \hat{S}_{ji,k}\}$	The kth apparent power estimates of the 1st and 2nd ends of the branch located between the ith and jth busbars
$\{\lvert S_{ij}\rvert, \lvert S_{ji}\rvert\}$	Apparent power flowing between the ith and jth busbars
$\{\lvert \bar{S}_{ij}\rvert, \lvert \bar{S}_{ji}\rvert\}$	Biggest apparent power flowing between the ith and jth busbars
$\{\phi_i, \phi_j\}$	Power angles at the ith and jth buses
$\{\psi_{i,k}, \psi_{j,k}\}$	IEC/BS-based coefficients respectively used to calculate the operating times of the ith primary and the jth backup relays, for a fault occurring at the kth location, when their plug settings are kept constant
$\{\mathbf{D}_{ij}^{P}, \mathbf{D}_{ij}^{Q}\}$	Active and reactive power flow directions of the branch located between the ith and jth busbars
$\{\mathbf{L}_{ij,k}^{\mathbf{th}}, \mathbf{L}_{ji,k}^{\mathbf{th}}\}$	The kth thermal indices of the 1st and 2nd ends of the branch located between the ith and jth busbars

$\{\theta_i, \theta_j\}$	Current angles at the ith and jth buses
$\{\varphi_{i,k}, \varphi_{j,k}\}$	ANSI/IEEE-based coefficients respectively used to calculate the operating times of the ith primary and the jth backup relays, for a fault occurring at the kth location, when their plug settings are kept constant
$\{\vartheta_i, \vartheta_j\}$	Coefficients used, with either the IEC/BS or the ANSI/IEEE standard, to calculate the operating times of the ith primary and the jth backup relays, respectively, when their time multiplier settings are kept constant
$\{\widetilde{PS}_{i,k}, \widetilde{PS}_{j,k}\}$	Transformed plug settings obtained at the kth fault location for the ith primary and the jth backup relays, respectively
$\{\xi_i, \xi_j\}$	ANSI/IEEE-based coefficients respectively used to calculate the operating times of the ith primary and the jth backup relays when their time multiplier settings are kept constant
$\{h_i, h_o\}$	Inside and outside surface convective coefficients
$\{i_1, i_2\}$	Incident and reflected current; in the time-domain
$\{v_1, v_2\}$	Incident and reflected voltage; in the time-domain
$\{w_{ia,k}, w_{jb,k}\}$	These weights are similar to $w_{i,k}$, except that the subscripts ia and ib are used to denote main-1 and main-2 relays
$\{x_j^{\mathbf{clb}}, x_j^{\mathbf{cub}}\}$	Closest lower and upper bounds or points predefined near the actual fault location
$\{y_i, u_j\}$	Technologies used in manufacturing protective relays; which can be electromechanical, static, electronic "hardware-based", or numerical "software-based" relays.
$\{y_{ij}, y_{ji}\}$	Series admittance between the ith and jth busbars
$a_{0,j}$	Intercept of the jth block of UFO; where $a_{0,j} \in [a_{0,j}^{\min}, a_{0,j}^{\max}]$
$a_{k,j}$	The kth weight assigned to the kth predictor located in the jth block of UFO; where $a_{k,j} \in [a_{k,j}^{\min}, a_{k,j}^{\max}]$

$b_{k,j}$	The kth exponent assigned to the kth predictor located in the jth block of UFO; where $b_{k,j} \in [b_{k,j}^{\min}, b_{k,j}^{\max}]$
c	Per-unit-length capacitance
c_0	Speed of light in vacuum
c_j	Exponent assigned to the jth analytical function f_j located in the jth block of UFO; where $c_j \in [c_j^{\min}, c_j^{\max}]$
c_k	Shunt capacitance at the kth zone
$c_{o/p}$	Chemical energy output
d	Polynomial degree
$d(X)$	Distance Value
d_p	Penalty multiplier for the pth equality constraint
e_L	Electrical energy losses
$e_{o/p}$	Electrical energy output
$elit$	elite solutions
f	Analytical function, where $f(X)$ means a dependent variable or function of X
f	Regular frequency
f^*	Exact global optimal solution
$f_i^a\left(\varpi_i\right)$	Active fuel-cost function when the ith gas turbine is not connected to the grid
$f_i^r\left(\varpi_i\right)$	Reactive fuel-cost function when the ith gas turbine is not connected to the grid
f_0	Nominal regular frequency
$f_i\left(\dot{V}_i^M\right)$	Active power generated by the ith unit, which is expressed as a function of the total volumetric flow-rate of the M fuels assigned to that unit
f_j	Analytical function assigned to the jth block of UFO; it could be $1 \times ()$, $1/()$, $\sin()$, $\cos()$, $\cot^{-1}()$, $\cosh()$, $\text{csch}^{-1}()$, $\exp()$, $\ln()$, $\log_2()$, $\log_{10}()$, etc
$f_{\text{best\_ever}}$	The best ever fintess among all the trials (i.e., the best of the best)
$f_{\text{best},i}$	Estimated global optimal solution at the ith trial

f_{mean}	Mean of all f_{best} obtained from T_r runs
$f_{\text{std\_dev}}$	Fitness standard deviation
$f_{\text{worst\_ever}}$	The worst ever fintess among all the trials (i.e., the worst of the best)
$f_{\max}$	The largest element of f
$f_{\min}$	The smallest element of f
g	Per-unit-length conductance
$g_i^a\left(P_i\right)$	Active fuel-cost function when the ith gas turbine is connected to the grid
$g_i^r\left(Q_i\right)$	Reactive fuel-cost function when the ith gas turbine is connected to the grid
$g_j\left(X\right)$	Function occupied by the jth block of UFO
g_k	Shunt conductance at the kth zone
g_q	The qth inequality constraint
h	Heat transfer coefficient
h_p	The pth equality constraint
i	Current along a transmission line; in the time-domain
k	Fault location
k	Skin correction factor
k	Thermal conductivity
k_B	Boltzmann's constant
l	Number of fault locations specified on one line
l	Per-unit-length inductance
l_0	Per-unit-length inductance at $\breve{T}_0$
$l_{L_{ij}}$	Total length of the line located between the ith and jth busbars
$l_{\mathbf{turn}}$	Length of one spiral turn of a wound conductor
m	Mutation rate
$m_{\max}$	User-defined maximum mutation rate
$m_{o/p}$	Mechanical energy output
n	Number of generating units

$n_{\mathbf{air}}$	Refractive index of air
q	Heat
r	Conductor radius
r	Per-unit-length resistance
r_C	Penalty hyperparameter of the SVR error term, which represents one element of Υ
r_e	Equivalent radius of multiple conductors per bundle
r_k	Conductor resistance at the kth zone
r_q	Penalty multiplier for the qth inequality constraint
$r_{\omega,0}$	Per-unit-length resistance at ω and $\breve{T}_0$
$r_{\mathbf{layer}}$	Layer radius
$r_{i,J}$	Ratio of the Jth fuel assigned to the ith multi-fuel-based unit
$r_{i,k_i,J}$	Ratio of the Jth fuel assigned to the k_ith unit of the ith power station
s	Variance
s_ς	Geometric series sum of ς terms
t	Time
t_L	Thermal energy losses
t_d	Time in days
t_h	Time in hours
t_m	Time in months
$t_{o/p}$	Thermal energy output
u_c	Clouds
u_d	Dust
u_e	economic growth
$u_i\left(\dot{V}_i^M\right)$	Operating cost of the ith unit, which is expressed as a function of the total volumetric flow-rate of the M fuels assigned to that unit
u_r	Sun rays
u_s	Social and political events

v	Voltage across a transmission line; in the time-domain
$v(X)$	Indication of constraint violations
w_j	Weight assigned to the jth block of UFO; where $w_j \in [w_j^{\min}, w_j^{\max}]$
$w_{i,k}$	Fault probability that might happen at the kth location of a branch protected by the ith relay
x	Specific distance of a transmission line
$x\left(T_{R_i}\right)$	Fault location, which is expressed as a function of the operating time of the ith relay
$x_{l,k}$	Conductor inductive reactance at the kth zone
y	Per-unit-length shunt admittance
y_k	Shunt admittance at the kth zone
z	Per-unit-length impedance
z_0	Sector impedance before sag
z_V	Half sector impedance of the M-model under leveled-spans scenario
z_k	Conductor impedance at the kth zone
Ł	Number of layers; including: input, hidden, and output layers
ß	Number of branches in an electric power network
`bsxfun`	A MATLAB function that applies an element-wise operation to two arrays with implicit expansion enabled
`compet`	Competitive transfer function
`elliotsig`	Elliot sigmoid transfer function
`hardlims`	Symmetric hard limit transfer function
`hardlim`	Positive hard limit transfer function
`logsig`	Logarithmic sigmoid transfer function
`netinv`	Inverse transfer function
`poslin`	Positive linear transfer function
`purelin`	Linear transfer function

`radbasn`	Radial basis normalized transfer function
`radbas`	Radial basis transfer function
`satlins`	Symmetric saturating linear transfer function
`satlin`	Positive saturating linear transfer function
`softmax`	Soft max transfer function
`tansig`	Symmetric sigmoid transfer function
`trainbfg`	BFGS quasi-Newton training algorithm
`traincgb`	Conjugate gradient with Powell/Beale restarts training algorithm
`traincgf`	Fletcher-Powell conjugate gradient training algorithm
`traincgp`	Polak-Ribiére conjugate gradient training algorithm
`traingda`	Gradient descent with adaptive learning rate training algorithm
`traingdm`	Gradient descent with momentum training algorithm
`traingdx`	Variable learning rate gradient descent training algorithm
`traingd`	Gradient descent training algorithm
`trainlm`	Levenberg-Marquardt training algorithm
`trainoss`	One step secant training algorithm
`trainrp`	Resilient backpropagation training algorithm
`trainscg`	Scaled conjugate gradient training algorithm
`tribas`	Triangular basis transfer function
$\mathbf{1}\phi$	Single-phase
2oo3	2 out of 3
$\mathbf{3}\phi$	Three-phase

A&C	Automation and Control
AAAC	All-Aluminum-Alloy-Conductor
AAC	All-Aluminum-Conductor

ABC	Artificial Bee Colony
ABCD	Two-port network matrix elements
AbsDiff	Absolute Difference
AC	Alternating Current
ACAR	Aluminum-Conductor-Alloy-Reinforced
ACO	Ant Colony Optimization
ACSR	Aluminium-Conductor-Steel-Reinforced
Adam	Adaptive Moment Estimation
AE	AutoEncoder
AGC	Automatic Generation Control
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
AMMs	Al-Roomi's Mathematical Machines, which include UFO, UTU, UTR (ULR and UNR), SFM, AMN, MNR, and MAM.
AMN	Artificial Mathematical Network
ANM	Active Network Management
ANN	Artificial Neural Network
ANSI	American National Standards Institute
APL	Adaptive Piecewise Linear
APPSO	Adaptive Personal-best oriented Particle Swarm Optimization
AR	Additive Regression
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
ARIMAX	Auto-Regressive Integrated Moving Average with eXogenous variables
ARMA	Auto-Regressive Moving Average
ARMAX	Auto-Regressive Moving Average with eXogenous variables
ART	Adaptive Resonance Theory
AVC	Automatic Voltage Control

AWG	American Wire Gauge
BA	Bat Algorithm
BBBO	Blended Biogeography-Based Optimization
BBO	Biogeography-Based Optimization
BBO-ANN	Biogeography-Based Optimization - Artificial Neural Network
BBO-EM	Biogeography-Based Optimization with Essential Modifications
BBO-LP	Biogeography-Based Optimization - Linear Programming
BBO/DE	Biogeography-Based Optimization/Differential Evolution
BFO	Bacterial Foraging Optimization
BP	Back-Propagation
BP/EA	Back-Propagation/Evolutionary Algorithm
BReLU	Bipolar Rectified Linear Unit
BREP	Bagging Reduced-Error Pruning tree
BS	British Standards
BS-EPF	Binary Static-Exterior Penalty Function
Btu	British thermal unit
C-inspection	Major overhaul maintenance
C&I	Control and Instrumentation
CA	Contingency Analysis
CAC	Context Aware Crossover
CB	Circuit Breaker
CCPP	Combined-Cycle Power Plant
CDEMD	Cultural Differential Evolution approach using a measure of population's Diversity
CEP	Classical Evolutionary Programming

CGA-MU	Conventional Genetic Algorithm with Multiplier Updating
CGP/SA	Cellular Genetic Programming/Simulated Annealing
CL	Confidence Level
CNN	Convolutional Neural Network
CO	Carbon Monoxide
CO-x	Circuit Opening, where x $\in \{2, 5, 6, 7, 8, 9, 11\}$, which are brands of the Westinghouse's electromechanical OCRs
CO_2	Carbon Dioxide
COA	Combinatorial Optimization Algorithm
Comb-model	A transmission line model designed based on a comb-shaped circuit of lumped/distributed series and shunt parameters
COx	Oxides of Carbon
CPU	Central Processing Unit
CSO	Civilized Swarm Optimization
CST	Common Storage Tank
CT	Current Transformer
CTCC	Current-Time Characteristic Curve
CTI	Coordination Time Interval
CTR	Current Transformer Ratio
CV	Control Valve
DC	Direct Current
DCOCR	Definite-Current Over-Current Relay
DCS	Distributed Control System
DD	Dynamic Demand
DE	Differential Evolution
DER	Distributed Energy Resources
DEU	Dimension Expansion Unit

DF	Degrees of Freedom
DFR	Digital Fault Recorder
DG	Distributed Generation
DL	Deep Learning
DLR	Dynamic Line Rating
DM	Discrimination Margin
DMS	Distribution Management System
DNP	Distributed Network Protocol
DOCR	Directional OverCurrent Relay
DP	Dynamic Programming
DPRS	Double Primary Relay Strategy
DPSO	Decisive Particle Swarm Optimization
DRU	Dimension Reduction Unit
DSP	Digital Signal Processor
DTCC	Distance-Time Characteristic Curve
DTOCR	Definite-Time Over-Current Relay
EA	Evolutionary Algorithm
EBFO	Effective Bacterial Foraging Optimization
EHG	Electro-Hydraulic Governor
ELD	Economic Load Dispatch
ELR	End of Line Resistor
ELU	Exponential Linear Unit
EMS	Energy Management System
EOC	Equivalent Operating Cycles
EOH	Equivalent Operating Hours
EP	Evolutionary Programming
EPA	Environmental Protection Agency
EPF	Exterior Penalty Function

Equals-Sign-model	A transmission line model designed based on an Equals-Sign-shaped circuit of lumped series and shunt parameters
ESN	Echo State Network
ETLBO	Elitism-based Teaching-Learning-Based Optimization
EV	Electric Vehicle
EV Burners	EnVironmental Burners
F&G	Fire and Gas
FA	Fault Analysis
FACTS	Flexible AC Transmission System
FAPSO-NM	Fuzzy Adaptive Particle Swarm Optimization algorithm with Nelder–Mead simplex search
FAPSO-VDE	Fuzzy Adaptive Particle Swarm Optimization algorithm with Variable Differential Evolution algorithm
FC	Feasibility Checker
FCL	Fault Current Limiter
FEP	Fast Evolutionary Programming
FGS	Faculty of Graduate Studies
FME	Fourier–Motzkin Elimination
FPZ	Fault Probability Zone
FS	Fuzzy Systems
FSA	Fast Simulated Annealing
FSNL	Full-Speed No-Load
GA	Genetic Algorithm
GA-PS-SQP	Genetic Algorithm - Pattern Search - Sequential Quadratic Programming
GAB	Binary Genetic Algorithm
GAF	Floating-point Genetic Algorithm

GD	Gradient Descent
GER	Gas Emission Rate
GHP	Geothermal Heat Pump
GMD	Geometrical Mean Distance
GMR	Geometrical Mean Radius
GN	Gauss-Newton
GNU	GNU's Not Unix
GP	Genetic Programming
GPS	Global Positioning System
GSA	Gravitational Search Algorithm
GT	Gas Turbine
GUI	Graphical User Interface
h	Equality constraint
H-PSO	Hierarchical Particle Swarm Optimization
HART	Highway Addressable Remote Transducer
HMI	Human-Machine Interface
HRSG	Heat Recovery Steam Generator
HSI	Habitat Suitability Index
HVDC	High-Voltage Direct Current
HW	Holt–Winters' model
I-model	A transmission line model designed based on an ie-shaped circuit of lumped series and shunt parameters
I/O	Input/Output
IBk	Instance-Based/k-nearest neighbor algorithm
IBT	Inter-Bus Transformer
ICA	Independent Component Analysis
ID	Inner Diameter
IDMT	Inverse Definite Minimum Time

IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IFEP	Improved Fast Evolutionary Programming
IFPE	Instruments-Free Power Estimator
IGA-MU	Improved Genetic Algorithm with Multiplier Updating
IGV	Inlet Guide Vanes
ILP	Integer Linear Programming
IOCR	Instantaneous Over-Current Relay
IoT	Internet of Things
IPF	Interior Penalty Function
ISDA	Iterative Single Data Algorithm
ISI	Island Suitability Index
ISRLU	Inverse Square Root Linear Unit
ISRU	Inverse Square Root Unit
IT	Information Technology
ITOCR	Inverse-Time Over-Current Relay
Jr	Jumping Rate
K*	KStar method
KBES	Knowledge-Based Expert System
KCL	Kirchhoff's Current Law
kNN	k-Nearest Neighbors
KVL	Kirchhoff's Voltage Law
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
L-G Fault	Line-to-Ground Fault
L-L Fault	Line-to-Line Fault

L-L-G Fault	Double Line-to-Ground Fault
L-L-L Fault	Three-Phase Short-Circuit Fault
L-L-L-G Fault	Three-Phase-to-Ground Fault
L-model	A transmission line model designed based on an el-shaped circuit of lumped series and shunt parameters
L1QP	$L1$ soft-margin minimization by Quadratic Programming
LCR	Local Control Room
LDWPSO	Linear Decreasing inertia-Weight Particle Swarm Optimization
LF	Load Flow
LHD	Linear Heat Detector
LHS	Linear Heat Sensors
LM	Levenberg-Marquardt
LMS	Least Median Square
LOA	Limit of Access
LP	Linear Programming
LR	Linear Regression
LReLU	Leaky Rectified Linear Unit
LSTM	Long Short-Term Memory
LVQ	Learning Vector Quantization
LWL	Locally Weighted Learning
LWP	Line Wings Problem
M-circuit	An em-shaped circuit to represent the lumped series and shunt parameters of transmission lines
M-model	An equivalent model to the long-length transmission line model using the M-circuit
M5P	Model Trees Regression
M5R	Model Tree Rules
MA	Moving Average

MAE	Mean Absolute Error
MAL	Maximum Allowable Limit
MAM	Mathematical Artificial Machine
MAPE	Mean Absolute Percentage Error
MCR	Main Control Room
MDO	Multidisciplinary Design Optimization
MFEP	Mean Fast Evolutionary Programming
microSCADA	micro Supervisory Control and Data Acquisition
MINLP	Mixed-Integer Non-Linear Programming
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLR	Multiple Linear Regression
MMS	Manufacturing Message Specification
MNR	Mathematical Neural Regression
MOP	Multi-Objective Problem
MpBBO	Metropolis Biogeography-Based Optimization
MpBBO-SQP	Metropolis Biogeography-Based Optimization - Sequential Quadratic Programming
MPPSO	Mean Personal-best oriented Particle Swarm Optimization
MPSO	Modified Particle Swarm Optimization
MSDO	Multidisciplinary System Design Optimization
MSE	Mean Squared Error
NARMA	Nonlinear Auto-Regressive Moving Average
NASA	National Aeronautics and Space Administration
NB	Number of Busbars
NFE	Number of Function Evaluations
NFT	Near Feasibility Threshold
NIB	Maximum number of busbars interconnected with the jth busbar

NLP	Non-Linear Programming
NLR	Non-Linear Regression
NNPF	Neural Network-based Power Flow
NO	Normally Open
NOx	Oxides of Nitrogen
NPSO	New Particle Swarm Optimization
NPSO-LRS	New Particle Swarm Optimization with Local Random Search
NR	Newton-Raphson
NRCC	National Research Council Canada
NRPF	Newton-Raphson Power Flow
NSERC	Natural Sciences and Engineering Research Council
NSM	No Same Mate
NSUARB	Nova Scotia Utility and Review Board
O-model	A transmission line model designed based on an oh-shaped circuit of lumped series and shunt parameters
O/P	OutPut
OBBO	Oppositional Biogeography-Based Optimization
OBJ	Objective Function
OBL	Opposition-Based Learning
OCR	Over-Current Relay
OD	Outer Diameter
ODE	Ordinary Differential Equation
OEM	Original Equipment Manufacturer
OLF	Over-Load Factor
OLS	Ordinary Least Squares
OMF-ELD	Optimization/Modeling-Free Economic Load Dispatch
OMF-UC	Optimization/Modeling-Free Unit Commitment

ONNX	Open Neural Network Exchange
OPF	Optimal Power Flow
OR	Operations Research
ORC	Optimal Relay Coordination
OS	Operation System
P/B	Primary/Backup
P&ID	Piping and Instrumentation Diagram
PB	Population-Based
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PE-RT	Polyethylene of Raised Temperature
PerErr	Percentage Error
PEX	Cross-linked PolyEthylene
PF	Power Flow
PLC	Programmable Logic Controller
PLW	Public Logs Warehouse
PM	Periodic Maintenance
PMB	Partial Migration-Based
PMB-BBO	Partial Migration-Based Biogeography-Based Optimization
PMU	Phasor Measurement Unit
PNN	Probabilistic Neural Network
ppm	parts per million
PPSO	Personal-best oriented Particle Swarm Optimization
PQ bus	Load bus
PR	Pace Regression
PReLU	Parametric Rectified Linear Unit
Profibus	Process Field Bus
PS	Plug Setting or Pick-up Setting

PSM	Plug Setting Multiplier
PSO	Particle Swarm Optimization
PSO-B-SA	Particle Swarm Optimization-Based-Simulated Annealing
PSO-LRS	Particle Swarm Optimization with Local Random Search
PSO-SQP	Particle Swarm Optimization - Sequential Quadratic Programming
PSS	Power System Stability
PSTLBO	Producer-Scrounger Teaching-Learning-Based Optimization
PT	Potential Transformer
PTW	Permit to Work
PV	PhotoVoltaic
PV bus	Generator or voltage-controlled bus
PVC	PolyVinyl Chloride
QPSO	Quantum-behaved Particle Swarm Optimization
R	Coefficient of Correlation
r.p.m.	revolutions per minute
R^2	Coefficient of Determination
RAM	Random Access Memory
RBF	Radial Basis Function neural network
REG	Voltage regulation
RelErr	Relative Error
ReLU	Rectified Linear Unit
REP	Reduced Error Pruning trees
RF	Radio Frequency
RH	Relative Humidity
RMSE	Root Mean Squared Error

RNN	Recurrent Neural Network
RO	Reverse Osmosis
RReLU	Randomized leaky Rectified Linear Unit
RSA	Random Search Algorithm
RTU	Remote Terminal Unit
RUFO	Recurrent Universal Functions Originator
SA	Simulated Annealing
SA	Situational Awareness
SA/TS	Simulated Annealing/Tabu Search
SAC	Semantic Aware Crossover
SAPF	Self-Adaptive Penalty Function
SAPS	Small Autonomous Power Systems
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
SARIMAX	Seasonal Auto-Regressive Integrated Moving Average with eXogenous variables
SARMAX	Seasonal Auto-Regressive Moving Average with eXogenous variables
SBS	Soft Brood Selection
SC	Standard Crossover
SC	Super-Capacitor
SCADA	Supervisory Control and Data Acquisition
SCPP	Simple-Cycle Power Plant
SE	Standard Error
SE	State Estimation
SELU	Scaled Exponential Linear Unit
SFC	Static Frequency Converter
SFCC	Sliced Fuel-Cost Curves
SFM	Support Function Machine
SFP	Superiority of Feasible Points

sGA	Stud Genetic Algorithm
SiLU	Sigmoid Linear Unit
SIV	Suitability Index Variable
SLP	Single-Layer Perceptron
SMB	Single Migration-Based
SMIB	Single Machine Infinite Bus
SMO	Sequential Minimal Optimization
SMOReg	Support Vector Poly Kernel Regression
SOM	Self-Organizing Map
SOx	Oxides of Sulfur
SP3	Service Pack 3
SPMB	Simplified Partial Migration-Based
SPSO	Standard Particle Swarm Optimization
SQNL	Square Nonlinearity
SQP	Sequential Quadratic Programming
SR	Slicing Resolution
SR	Symbolic Regression
SReLU	S-shaped Rectified Linear Activation Unit
SSC	Semantic Similarity-based Crossover
SSE	Sum of Squared Errors; or, Error Sum of Squares
SSMB	Simplified Single Migration-Based
ST	Steam Turbine
StDev	Standard Deviation
STI	Selective Time Interval
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVR	Support Vector Regression
SWER	Single Wire Earth Return
SWH	Seasonal Holt–Winters' linear exponential smoothing

T-circuit	A ti-shaped circuit to represent the lumped series and shunt parameters of transmission lines
T-model	An equivalent model to the long-length transmission line model using the T-circuit
TAT	Temperature After Turbine
TCCC	Time-Current Characteristic Curve
TDNN	Time Delay Neural Network
TDR	Time Domain Reflectometer
TDS	Time Dial Setting
TEM	Transaction Evaluation and Management
TF	Transfer Function
TFB	Temperature/Frequency-Based
TIT	Turbine Inlet Temperature
TLBO	Teaching-Learning-Based Optimization
TLS	Time Lever Setting
TMS	Time Multiplier Setting
TotErr	Total Error
TRR	Trust-Region Reflective
TS	Tabu Search
TS	Tap Setting
UC	Unit Commitment
UCA2	Utility Communication Architecture - version 2
UFO	Universal Functions Originator
UHCs	Unburned HydroCarbons
ULR	Universal Linear Regression
UNR	Universal Nonlinear Regression
UPFC	Unified Power Flow Controller
USR	Universal Symbolic Regression
UTR	Universal Transformation-based Regression

UTU	Universal Transformation Unit
UV	UltraViolet
VAT	Value-Added Tax
VIF	Variance Inflation Factor
VPP	Virtual Power Plant
VS	Voltage Stability
VT	Voltage Transformer
WP	Water Pump
WSCC	Western System Coordinating Council
WSO	Wasp Swarm Optimization

Acknowledgements

I begin in the name of ALLAH, the Gracious, the Merciful. The greatest who created the Universe and sustaining it. To Him are due all the gratitude and praise. Thank God for the wisdom and perseverance bestowed upon me during this research project, and indeed, throughout my life.

In memory of my father
I owe it all to you.

To my mother
With love and eternal appreciation.

To my wife
Thank you for supporting me to finish my dissertation.

I would like to express my heartfelt thanks to my late supervisor Prof. Mohamed E. El-Hawary. A more supportive and considerate supervisor I could not have asked for. There were many times where I had reached the 'crossroads' and each time Prof. El-Hawary was there to steer me towards the right path. He was prepared to sit and listen to my troubles and always made me feel as if my work mattered. His willingness to offer me so much of his time and intellect is the major reason the researches listed in this dissertation were published. May his soul rest in peace.

I would also like to repeat my great thanks to my late supervisor Prof. El-Hawary for providing me the NSERC research assistantship funds, which really helped me and supported my PhD. Besides, I gratefully acknowledge the funding received from FGS and the Bruce and Dorothy Rosetti Engineering Research Scholarship.

I cannot forget the favor of Prof. Dmitry V. Trukhachev who stands with me, especially during my suffering from losing Prof. El-Hawary, by accepting to co-supervise me until finalizing my graduation. Many thanks to you and your kind spirit.

I would like to thank the external examiner, Prof. Sherif O. Faried (University of Saskatchewan), and the internal examiners, Prof. William J. Phillips and Prof. Jason J. Gu, to serve as my Ph.D. committee members. Their support and knowledge was

a great help throughout my research and it is my honor to have them review my work and provide me with valuable feedback.

And of course, I would like to thank the departmental representative, Prof. Sergey A. Ponomarenko, and the chair, Prof. Norbert R. Zeh, for their valuable time in monitoring and arranging my Ph.D. defense. The same thanks go to Prof. Jose A. Gonzales-Cueto for moderating my P.hD. candidacy exam.

Last but not least, I would like to thank the administrative and secretaries, Mrs. Nicole Smith, Mrs. Jascinth Butterfield, Mrs. Mary Little, Ms. Rebecca Baccardax, Mrs. Tamara Cantrill, and Mrs. Helena Martel for their continued support and help.

Chapter 1

Introduction

A century ago, pioneers (such as Edison, Tesla, Heaviside, etc) were working to pave the first infrastructure of what it is now known as electric power systems[1] [80, 139, 244, 268]. From that time until now, many other pioneers (such as Nathan Cohn, Leon Kirchmeyer, Thomas E. DyLiacco, William F. Tinney, and Lester H. Fink) were continuously working to develop some new methods and approaches to solve the inherent problems and weaknesses faced in those old systems [109, 123].

During the last two decades[2], many advanced techniques and technologies have been either theoretically proposed or practically employed to enhance the operation of existing electric power systems and how they can be effectively monitored, controlled, and protected. In reality, electric power systems cannot be efficiently operated without considering the real variables that have direct and indirect effects on the overall performance and accuracy [209, 326, 386, 399]. Also, most of the techniques proposed by these pioneers and other leading researchers are presented for some specific applications. For example, if a design is suggested to be exclusively used for protection purposes, then its side effects on other power systems analysis cannot be predicted if a lack of information is faced with that design [52, 301]. Actually, modern power systems are highly interconnected between each other and they are a result of an integration of many disciplines and fields of science. To show how this research area is highly complex, a graphical overview is illustrated in Figure 1.1 [52, 109, 123, 209, 301, 326, 386, 399].

As can be clearly seen from this figure, electric power systems engineering can be seen as a combination or mixture of different branches and disciplines of engineering

[1] Electric power systems consist of three principal parts: generation, transnimission/subtransmission, and distribution systems.

[2] This period is called the digital or information age.

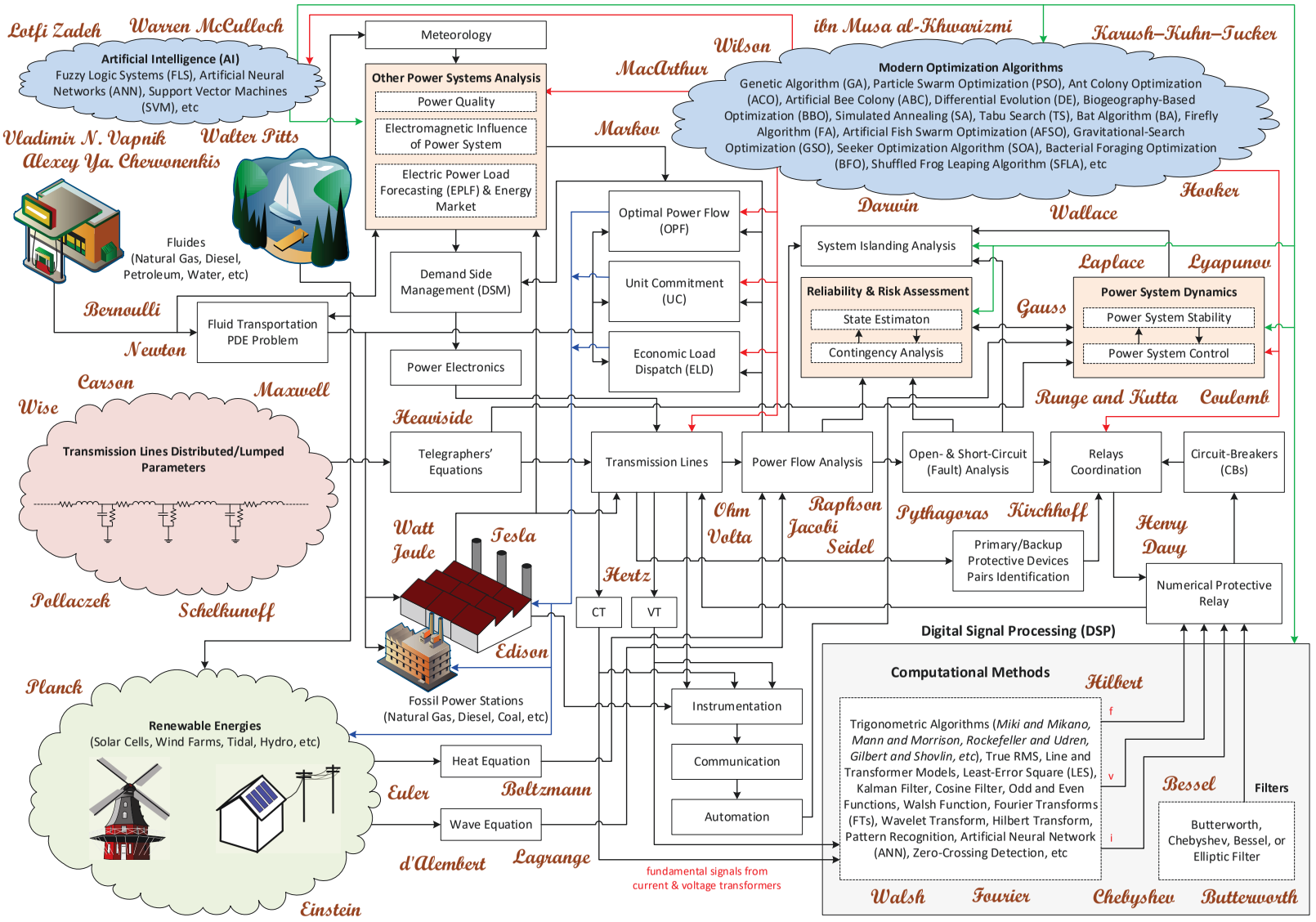


Figure 1.1. An Illustration of Realistic Modern Electric Power Systems (Version No. 6 - Last Update: Feb. 23, 2020)

(including electrical and electronics, communication, computer, process instrumentation and control, mechanical, and chemical), pure and applied mathematics, theoretical and applied physics, computer and data science, etc[3]. From this fact, someone may observe that there are a few studies presented in the literature to cover some of these interconnected topics. This means that the door is widely opened to the researchers who have the ability and capability to conduct their researches on these stages as one realistic design. We believe that such designs must be established to meet the requirements of modern highly complicated power systems. This forces us to employ many advanced methods and utilizing many practical skills and experiences. Merging these two essential parts in our realization process can reveal many hidden facts and phenomena and can effectively minimize the current gap between the real-world problems and their mathematical models. Also, this approach can point out the flaws associated with many well-established mathematical models that are presented in the literature and considered as the backbone of many other power system problems. Furthermore, some significant modifications and/or replacements can be made to enhance the system performance with more accuracy. This can lead us to some completely new promising achievements where theoretical and practical problems can be addressed and then solved by some innovative smart tools and methods. All these things can be directly and indirectly reflected on many future power products where the four important criteria (simplicity, optimality, feasibility, and processing speed) are satisfied.

1.1 Motivations

Selecting this challenging Ph.D. topic is made based on my personality that loves to think outside the box and flees from flying with the swarm. It does not like to stick with the traditional ways and tries its best to find other possible approaches to

[3]If someone looks into it from the optimization side, then he/she may find that there is a new specialized field called "Multidisciplinary Design Optimization (MDO)". It is also known as "Multidisciplinary System Design Optimization (MSDO)". It is a field of engineering, which uses optimization techniques to solve design problems incorporating a number of disciplines. This field is just one stage of our broad journey! Here, we are dealing with everything; analytical methods, numerical methods, inspecting the current techniques, investigating the effects of hidden and neglected phenomena, merging existing topics, suggesting new topics, raising our concerns, proposing our ideas, and finally presenting our innovative and novel solutions.

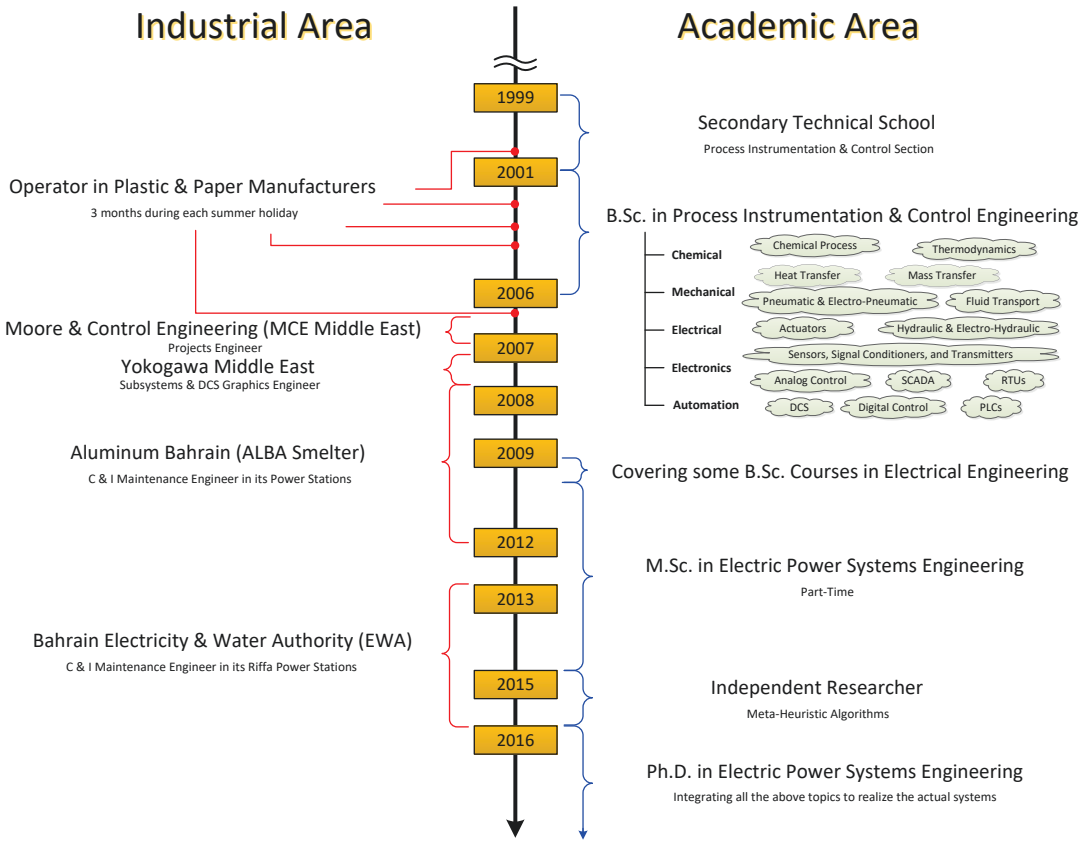


Figure 1.2: Background Timeline Before Joining Dalhousie University

achieve the same objectives. I made that decision based on my background history in academic and industrial areas. These areas are depicted in Figure 1.2.

I strongly believe that the path to get some innovative ideas and leading studies is by merging different fields of electric power systems engineering; as shown in Figure 1.3. Moreover, other engineering disciplines cannot be ignored! For example, Figure 1.4 shows four different engineering disciplines. Mechanical engineering is important to study the heat transfer and fluid transport of thermal generating plants (such as steam turbines, gas turbines, diesel generators, etc). Chemical engineering is important to study the reaction inside combustion chambers, fouling in heat-exchangers, corrosion in metallic materials, etc. Instrumentation and control (C&I) engineering is important to see the complete automation picture from the field level to the supervisory level in every part of electric power systems. Furthermore,

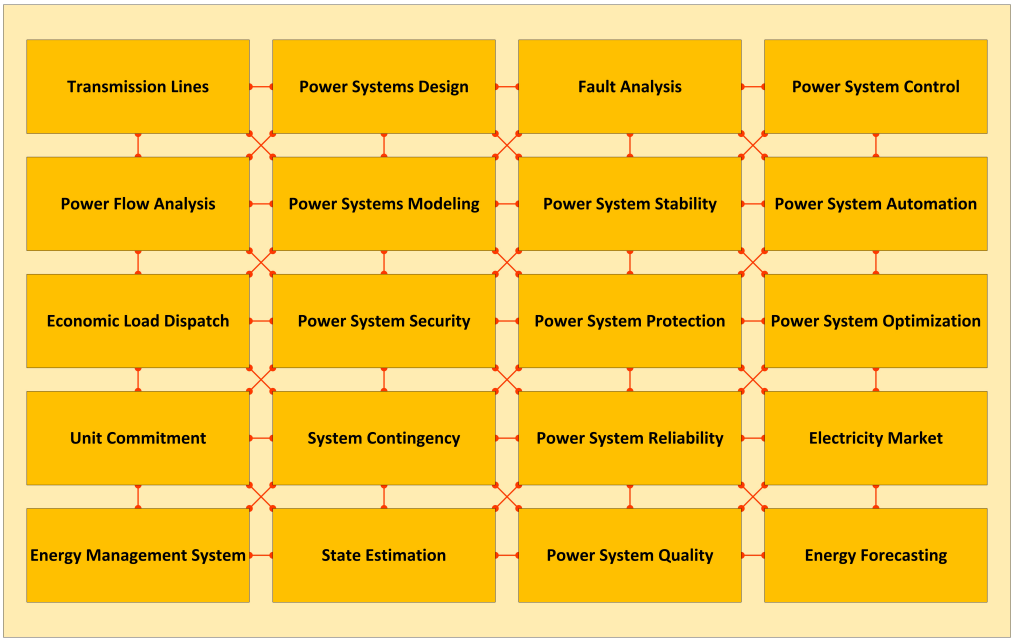


Figure 1.3: Merging Different Fields of Electric Power Systems Engineering

because engineering is the use of scientific principles to design and build things, so the other disciplines (such as mathematics, physics, and computer and data science) are also highly important.

1.2 Objectives and Goals

To sum all the preceding points, the objectives and goals from doing this Ph.D. are split into five main streams:

1. Creating superior hybrid optimization algorithms.

2. Realizing electric power systems and integrating topics from different fields and disciplines[4].

3. Embedding artificial intelligence (AI) algorithms to find more efficient solutions to some power system problems.

4. Developing innovative machine learning (ML) computing systems.

[4]The optimization algorithms designed before will be used to solve unrealistic and realistic optimization problems.

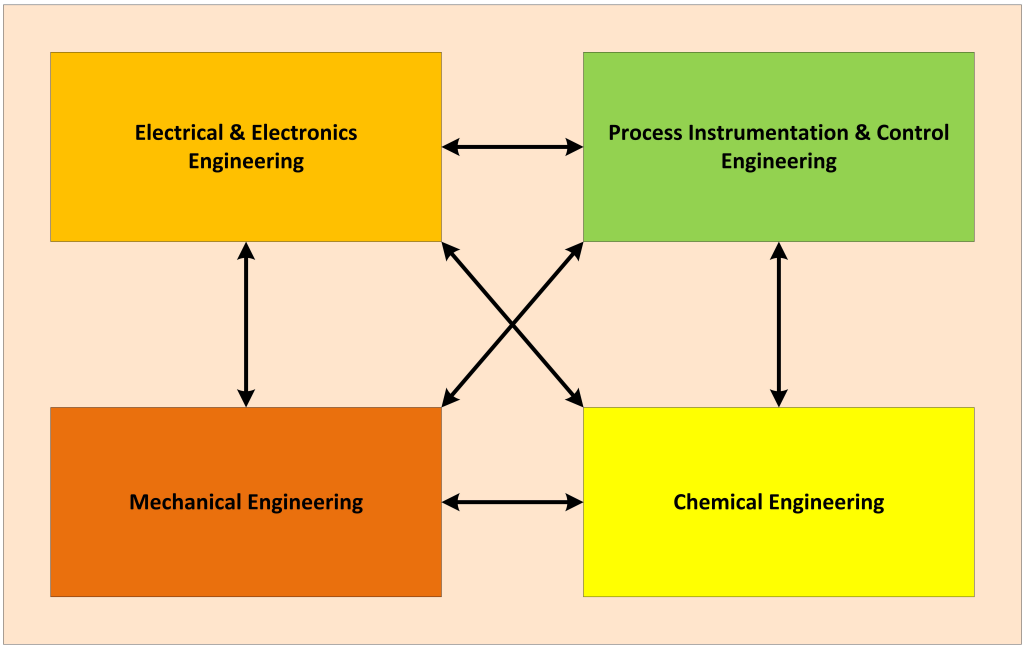


Figure 1.4: Interconnection Between Different Disciplines of Engineering

5. Designing novel optimization-free/modeling-free techniques.

The following subsections describe these five streams in detail:

1.2.1 Designing Superior Optimization Algorithms

Finding the best solutions to complex engineering problems cannot be achieved with traditional mathematical analysis. They need some special tools called optimization algorithms to find the minimum or maximum values based on the statements given in the problem. In the past, the researchers did not have another choice except to use the classical optimization algorithms. Because the state-of-the-art mathematical models are very complex, so obtaining the best optimal values cannot be attained, or precisely very hard to be obtained, using these algorithms. Also, coding them in computing machines require high efforts from the researchers. Thus, some pioneers suggested new optimization algorithms that are built based on some mechanisms inspired by nature. These algorithms come with different names, such as nature-inspired algorithms, evolutionary algorithms, meta-heuristic algorithms, stochastic algorithms, modern optimization, non-traditional optimization, etc. For example, genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO),

biogeography-based optimization (BBO), differential evolution (DE), simulated annealing (SA), artificial bee colony (ABC), and wasp swarm optimization (WSO) are some types.

In this part, our mission is to design new hybrid optimization algorithms, so the strengths of each individual algorithm can be integrated and at the same time, their weaknesses can be eliminated or minimized. Then, validating the performance of these proposed superior algorithms by using a wide range of popular benchmark functions[5]. The evaluation process should be carried out using different statistical tests and sensitivity analysis.

Here, the hybrid approaches will be between classical and meta-heuristic optimization algorithms. Linear programming (LP) and nonlinear programming (NLP) will be used for the first category. For the second one, BBO, SA, and DE are used. Some of them will be double-hybrid approaches, while others will be triple-hybrid approaches. The goal of hybridizing two meta-heuristic optimization algorithms is to have a good balance between the exploration level and the exploitation level. The implementation of LP or NLP is to have a local optimizer; or, in other words, to act as a fine-tuning stage. These hybrid optimization algorithms will be used to solve many ELD and ORC problems. Some of them are continuous problems, and some others are mixed-integer and combinatorial problems.

The BBO algorithm will also be used to optimize the hyperparameters of feed-forward neural networks to have precise energy forecasting results; for some datasets taken from Nova Scotia Power Inc. Furthermore, the primitive random search algorithm (RSA) is modified by embedding a multi-start strategy with a sub-algorithm to minimize its search space in each new start.

For some of our novel ML computing systems covered in Subsection 1.2.4, BBO and DE can be hybridized with the opposition-based learning (OBL) algorithm before being used to build the overall ML model. For the curve fitting stage, the classical tools will be the ordinary least squares (OLS) algorithm, the trust-region reflective (TRR) algorithm, and Levenberg-Marquardt (LM) algorithm.

[5]It has to be said that hundreds of these benchmark functions have been revised and published in my comprehensive repository that can be logged-in via the link: https://www.al-roomi.org.

1.2.2 Power System Realization and Integration

In this phase, the goal is to realize many power system studies presented in the literature. It starts from the Telegrapher's equations[6], which have a major impact on power transmission lines and their precise distributed parameter models. Realizing transmission lines have a direct impact on power flow (PF) analysis, which is considered as the backbone of many other power system studies, such as optimal power flow (OPF), power system operation and control, power system security and contingency, state estimation, power system dynamics and stability, fault analysis, power system protection, load forecasting, electricity market, etc [49, 57–59, 64, 65, 109, 123, 149, 155, 156, 162, 175, 211, 213, 291, 301, 311, 326, 386, 399]; which are depicted in Figure 1.1 and Figure 1.3. These are some prospective areas of our research. The techniques used in smart grids [325], renewable energy [6, 365, 383], and energy storage elements [215, 383] could be included as extensions of the preceding conventional models.

In our realization mission, we will study many hidden variables, such as the surrounding temperature, pressure, humidity, aging of electrical components, etc, and their impacts on the overall performance. This means that many topics from various disciplines will be covered, such as heat transfer, fluid transport, instrument devices, etc, with the help of regression analysis and other ML tools.

Therefore, the gap between the actual problems and the corresponding mathematical models presented in the literature (as books, chapters, technical reports, journals, conferences, websites, software, etc) can be effectively minimized. Based on this, the confidence level of electric power systems modeling and the dependability on energy companies and industries can be increased. Such studies can be considered as bases of future infrastructures for prospective next-generation smart grids and modern electric power systems. Focusing on these highly important researches can shape our understanding of electric power systems engineering and the interconnections and relations with other branches of science.

The integration phase is done by merging different topics into one. That is, creating new topics from existing topics. This approach can open the door wide to conduct many types of research and being a pioneer in many future areas. As said

[6] These partial differential equations are known in mathematics as the Telegrapher's equations. These equations have many applications in applied mathematics and electrical engineering [205, 259].

before, many proposals are raised in this stream of our Ph.D. research.

1.2.3 Embedding AI Tools

In electric power system problems, many types of uncertainty could happen at any time. Thus, by just depending on the existing classical- and nature-inspired optimization algorithms will not help to solve the subjectively judged problems. For example, some phenomena of real electric generators are mathematically modeled based on some data collected from sensors and judgments of expert engineers where the settings of some unknown/ignorable variables are approximated and compromised[7]. The main question that should be raised here is: *How can we ensure that these observed data, judged models, and approximated/compromised settings are 100% correct?*

Therefore, hybridizing AI computing systems with classical and evolutionary optimization algorithms could solve many real-world technical problems with some promising results. For example, artificial neural networks (ANNs) and support vector machines (SVMs) have some distinct capabilities where, through their learning process, good results could be obtained smartly without referring to any highly accurate mathematical models.

1.2.4 Developing Innovative ML Computing Systems

The goal of this phase is to try to design new computing systems that can merge the strengths of linear regression (LR) and nonlinear regression (NLR) with AI-based techniques. For example, classical regression techniques can provide good explanations and interpretations for simple datasets, but they lack the accuracy for complicated datasets; and vice versa for ANNs and SVMs. Our novel ML computing systems are supposed to generate highly accurate models without facing any linearization problem (as faced in LR), a user-defined model with its initial coefficients and their side constraints (as faced in NLR), and without trapping into the black-box issue (as faced in ANNs and SVMs).

[7] This could be due to the dynamic nature of the surrounding weather conditions and market.

1.2.5 Designing Novel Optimization-Free & Optimization-Free / Modeling-Free Techniques

The main issue associated with optimization-based techniques is that precise mathematical models are required to explain the variability of real-world problems, and then these models need to be optimized effectively and correctly. This is why electric power systems engineering is considered as one of the most complicated fields in engineering.

Here, we are trying to find some root solutions to these very stiff/headache problems by designing totally different techniques that can handle these problems without implementing any optimization algorithm; and sometimes, without even using any precise model. The operating principles of these algorithms are based on executing some steps to create big real data, then filtrating infeasible settings, and finally sorting the best feasible settings. Actually, we have partially succeeded in ELD.

1.3 Main Contributions

This dissertation contains many contributions that have been successfully achieved toward each one of the preceding five goals. The main contributions are listed below:

1. Different highly powerful hybrid optimization algorithms are successfully designed. Some of them are between meta-heuristic optimization algorithms and the others between meta-heuristic and classical optimization algorithms. Some of them are double-hybrid schemes and the others are triple-hybrid schemes.

2. One of the preceding superior hybrid optimization algorithms is applied to solve some standard ELD problems and it beats all the best records known in the literature.

3. The regression transformation technique is applied to the classical optimal relay coordination (ORC) model so that the time multiplier settings can be fixed to have a linearized objective function that can be further minimized by tuning the plug settings via integer linear programming (ILP).

4. The infeasibility of multiple time-current characteristic curves (TCCCs) of existing ORC problems is solved by searching for the optimal unified TCCC for

all the relays.

5. The biogeography-based optimization (BBO) algorithm is hybridized with a classical feed-forward ANN so that its hyperparameters can be optimized. This hybrid ML algorithm is applied to forecast some Nova Scotian heat consumption profiles.

6. A basic random search algorithm (RSA) is hybridized with a standard support vector machine (SVM) so that its hyperparameters can be optimized. This hybrid ML algorithm is applied to forecast the peak-load of Nova Scotia during Winter 2018-2019.

7. A new precise lumped-parameter transmission line model, called the M-model, is presented and its performance is evaluated against all the known standard models. This new model excelled in most performance criteria.

8. A comprehensive realization study is done in ideal transmission lines to integrate the effects of the surrounding weather conditions (temperature, humidity, pressure, etc), system frequency, load current, and cable design/status/age. This study includes many uncovered phenomena that might affect the values of distributed series and shunt parameters.

9. The preceding M-model is applied to solve the difference in values of ideal and sag transmission line parameters by using two innovative ways.

10. This is the first time in the literature to mathematically estimate the inductance, capacitance, and conductance temperature coefficients of sag transmission lines. Three innovative approaches are introduced with/without knowing the conductor temperature and with/without involving the heat transfer topic.

11. Applying some of the preceding transmission line realizations to solve different power system problems.

12. Realizing the classical ELD model to include many hidden real-world challenges and technical problems. This study shows, with some proofs, that all the existing ELD models could violate the optimality and/or the feasibility criteria. Thus, a completely new non-traditional ELD model is presented.

13. The classical ORC model is realized by considering different relay technologies[8].

14. The classical ORC model is realized by considering the double primary relay strategy (DPRS) where each bus is protected by main-1 and main-2 relays[9]. This real phenomenon has not been covered in the literature yet, so a new mathematical model is presented where the dimension of this non-conventional ORC problem is larger and thus much harder to be solved.

15. This is the first time in the literature to apply ANNs to estimate the power flow magnitudes, directions, and losses with satisfying the power balance equation of ELD. These estimators need just to know the power inputs and outputs (i.e., generators and loads) and the status of network branches to provide their highly precise estimations.

16. This is the first time in the literature to solve ELD problems without using any optimization algorithm.

17. This is the first time in the literature to solve ELD problems without using any optimization algorithm or mathematical model[10]. The method is applied to solve a real ELD problem and it shows an encouraging result.

18. A new transformation-based non-piecewise linear fuel-cost model is developed. This model can compete with quadratic and cubic models and at the same time, it can be optimized by LP. Also, it is much simpler and faster than the classical piecewise linear model.

19. The fuel-cost function is borrowed from power system operation and utilized in state estimation to provide an indirect backup measurement of power generated by thermal units. Then, the well-known "2 out of 3" voting logic is borrowed from C&I engineering to compare the signals received in both energy management and automation systems; i.e., EMS and SCADA/DCS. This approach

[8] i.e., electromechanical, solid-state, digital, and numerical technologies.

[9] Some references called them primary and local-backup relays.

[10] ANNs and SVMs can solve ELD problems without referring to any mathematical model, but they contain internal optimization algorithms coded in the learning stage. Also, employing ML tools in place of optimization algorithms will complicate the process, and thus it violates the simplicity criterion.

can ensure that precise information is shared between power system operation, protection, state estimation, control, and automation; which has a significant impact on the overall performance.

20. Some possible approaches are presented to trade non-electric energy sources in the next-generation smart grids. The study proposes to install smart pipelines in parallel with powerlines. These pipelines can deliver hot water from thermal solar collectors, hot/cool air from geothermal heating/cooling systems, methane gas from biomass units, and wood fuel from public logs warehouses (PLWs). In this study, a heat-transfer analysis is covered to show the effects of pipe diameter, pipe material, insulation material, and surrounding weather conditions across different pipeline lengths. This study proposes many modifications and extensions towards the classical models used in ELD and unit commitment (UC) studies.

21. It proposes to use linear heat sensors (LHS), taken from fire and gas (F&G) systems, as a very cheap alternative to existing electric power protective devices.

22. The classical TCCC is transposed to act as a current-time characteristic curve (CTCC), and then to a distance-time characteristic curve (DTCC) to locate faults in mesh networks. In this study, the operating times and fault type, detected by the two-end numerical directional overcurrent relays (DOCRs), are utilized to precisely approximate the fault locations by using interpolation- and regression-based models.

23. A novel ML computing system called "universal functions originator (UFO)" is presented. This system can approximate functions by generating highly precise linear and nonlinear mathematical models, and thus it can act as a general LR/NLR unit. UFO can achieve the accuracy and precision criteria without losing the explainability and interpretability criteria.

24. The UFO computing system is successfully hybridized with LR and SVMs to act as a universal transformation unit (UTU). This unit is hybridized with ANNs as a "calibration layer" to take care of the nonlinearity of the given dataset

before being processed. The study also proposes some possible highly advanced UFO-based frameworks that could be designed in the future.

25. All the preceding UFO and UFO-based ML computing systems are successfully implemented to solve some power system problems; particularly, predicting electric power and locating faults.

1.4 List of Publications

The following list shows the published and completed studies:

Published:

1. A. R. Al-Roomi and M. E. El-Hawary, "Estimated Economic Load Dispatch Based on Real Operation Logbook," in *2016 IEEE Electrical Power and Energy Conference (EPEC)*, Ottawa, ON, 2016, pp. 1-6.

2. A. R. Alroomi and M. E. El-Hawary, "Optimal Coordination of Directional Overcurrent Relays Using Hybrid BBO/DE Algorithm and Considering Double Primary Relays Strategy," in *2016 IEEE Electrical Power and Energy Conference (EPEC)*, Ottawa, ON, 2016, pp. 1-7, 2016.

3. A. R. Al-Roomi and M. E. El-Hawary, "A New Technique to Locate Faults in Distribution Networks Based on Optimal Coordination of Numerical Directional Overcurrent Relays," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, 2017, pp. 1-6.

4. A. R. Al-Roomi and M. E. El-Hawary, "Optimal Coordination of Directional Overcurrent Relays Using Hybrid BBO-LP Algorithm with the Best Extracted Time-Current Characteristic Curve," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, 2017, pp. 1-6.

5. A. R. Al-Roomi and M. E. El-Hawary, "A New Realistic Optimization-Free Economic Load Dispatch Method Based on Maps Gathered from Sliced Fuel-Cost Curves," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, 2017, pp. 1-6.

6. A. R. Al-Roomi and M. E. El-Hawary, "A Novel Multiple Fuels' Cost Function for Realistic Economic Load Dispatch Needs," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Saskatoon, SK, 2017, pp. 1-6.

7. A. R. Al-Roomi and M. E. El-Hawary, "Effective Weather/Frequency-Based Transmission Line Models−Part I: Fundamental Equations," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Saskatoon, SK, 2017, pp. 1-6.

8. A. R. Al-Roomi and M. E. El-Hawary, "Effective Weather/Frequency-Based Transmission Line Models−Part II: Prospective Applications," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Saskatoon, SK, 2017, pp. 1-6.

9. A. R. Al-Roomi and M. E. El-Hawary, "Metropolis Biogeography-Based Optimization," in *Information Sciences*, vol. 360, pp. 73-95, Sept. 2016.

10. A. R. Al-Roomi and M. E. El-Hawary, Economic Load Dispatch Using Hybrid MpBBO-SQP Algorithm, in: *Nature-Inspired Computation in Engineering* (X.-S. Yang, editor) Springer International Publishing, pp. 217-250, 2016.

11. A. R. Al-Roomi and M. E. El-Hawary, "Fuel Cost Modeling for Spinning Reserve Thermal Generating Units," in *2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Québec City, QC, 2018, pp. 1-6.

12. A. R. Al-Roomi and M. E. El-Hawary, "Optimizing Load Forecasting Configurations of Computational Neural Networks," in *2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Québec City, QC, 2018, pp. 1-6.

13. A. R. Al-Roomi and M. E. El-Hawary, "New Heat Energy Trading Concepts for the Next Generation Smart Grids," in *2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Québec City, QC, 2018, pp. 1-6.

14. A. R. Al-Roomi and M. E. El-Hawary, "Possible Approaches to Trade Non-Electric Energy Sources in the Next Generation Smart Grids," in *2018 IEEE Electrical Power and Energy Conference (EPEC)*, Toronto, ON, 2018, pp. 1-6.

15. A. R. Al-Roomi and M. E. El-Hawary, "Can Linear Heat Sensors be a Good and Practical Replacement of Traditional Protective Fuses?," in *2018 IEEE Electrical Power and Energy Conference (EPEC)*, Toronto, ON, 2018, pp. 1-6.

16. A. R. Al-Roomi and M. E. El-Hawary, "How to Improve Linear Fuel-Cost Function to Compete with Quadratic and Cubic Functions," in *2018 IEEE Electrical Power and Energy Conference (EPEC)*, Toronto, ON, 2018, pp. 1-6.

17. A. R. Al-Roomi and M. E. El-Hawary, "Is It Enough to Just Rely on Near-End, Middle, and Far-End Points to Get Feasible Relay Coordination?," in *2019 IEEE 32th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Edmonton, AB, 2019, pp. 1-5.

18. A. R. Al-Roomi and M. E. El-Hawary, "Universal Functions Originator−Part I: Design," in *2019 IEEE 32th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Edmonton, AB, 2019, pp. 1-6.

19. A. R. Al-Roomi and M. E. El-Hawary, "Universal Functions Originator−Part II: Evaluation," in *2019 IEEE 32th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Edmonton, AB, 2019, pp. 1-6.

20. A. R. Al-Roomi and M. E. El-Hawary, "Optimal Coordination of Double Primary Directional Overcurrent Relays Using a New Combinational BBO/DE Algorithm," in *Canadian Journal of Electrical and Computer Engineering*, vol. 42, no. 3, pp. 135-147, Summer 2019. doi: 10.1109/CJECE.2018.2802461

21. A. R. Al-Roomi and M. E. El-Hawary, "Estimating Power Flow Directions Using Off-Line PF Analysis and Artificial Neural Networks," in *2019 IEEE Electrical Power and Energy Conference (EPEC)*, Montréal, QC, 2019, pp. 1-6.

22. A. R. Al-Roomi and M. E. El-Hawary, "Estimating Complex Power Magnitudes Using a Bank of Pre-Defined PFs Embedded in ANNs," in *2019 IEEE Electrical Power and Energy Conference (EPEC)*, Montréal, QC, 2019, pp. 1-7.

23. A. R. Al-Roomi and M. E. El-Hawary, "Novel Highly Precise Power Loss Estimators That Directly Solve Power Balance Equality Constraints," in *2019 IEEE Electrical Power and Energy Conference (EPEC)*, Montréal, QC, 2019, pp. 1-8.

Accepted:

24. A. R. Al-Roomi, and M. E. El-Hawary, "M-Model: A New Precise Medium-Length Transmission Line Model," in *2020 IEEE 33rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, London, ON, 2020, pp. 1-4.

25. A. R. Al-Roomi, and M. E. El-Hawary, "Diagnosing Fuel Pumps, Power Transducers, CTs, and PTs via Fuel-Power Function and 2Oo3 Voting," in *2020 IEEE 33rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, London, ON, 2020, pp. 1-4.

26. A. R. Al-Roomi, and M. E. El-Hawary, "Mathematical Schemes to Linearize Operating Times of Overcurrent Relays by Sequentially Fixing Plug Settings and Time Multiplier Settings," in *2020 IEEE 33rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, London, ON, 2020, pp. 1-4.

27. A. R. Al-Roomi, and M. E. El-Hawary, "Hybridizing UFO with Other ML Tools to Locate Faults by Just Knowing Relay Operating Times," in *2020 IEEE 33rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, London, ON, 2020, pp. 1-4.

Completed/Submitted:

28. A. R. Al-Roomi and M. E. El-Hawary, "Locating Distribution Faults by Distance-Time Characteristic Curves Suggested for Directional Overcurrent Relays Using Interpolation- and Regression-Based Models," *Submitted*.

29. A. R. Al-Roomi and M. E. El-Hawary, "Universal Functions Originator," *Submitted*.

30. A. R. Al-Roomi and M. E. El-Hawary, "Realistic Economic Load Dispatch Model," *Submitted*.

31. A. R. Al-Roomi and M. E. El-Hawary, "Optimization/Modeling-Free Economic Load Dispatcher," *Submitted*.

32. A. R. Al-Roomi and M. E. El-Hawary, "Fast AI-Based Power Flow Analysis for High-Dimensional Electric Networks," *Completed*.

33. A. R. Al-Roomi and M. E. El-Hawary, "Finding the Best Time-Current Characteristic Curve for Optimizing the Coordination of Directional Overcurrent Relays Using BBO Algorithms," *Submitted*.

34. A. R. Al-Roomi and M. E. El-Hawary, "Optimal Coordination of Directional Overcurrent Relays Using BBO When Electromechanical, Static, Digital, and Numerical Relays All Exist," *Submitted*.

35. A. R. Al-Roomi and M. E. El-Hawary, "Effects of Surrounding Weather, Cable Design, Load Condition and System Frequency on Distributed Parameters of Transmission Lines," *Completed*.

36. A. R. Al-Roomi and M. E. El-Hawary, "A Novel Approach to Precisely Calculate Lumped Parameters for Transmission Lines with Sag Using the M-Model Equivalent Circuit," *Completed*.

37. A. R. Al-Roomi and M. E. El-Hawary, "How to Estimate Temperature Coefficients of Series and Shunt Parameters of Transmission Lines with Sag," *Completed*.

38. A. R. Al-Roomi and M. E. El-Hawary, "Peak-Load Forecasting of Nova Scotia During Winter Using Support Vector Machine with Optimally Configured Hyperparameters," *Completed*.

39. A. R. Al-Roomi and M. E. El-Hawary, "Generating Many Distance-Time Characteristic Curves by UFO Computing System for Locating Faults in Interconnected Networks via Numerical Directional Overcurrent Relays," *Completed.*

Technical Reports:

40. Ali R. Al-Roomi, "Telegrapher's Equations: The Hidden Backbone of Electric Power Systems Analysis," pp. 1-108, 2016.

41. Ali R. Al-Roomi, "Fuzzy Linear Optimization," pp. 1-70, 2016.

42. Ali R. Al-Roomi, "Optimal Fuzzy Input Set of Fuzzy Relation Equations Using BBO Algorithm," pp. 1-29, 2016.

43. Ali R. Al-Roomi, "Benchmark Functions Catalog−Part I: Unconstrained Single-Objective Optimization," pp. 1-73, 2016.

44. Ali R. Al-Roomi, "A Catalog of IEEE Optimal Relay Coordination Test Systems," pp. 1-71, 2017.

45. Ali R. Al-Roomi, "A Catalog of IEEE Economic Load Dispatch Test Systems," pp. 1-30, 2016.

Add to that, many uncompleted and pending studies, which are a product of this Ph.D. journey, will rise into the light soon.

1.5 Dissertation Outline

This dissertation is organized into eleven chapters and eighteen appendices as follows:

Chapter 1 This chapter covers our motivations, the goals from the dissertation, the main contributions, the list of publications, and the dissertation outline.

Chapter 2 This chapter presents the innovative hybrid optimization algorithms used in the other chapters.

Chapter 3 This chapter covers the classical modeling of economic load dispatch problems and then applying one of the hybrid optimization algorithms designed earlier in the second chapter.

Chapter 4 This chapter covers the classical modeling of optimal relay coordination problems and then applying one of the hybrid optimization algorithms designed earlier in the second chapter.

Chapter 5 This chapter hybridizes some meta-heuristic optimization algorithms with some machine learning tools to optimally forecast the consumption of energy and power based on some datasets taken from the province of Nova Scotia.

Chapter 6 This chapter presents a new precise lumped-parameter transmission line model and then compares its performance against all the standard models. It also outlines some possible highly precise models by hybridizing both the distributed-parameter and lumped-parameter models. Further, this chapter realizes the effects of the surrounding weather conditions (temperature, relative humidity, wind speed and direction, etc), system frequency, load current, and cable design/status/age. Moreover, different innovative techniques are presented to estimate the distributed series and shunt parameters of transmission lines during sag. Finally, this chapter shows the first trial to estimate many temperature-coefficients of transmission lines, which are not covered in the literature yet.

Chapter 7 This chapter realizes the existing economic load dispatch models by applying the weather and frequency effects covered in the sixth chapter.

Chapter 8 This chapter realizes the existing optimal relay coordination models by applying the weather and frequency effects covered in the sixth chapter. This chapter also considers the relay technology and DPRS. Further, it investigates the violation of the selectivity criterion, which is an inequality constraint, when only some specific fault points (i.e., near-end, middle, and far-end points) are considered.

Chapter 9 This chapter presents some of our innovative studies conducted in electric power systems, which are derived from the realization and integration phases done in the preceding chapters. It includes the following studies: 1. power flow quantities estimation using ANNs, 2. optimization-free economic load dispatcher, 3. optimization/modeling-free economic load dispatcher, 4. developing new transformation-based linear fuel-cost models to compete with quadratic and cubic models, 5. diagnosing fuel pumps, power transducers, current transformers (CTs), and potential transformers (PTs) via fuel-power function and "2 out of 3 (2oo3)" voting logic, 6. possible approaches to trade non-electric energy sources

in the next generation smart grids, 7. using linear heat sensors as very cheap alternative to existing electric power protective devices, and 8. locating faults in mesh networks by distance-time characteristic curves designed for directional overcurrent relays using interpolation- and regression-based models.

Chapter 10 This chapter introduces new machine learning computing systems that have lots of applications in many fields; including electric power systems engineering. It shows how they can be designed by merging some concepts taken from different disciplines. Also, the chapter shows how these new computing systems can compromise between the accuracy/precision and the explainability/interpretability performance criteria. The last part of the chapter applies these novel computing systems to solve some power prediction and fault location problems.

Chapter 11 This chapter concludes our dissertation and lists some suggestions for future work.

Appendix A This appendix shows how to linearize the objective function of ORC problems by fixing the time multiplier setting of DOCRs in the IEC/BS model.

Appendix B This appendix shows how to linearize the objective function of ORC problems by fixing the time multiplier setting of DOCRs in the ANSI/IEEE model.

Appendix C This appendix shows how to find the number of generations assigned to each stage of our multi-start RSA.

Appendix D This appendix covers the derivation of the short-length transmission line model.

Appendix E This appendix covers the derivation of the medium-length transmission line model when it is represented by the nominal Γ-circuit.

Appendix F This appendix covers the derivation of the medium-length transmission line model when it is represented by the nominal Ƭ-circuit.

Appendix G This appendix covers the derivation of the medium-length transmission line model when it is represented by the nominal T-circuit.

Appendix H This appendix covers the derivation of the medium-length transmission line model when it is represented by the nominal Π-circuit.

Appendix I This appendix covers the derivation of the medium-length transmission line model when it is represented by the nominal M-circuit.

Appendix J This appendix covers the derivation of the long-length transmission line model when each distributed section is represented by the nominal Π-circuit.

Appendix K This appendix derives the equivalent T-model based on the ABCD parameters of the long-length transmission line model.

Appendix L This appendix derives the equivalent Π-model based on the ABCD parameters of the long-length transmission line model.

Appendix M This appendix derives the equivalent M-model based on the ABCD parameters of the long-length transmission line model.

Appendix N This appendix derives the equation of the permeability of air.

Appendix O This appendix derives the equations of the incident voltage and current and the reflected voltage and current in the time-domain.

Appendix P This appendix covers the derivation of the M-model for sag transmission lines.

Appendix Q This appendix shows the full data of the IEEE 42-bus test system (IEEE Std. 399-1997).

Appendix R This appendix shows the full data of the turbo-generator used with the 2oo3 voting logic.

Appendix S This appendix lists the pseudocodes of all the algorithms used in the dissertation.

Chapter 2

Developing Innovative Hybrid Optimization Algorithms

2.1 Overview

The term **mathematical optimization**, or just **optimization**, is frequently heard in mathematics, computer science, engineering and even in economic and management science. Also, it can be found in proceedings, journals, books, encyclopedias, websites, etc, under different sections and names, like **soft computing**, **applied mathematics and optimization**, **evolutionary computation**, **numerical analysis**, etc.

From the basic of mathematics, suppose that there is a function f and it changes as the **independent variable** x changes, then f becomes the **dependent variable** of x and known shortly as $f(x)$.

Based on the system requirements, or in another word the **objective function**, the best solution to such a problem is called the **optimum** (or **optimal**) **solution**. This solution is located at a specified value of the **design variable** "x". The **optima** could be either **maxima** or **minima**, and the tool used to find this point is called an **optimization algorithm**. Figure 2.1 graphically represents the optimized solution of an arbitrary minimization problem.

When dealing with optimization, there are so many types of algorithms. In general, they can be classified into three main categories:

- Classical "Traditional" Optimization Algorithms
- Modern "Non-Traditional" Optimization Algorithms
- Hybrid Optimization Algorithms

The last category contains algorithms that are constructed by combining or merging multi-algorithms into one final algorithm. The algorithms of these combinations

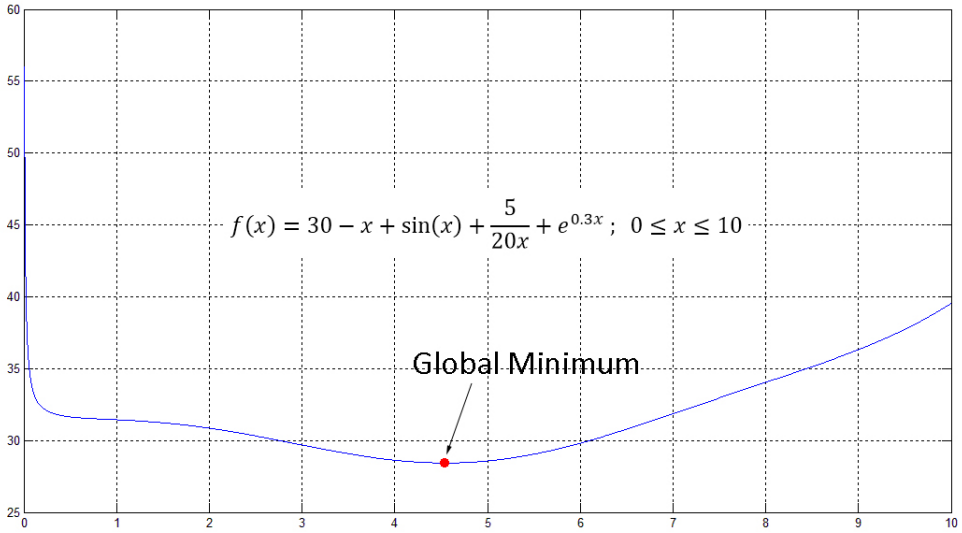


Figure 2.1: Minimization Problem with the Optimum Point

could be taken from one category as well as from different categories. The main reason behind going to the hybrid approach is to accumulate the strengthes of different techniques and at the same time to prevent, or at least to reduce, the associated weaknesses of each technique[1].

The general optimal design formulation of any problem can be depicted by the flowchart shown in Figure 2.2 [107], which is aexplained in the following subsections:

2.1.1 Design Variables

These variables are also known as **decision variables** or **solution features**, which are the independent variables of the optimization problem. If the function f depends on two or more variables, then it can be mathematically expressed as $f(x_1, x_2, \cdots, x_n)$. The subscript n denotes the total number of independent variables, and it is called the **dimension** of the problem. In optimization, if the given problem contains n decision variables, then it can be expressed in vector notation as $f(X)$ where

[1]Also, if a system problem needs an algorithm having some abilities of self-learning, logical thinking and decision making, then the techniques built based on **artificial intelligence (AI)** are the correct choice here. For example, finding the optimum solution of a numerical problem is one task that can be assigned to **artificial neural networks** (**ANNs**). Thus, by enough training, ANNs can solve that problem smartly without expressing any mathematical model (i.e., **modeling-free techniques**).

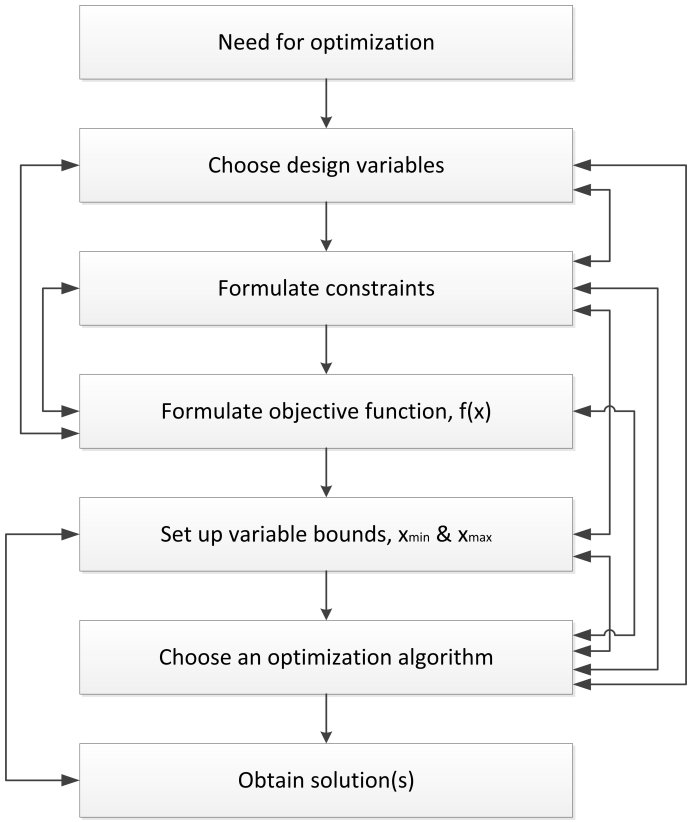


Figure 2.2: Flowchart of the Optimal Design Procedure

$X = [x_1, x_2, ..., x_i, ..., x_n]$ and $i = 1, 2, ..., n$. $[X]$ is called the set of design variables or simply the **design vector**, and x_i is the ith element of the design vector $[X]$ [107, 345, 374]. For more understanding, let's take the following simple functions:

$$f(x) = ax^2 + bx + c = 0 \tag{2.1}$$

$$f(x_1, x_2, x_3) = (a - x_1)^2 + (b - x_2)^2 + (c - x_3)^2 \geqslant 4.5 \tag{2.2}$$

$$f(t) = \frac{\sigma}{\sqrt{8 - t} + \sqrt{t - 1}}, \quad 1 \leqslant t \leqslant 8 \tag{2.3}$$

From these equations, the problem could contain just one design variable (i.e., ***one*-dimensional problem**) as in (2.1) and (2.3), or it could contain n design variables (i.e., ***n*-dimensional problem**) as in (2.2). The independent variables of these equations are: x for (2.1), $\{x_1, x_2, x_3\}$ for (2.2), and t for (2.3).

These design variables could be defined as *continuous*, *discrete*, *integer*, *mixed-integer* or even *binary*; which is a subset of discrete and integer types. The elements of the X vector are the unknown variables that need to be determined by solving the

given problem numerically (i.e., via using optimization algorithms). This can be done by ensuring that the variable f settled on its optimum value. Two important points have to be taken into account:

- The speed and efficiency of any proposed optimization algorithm significantly decrease as the problem dimension "n" increases. This phenomenon could be used as one performance criterion in the evaluation stage.
- The set of design variables must be linearly independent; i.e., since they are considered as independent variables, then each ith element x_i should not be affected by other $n-1$ elements of the design vector X.

2.1.2 Design Parameters

The "**parameters**" are fixed values or "**constants**" that have to be defined in the initialization stage of the optimization algorithm used in solving the given numerical problem. For example, by referring to the preceding three equations, the parameters are: $\{a, b, c\}$ for (2.1) and (2.2), and σ for (2.3).

Note that, from mathematics, the word "parameter" has a different meaning than the word "constant". If some variables are held constant, inactive, or depend on other external ineffective variables during optimizing the function, then they are treated as constants at some given conditions of those external variables. For example, if the parameters $\{a, b, c\}$ of (2.1) are varied, then a family or set of quadratic functions can be generated[2]. As a real example from physics, the weight of a body W is equal to its mass "m" multiplied by the gravity acceleration "g":

$$W(m) = m \times g \tag{2.4}$$

The standard gravity acceleration is $g = 9.80665 m/s^2$. It is taken as a constant value since all the simulations are done within the Earth's surface. But, if there is a significant difference in the altitude, then the value of g in (2.4) will definitely change too. To clarify it more, let's take the following arbitrary example:

$$f(x) = 3x + \frac{\beta(t)}{\sqrt{(x)}} \tag{2.5}$$

[2]This part is covered in fuzzy optimization where both the goals and parameters are fuzzified.

In (2.5), f depends only on x, but β is a function of t that does not have any effect on f if it is held constant. Thus, β has to be defined first at a predetermined value of t, then it can be treated as a constant in f.

In addition to these design parameters, there are other parameters that need to be defined before starting the optimization program, such as the total number of **iterations**[3], **tolerance**[4], etc.

2.1.3 Design Function

To solve any faced numerical problem, first it has to be expressed mathematically so the variability and behaviour of that problem can be translated into a meaningful and measurable format.

This mathematical model can be created arbitrary for virtual "non-real" problems, such as (2.1)-(2.3). Alternatively, that model can be created based on some data collected from records, readings, analysis, surveys, and/or inspections of a specific machine, system, factory, goods, market, etc, at different conditions and times. One of the sways to do that is to employ linear and nonlinear regression analysis. The other innovative ways will be presented later in Chapter 10.

All the required information about the design function, such as its objective (maximization or minimization mode) and the associated constrains (less than, equal to, greater than, etc), needs to be identified and then defined in the design function.

2.1.4 Objective Function(s)

It has been seen that f depends on the design vector X, and it varies with any change in any element of X. The variable f could be a *linear* or *nonlinear* function, in a *continuous* or *non-continuous* domain, and its value could be a *continuous*, *discrete*, *integer*, *mixed-integer* or even *binary*.

The term **objective function**, which is also known as **criterion** or **merit** [316], means finding the optimum value of f. Based the objective function, the term "optimum", as covered before, could mean minimum or maximum. For example, minimizing the cost of a specific product or minimizing the time consumed to manufacture a

[3] Also known as **loops** or, in modern optimization, as **generations**.
[4] Also known as the minimum acceptable error "ε" or the **early stopping criterion**.

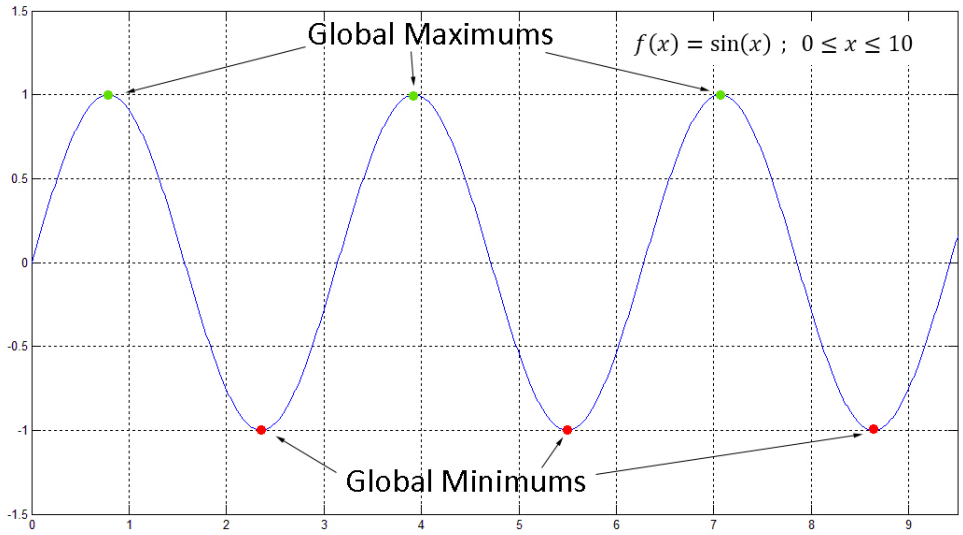


Figure 2.3: Multi-Optimum Points of a Single-Objective Function

product could be considered as the objective of such a problem. Also, that objective could be formulated as finding the maximum profit/revenue of a project suggested in a risky market or finding the maximum electric power delivered within different material characteristics of transmission lines. In general, the objective function is called the **cost function** when dealing with minimization problems and is called the **fitness** when dealing with maximization problems [345]:

$$\min_{x} f(x) \Rightarrow f(x) \text{ is called "objective" or "cost"}$$

$$\max_{x} f(x) \Rightarrow f(x) \text{ is called "objective" or "fitness"} \tag{2.6}$$

If a **single-objective function** has multi similar peaks and valleys, then its multi-optimum points mean either multi-minimum or multi-maximum points as illustrated in Figure 2.3. Thus, finding multi-optimum points does not mean finding a mixture of minimum and maximum points at the same time. Based on this, if the objective is to find the maximum value of a function, then the minimum value is considered as the worst solution, and vice versa if the objective is to minimize that function.

Suppose that the programming code of an optimization algorithm is designed to act as a minimizer. If the objective of a new design function is to find the maximum point, then the algorithm designer has to modify the structure of his/her code to

act as a maximizer. Practically, this approach is totally not preferred and impractical, simply because this primitive correction is a time-consuming approach and it requires unnecessary effort and new codes to accomplish that task; especially when that program was coded in the past without embedding enough comments or supporting documents from the programmer. Alternatively, the **duality principle** can solve this technical issue directly by just reforming the objective[5]. One possible way to do that is by adding a negative sign as [345, 374]:

$$\min_x f(x) \Leftrightarrow \max_x [-f(x)]$$

$$\max_x f(x) \Leftrightarrow \min_x [-f(x)] \tag{2.7}$$

Thus, the optimization problem can be switched easily from the maximization mode to the minimization mode, and vice versa, by just changing the sign. This is a very useful trick, especially when the researcher wants to find both the minimum and maximum points of f. That is, this trick can convert such optimization algorithms into general-purpose optimizers where the design function can be plugged-in externally by a user and its objective function can be defined via a separate command coded somewhere in the program.

2.1.5 Design Constraints

If f exists in all the points of X, then optimizing $f(X)$ becomes a relatively easy task. Such function is known as **unconstrained function**. If the objective is to minimize $f(X)$, then the formulation of the optimization problem can be expressed as:

$$\text{Find } X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{Bmatrix} \text{ which minimizes } f(X) \tag{2.8}$$

The large expression of (2.8) can be replaced by the following small expression:

$$\min_x f(x_1, x_2, \cdots, x_n) \tag{2.9}$$

[5]This is valid with modern optimization algorithms. In linear programming, the duality approach is not an easy task.

Unfortunately, most of the problems faced in real-life applications have many obstacles and restrictions called **design constraints**. They could exist on f as **behavior**[6], or could exist on each element of $[X]$ as **side constraints** [316]. The functional constraints can be further classified into:

- **Equality Constraints**
- **Inequality Constraints**

Since the side constraints are assigned to the design features (independent variables), so they are defined as just constant values (lower and upper limits). On the opposite side, the functional constraints could be defined as linear or nonlinear equations [246].

If all these design constraints are satisfied, then the obtained solution becomes useful, and thus it can be effectively utilized and implemented. Such solution is called a **feasible solution**. If any one of these constraints is **violated**, then that solution is called an **infeasible solution**. It is a worthless solution[7], which might not have any logical representation[8].

2.1.5.1 Equality Constraints

These constraints force the solution to be equal to only one value, which is equal to zero in (2.1). If (2.1), for instance, equals 30 instead of zero, then this right-hand side value should be subtracted in the left-hand side for getting a standard equality constraint format as follows:

$$f(x) = ax^2 + bx + c = 30 \tag{2.10}$$

$$f(x) = ax^2 + bx + c - 30 = 0 \tag{2.11}$$

If (2.1) or (2.11) is optimized, then the numerical solution should approach the analytical one. Unfortunately, satisfying equality constraint is a very hard task and

[6] Also called **functional constraints**.

[7] This statement is applied in conventional optimization. In fuzzy optimization, the violated constraints and parameters are accepted with a certain degree, as well be seen later.

[8] To be more precise, these infeasible solutions could contain good information that can guide the optimization algorithm to settle on the optimum value. This is the main reason why some **constraint-handling techniques** have better performance than others.

it needs a large **number of function evaluations** (**NFE**)[9] in the optimization algorithm, and the desired answer may not be reached at all. The reason behind this is that when the equality constraint is assigned to a function, then the algorithm will accept the obtained solution as a feasible solution if and only if the value of the function is equal to the value of the equality constraint; it is zero in the standard format.

Practically, if the equality constraints exist in the design function, then they could be satisfied by a certain amount of tolerance ($\pm\ \varepsilon$) rather than setting them to zero. This approach can save a significant amount of CPU time and it can avoid getting infinite loops if the algorithm stopping criterion is activated with zero value.

The symbol h is frequently used in references to represent one equality constraint. If l equality constraints exist in the design function, then they could be represented as $[h_1, h_2, \cdots, h_l]$, or $h_p:\ p = 1, 2, \cdots, l$, or simply by using the vector notation $[H]$.

It is important to check if these l equality constraints are linearly independent or not. If not, then the design function has a nonlogical expression and thus the entire model needs to be reformulated again. If the given problem has a vector $[X]$ with a length n and a vector $[H]$ with a length l, then the optimization problem can be mathematically expressed as follows:

$$\min_{x}\ f(x_1, x_2, \cdots, x_n)$$

Subjected to:

$$\begin{aligned} h_1(x_1, x_2, \cdots, x_n) &= 0 \\ h_2(x_1, x_2, \cdots, x_n) &= 0 \\ h_l(x_1, x_2, \cdots, x_n) &= 0 \end{aligned} \tag{2.12}$$

By referring to linear algebra, if (2.12) is a linear model, then there are three possibilities [374]:

- $n > l \ \rightarrow$ *under-determined case*: the problem has many solutions and thus the optimization technique is applicable.
- $n = l \ \rightarrow$ *critical case*: the problem has one unique solution, so it is a solvable problem.

[9] NFE = population size × number of iterations × number of evaluations.

- $n < l \rightarrow$ *over-determined case*: the problem has no solution, so the design function needs some corrections.

2.1.5.2 Inequality Constraints

Rather than the previous very intensive type of constraints that accepts only one solution as a feasible solution, this inequality constraint is more flexible. It requires fewer efforts from optimization algorithms, and it can be satisfied by many possible feasible vectors of X.

Let's take (2.2) as an example. If the objective here is to minimize $f(X)$, then the smallest point is zero. This point occurs when the three design variables equal their corresponding parameters $\{x_1 = a, x_2 = b, x_3 = c\}$. But, because of its inequality constraint, the answer zero is considered as an infeasible solution. That is, the feasible solutions start when $f(X)$ equals 4.5 or above. If (2.2) is subjected to an equality constraint with zero on the right-hand side (i.e., the condition $f(X) = 0$ must be satisfied), then the same difficulty faced with (2.1) will be faced here again.

Similar to the previous constraint, this type is represented by the symbol g. If the design function contains m inequality constraints, then they can be represented as $[g_1, g_2, \cdots, g_m]$, or $g_q : \ q = 1, 2, \cdots, m$, or simply by using the vector notation $[G]$.

In mathematics, the sign that represents equality constraints is limited to ($=$ "equal to"). The sign ($\neq$ "not equal to") means that the solution obtained for f should not equal to a predefined value. However, this sign does not give any additional information. For example, it does not show whether the solution is greater or less than the predetermined value. Therefore, four possible signs could be used to represent inequality constraints, which are classified into two main groups:

- **Strict inequalities**
  - $g_q(X) > \alpha$: $g_q(X)$ is **greater than** α
  - $g_q(X) < \beta$: $g_q(X)$ is **less than** β
- **Not-strict inequalities**
  - $g_q(X) \geqslant \alpha$: $g_q(X)$ is **greater than or equal to** α (also known as: **not less than** α or **at least** α)

– $g_q(X) \leqslant \beta$: $g_q(X)$ is **less than or equal to** β (also known as: **not greater than** β or **at most** β)

For the strict types, it is difficult to determine the endpoints (minimal and maximal points) of $g_q(X)$ to satisfy $(>)$ and $(<)$, respectively. The reason is that the boundary of α and β cannot be reached (i.e., open intervals). For example, if $(\alpha = 3.0)$ then the condition $(g_q(X) > \alpha)$ is satisfied by infinite solutions $\{g_q(X) = 3 + 10^{-c} : c = 1, 2, \cdots, \infty\}$. It shows that $g_q(X) \downarrow$ as $c \uparrow$, but it cannot reach the minimal point. Based on this, the non-strict types are used instead to provide endpoints, which are easy to be defined and coded in any numerical programming language.

To have the standard inequality constraint format, then only the sign "$\leqslant$" must be used with zero on the right-hand side. Thus, (2.2) has to be reformulated as follows:

$$f(x_1, x_2, x_3) = 4.5 - \left[(a - x_1)^2 + (b - x_2)^2 + (c - x_3)^2\right] \leqslant 0 \tag{2.13}$$

Thus, for $n-$dimensional problems given with m inequality constraints, they can be mathematically expressed as follows:

$$\min_x f(x_1, x_2, \cdots, x_n)$$

Subjected to:

$$\begin{aligned} g_1(x_1, x_2, \cdots, x_n) &\leqslant 0 \\ g_2(x_1, x_2, \cdots, x_n) &\leqslant 0 \\ g_m(x_1, x_2, \cdots, x_n) &\leqslant 0 \end{aligned} \tag{2.14}$$

2.1.5.3 Side Constraints

This type of constraints has many other names, like **domain**, **solution space**, **search space**, **variable bounds**, **choice set**, **feasible region**, **bound constraints**, etc [107, 272, 316]. From its name, this type of constraints is associated with the design features where each element of $[X]$ has two bounds called the lower and upper limits or bounds. Due to the same mathematical representation difficulties of the inequality constraints, the closed intervals are used here to define the side constraints in the design function as $x_i \in \left[x_i^{\min}, x_i^{\max}\right] : \ i = 1, 2, \cdots, n$, or as $x^{\min} \leqslant x_i \leqslant x^{\max} : \ i = 1, 2, \cdots, n$. Using the vector notation, it can be expressed as $X \in \left[X^{\min}, X^{\max}\right]$ or as $X^{\min} \leqslant X \leqslant X^{\max}$.

As a rule of thumb, the optimization algorithm performs better as the span between the lower and upper bounds decreases. The reason behind this phenomenon is that the algorithm needs less effort to search for the optimal solution within a very narrow domain. It is like making a zoom-in into a small spot of a bigger search space, and thus this cropped domain becomes very rich in good solutions.

Although the side constraints are classified as one type of design constraints[10], a problem containing only side constraints is considered as an unconstrained problem. The reason behind this is that, by nature, the logical problem has to be designed with side constraints so that the algorithm can search for the optimality within a specific search space. Suppose that the given problem is very complex and cannot be depicted or solved analytically and its search space is open (*i.e.*, $X^{-\infty} \leqslant X \leqslant X^{\infty}$), then no one can determine the location of the optima with this infinite domain. Thus, for $n-$dimensional optimization problems, they can be expressed as follows:

$$\min_{x} f(x_1, x_2, \cdots, x_n)$$

Subjected to:

$$x_i^{\min} \leqslant x_i \leqslant x_i^{\max} \quad i = 1, 2, \cdots, n \tag{2.15}$$

2.1.6 General Principles

Before describing the final standard format of optimization problems when all the design constraints are assigned, it is important to cover the following general principles:

2.1.6.1 Feasible Space vs. Search Space

From the preceding design constraints, it has been seen that the obtained optimum solution becomes useable only if it is feasible. The space of the design variables could be open to infinity, or could be bounded between two limits. Moreover, even with these side constraints, the feasibility also depends on some equality and inequality constraints that have to be satisfied as well. These classifications create three different layers on the entire space of any constrained optimization problem:

- **Infinite space:** lower and upper limits are open to infinity, $X \in [X^{-\infty}, X^{\infty}]$

[10] Except for some references, such as [272], where unconstrained optimization problems are called bound-constrained optimization problems.

- **Search space:** lower and upper limits are bounded, $X \in \left[X^{\min}, X^{\max}\right]$
- **Feasible space:** lower and upper limits are bounded and the functional constraints are satisfied, $X \in [X^{-\infty}, X^{\infty}]\,,\ [h(X)]_l = 0,$ and $[g(X)]_m \leqslant 0$

The difference between these three layers is graphically described in Figure 2.4. Thus, the feasible space must be inside the search space; i.e., part of the search space. Otherwise, the solution is considered infeasible. Also, as the number of equality and inequality constraints increases, the feasible space is shrunk more and more, and thus the optimum solution becomes very hard to be found.

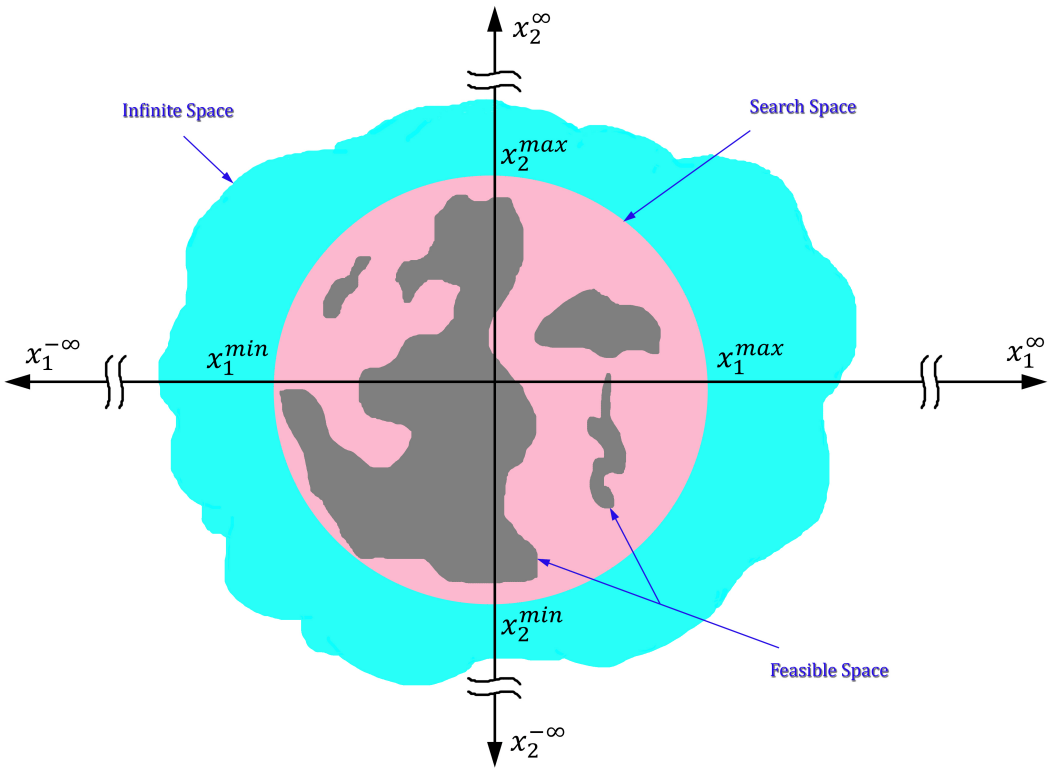


Figure 2.4: Infinite Space vs. Search Space vs. Feasible Space

2.1.6.2 Global Optimum vs. Local Optimum

It has been known that the definition of the optimum solution is either minimum or maximum solution. It could appear as a single optimum or multi-optimum solutions, as seen before in Figure 2.3. If multi-optimum solutions are not identical, then the most optimum solution of these points is called a **global optimum solution**, while

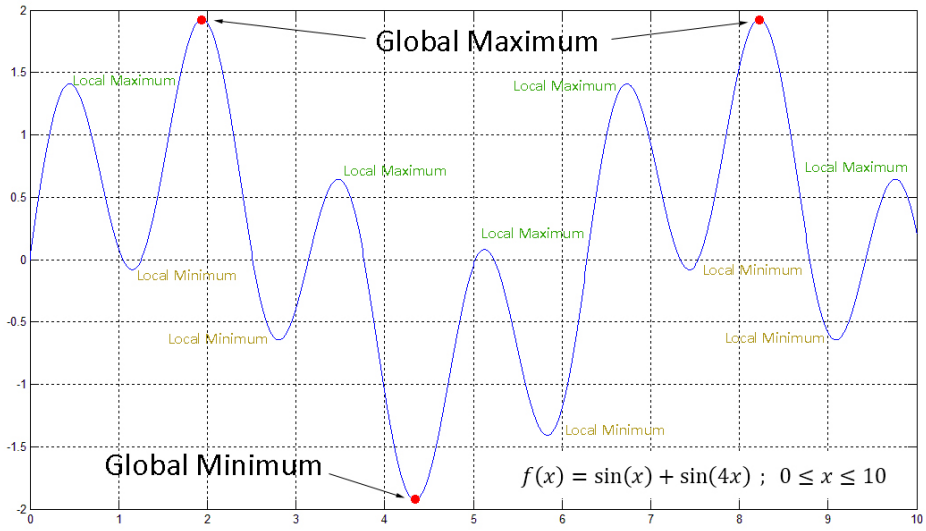


Figure 2.5: Global Optimum vs. Local Optimum

the others are called **local optimum solutions**. Optimization problems could have groups of multi-global and multi-local optimum solutions. These local optima are considered as **traps** where the efficient optimization algorithm has the ability to escape from these traps and settle on or close to the global optimum solution(s) quickly and smartly. Figure 2.5 shows a function that has a mixture of local and global optimum solutions. Again, if the objective is to minimize f, then there will be only one global optimum solution where the two global maximum points are considered as the worst solutions, and vice versa if the objective is to maximize f.

If the global optimum solution is analytically[11] predefined, then this solution is denoted by an asterisk. This symbol is assigned to both the dependent and independent variables; as $f_{\min}(X^*)$ and X^*, respectively.

2.1.6.3 Types of Problem

Based on the availability of objective(s) and/or constraint(s), the design problem could be one of the four possible types, as summarized in Table 2.1. For more details, please refer to [122, 220].

[11] Sometimes, the global optima can be numerically found with almost, or even exact, zero error.

Table 2.1: Types of Problem

	Objective Function	
Constraints	Yes	No
Yes	Constrained Optimization Problem	Constraint Satisfaction Problem
No	Free Optimization Problem	No problem

2.1.7 Standard Format

By taking into account all the preceding issues, any design function can be transformed into the following standard mathematical model [374]:

$$\min_{x} f(x_1, x_2, \cdots, x_n)$$

$$\begin{aligned}
\text{Subjected to: } h_1(x_1, x_2, \cdots, x_n) &= 0 \\
h_2(x_1, x_2, \cdots, x_n) &= 0 \\
h_l(x_1, x_2, \cdots, x_n) &= 0 \\
g_1(x_1, x_2, \cdots, x_n) &\leqslant 0 \\
g_2(x_1, x_2, \cdots, x_n) &\leqslant 0 \\
g_m(x_1, x_2, \cdots, x_n) &\leqslant 0
\end{aligned}$$

$$\begin{aligned}
x_1^{\min} \leqslant x_1 \leqslant x_1^{\max} \\
x_2^{\min} \leqslant x_2 \leqslant x_2^{\max} \\
x_n^{\min} \leqslant x_n \leqslant x_n^{\max}
\end{aligned}$$

$$\text{where: } f_{\min}(X^*) = a \quad , \quad x_i^* = \{s_1, s_2, \cdots, s_n\} \tag{2.16}$$

Also, it can be expressed as:

$$\min_{x} f(x_1, x_2, \cdots, x_n)$$

$$\begin{aligned}
\text{Subjected to: } h_p(x_1, x_2, \cdots, x_n) &= 0 \quad , \quad p = 1, 2, \cdots, l \\
g_q(x_1, x_2, \cdots, x_n) &\leqslant 0 \quad , \quad q = 1, 2, \cdots, m
\end{aligned}$$

$$x_i^{\min} \leqslant x_i \leqslant x_i^{\max} \quad , \quad i = 1, 2, \cdots, n$$
$$\text{where: } f_{\min}(X^*) = a \quad , \quad x_i^* = s_i \tag{2.17}$$

Or, if the vector notation form is employed, then it can be expressed as:

$$\min_x f(X), \ [X]_n$$

$$\begin{aligned} \text{Subjected to: } [h(X)]_p &= 0 \quad , \quad p = 1, 2, \cdots, l \\ [g(X)]_q &\leqslant 0 \quad , \quad q = 1, 2, \cdots, m \end{aligned}$$

$$X^{\min} \leqslant X \leqslant X^{\max}$$
$$\text{where: } f_{\min}(X^*) = a \quad , \quad X^* = S \tag{2.18}$$

2.1.8 Constraint-Handling Techniques

The side constraints can be easily satisfied by controlling the independent variables to be within the decided bounds. The following equation can be coded in any numerical programming language to generate random values of each ith design variable:

$$x_i = x_i^{\min} + \text{rand}\left(x_i^{\max} - x_i^{\min}\right) \tag{2.19}$$

where rand is a function that generates uniformly distributed random numbers in the interval between 0 and 1.

It is clear that the variable x_i is bounded between $x_i^{\min}$ and $x_i^{\max}$, which are respectively reached when rand = 0 and 1. Unfortunately, this direct solution approach does not work with functional constraints (i.e., equality and inequality). Rather, they require more complicated constraint-handling techniques. Selecting the correct type is a very important step because the algorithm speed and accuracy can be markedly affected by inefficient techniques [122]. Figure 2.6 summarizes the most popular constraint-handling techniques [107, 122, 199, 316, 345, 374, 390, 392].

The **penalty functions** are often used because the other approaches are hard to be modeled or/and need derivatives [122]. Besides, the **exterior penalty function** (**EPF**) is preferred in constrained EAs. The reason is that the **interior penalty function** (**IPF**) requires feasible individuals which in turn complicate the solution. In this subsection, a brief overview of some EPFs is given. Also, the **classical random search method**, which is classified as one of the **direct search methods**, is covered.

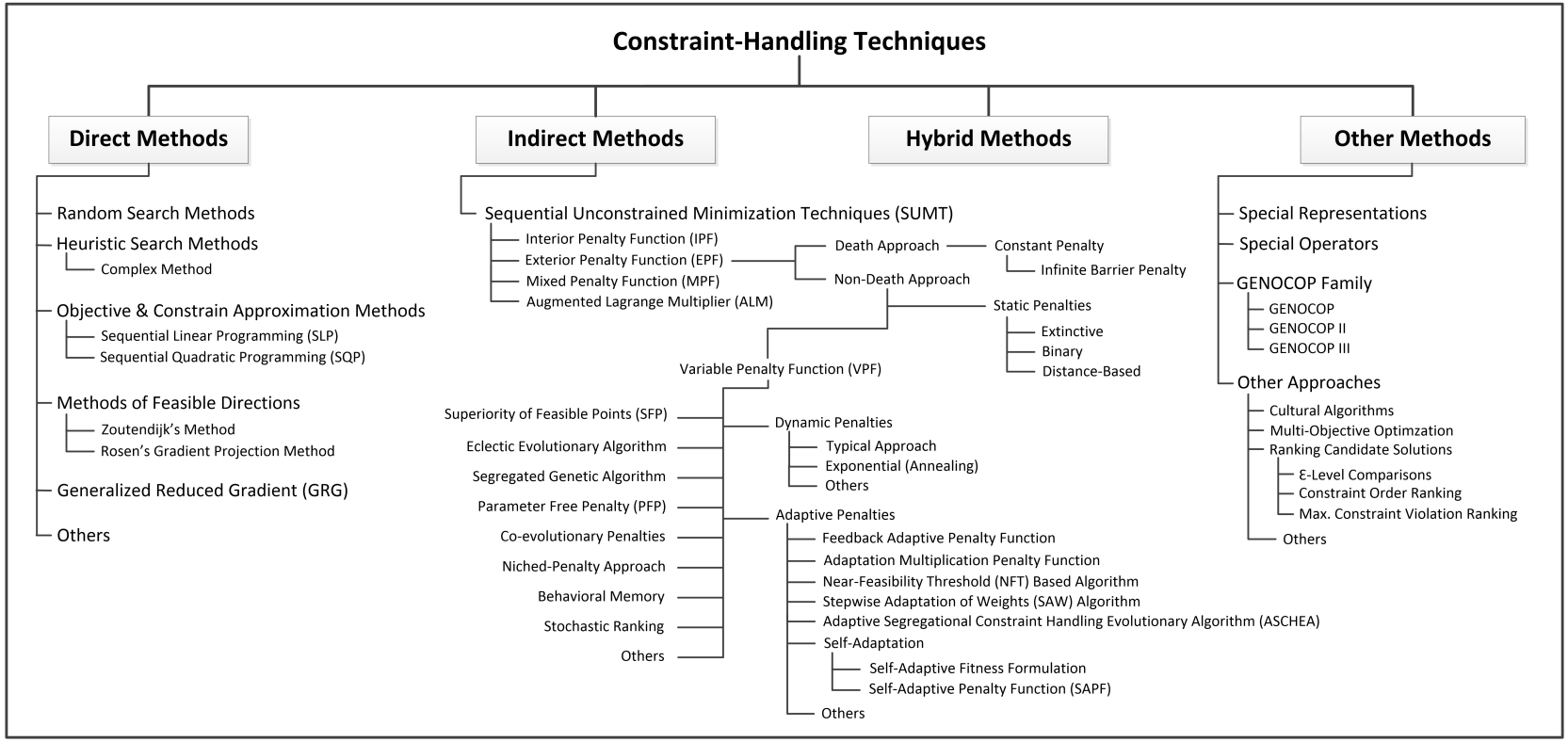


Figure 2.6: Summary of the Most Popular Constraint-Handling Techniques

By referring to (2.16)-(2.18), EPFs can easily transform them into unconstrained optimization problems by employing either the **additive** or the **multiplicative** approach. These two approaches are respectively described as follows [390]:

$$\min_X \phi(X), \quad \text{where} \quad \phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ f(X) + P(X), & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.20}$$

$$\min_X \phi(X), \quad \text{where} \quad \phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ f(X) \times P(X), & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.21}$$

where $P(X)$ is called the **penalty term**, which is equal to zero for feasible individuals ($X \in \mathcal{F}$) and be a positive value in case there is a violation of any constraint ($X \notin \mathcal{F}$). Thus, for minimization mode, the **penalized cost function** $\phi(X)$ becomes higher than its actual value $f(X)$. This $P(X)$ can be provided in different forms based on the type of penalty function employed. The most common form is:

$$\begin{aligned} P(X) &= \sum_{q=1}^{m} r_q \; \hat{g}_q(X) + \sum_{p=1}^{l} d_p \; \hat{h}_p(X) \\ \text{where: } \hat{g}_q(X) &= [\max(0, g_q(X))]^{\beta} \\ \hat{h}_p(X) &= |h_p(X)|^{\gamma} \end{aligned} \tag{2.22}$$

where r_q and d_p are called the **penalty multipliers**. The coefficients β and γ are user-defined positive constants, which are commonly set equal to either 1 or 2 [122,316,345].

Because $[H]_l = 0$ cannot be easily satisfied, so an acceptable tolerance ($\pm\ \varepsilon$) is adopted instead of crisp zero. Thus, the pth equality constrain is satisfied if:

$$-\varepsilon \leqslant h_p(X) \leqslant \varepsilon \quad , \quad p = 1, 2, \cdots, l \tag{2.23}$$

This equation can be split into two inequality constraints, which can be expressed using the standard format as follows:

$$\begin{aligned} h_p(X) - \varepsilon &\leqslant 0 \\ -h_p(X) - \varepsilon &\leqslant 0 \end{aligned} \tag{2.24}$$

Therefore, by using (2.24), (2.22) can be modified to be:

$$\begin{aligned} P(X) &= \sum_{q=1}^{m+l} r_q \; \hat{g}_q(X) \quad ; \quad \text{where} \\ \hat{g}_q(X) &= \begin{cases} [\max(0, g_q(X))]^{\beta}, & \text{for } q \in [1, m] \\ [\max(0, |h_q(X)| - \varepsilon)]^{\beta}, & \text{for } q \in [m+1, m+l] \end{cases} \end{aligned} \tag{2.25}$$

Before describing the popular exterior penalty functions, it is important to mention that the penalized cost function "$\phi(X)$" does not apply to the random search method. This classical direct search method depends only on an internal *while*-loop to continue generating random elements of the design vector $[X]$ until satisfying all the design constraints. These ten constraint-handling techniques are briefly described in the following lines:

2.1.8.1 Random Search Method

This constraint-handling technique is very simple, and it can be easily programmed by following the steps of the pseudocode in Algorithm 1.

Algorithm 1 Rebuild Infeasible Individuals Directly via the Random Search Algorithm (RSA)

Require: all the constraint values of each individual as a vector C

1: **for** $j \leftarrow 1$ *to* N **do** {where N = population size}
2: **while** any element of C_j is violated **do** {there are N vectors of C}
3: Randomly generate new design vector X
4: Determine new C_j
5: **end while**
6: **end for**

Unfortunately, this technique is not efficient and consumes high CPU time as the problem complexity increases [316]. This complexity could be faced in different locations of the problem. For example, it could be the dimension of the problem, its type (convex or non-convex, explicit or implicit, unimodal or multimodal, etc), type of design variables (continuous, discrete, mixed-integer, binary, etc), number of

functional constraints, type of functional constraints (equality or inequality), lower and upper bounds of design variables, etc.

2.1.8.2 Constant Penalty Function

In this type, the penalty term $P(X)$ is set to a very high value if there is any violation of any constraint. During the function evaluation process, this intensive penalty function rejects all the infeasible individuals. This is why it is called the **death penalty** approach [316, 345].

The main drawback of this simple type is seen when the infeasible solutions are very close to the border of the feasible space. These individuals have some good information that could guide the optimization algorithm to reach the feasible space. However, this death EPF approach erases all these useful data. Thus, individuals having a few or many violations will be treated by the same rejection action. This means that all the infeasible individuals will completely disappear in the next generation [122].

The most popular sub-type is called the **infinite barrier penalty** [107, 390]. It comes in an additive form where $P(X)$ given in (2.25) is calculated as follows:

$$P(X) = \mathcal{R} \sum_{q=1}^{m+l} \hat{g}_q(X) \tag{2.26}$$

where the penalty multiplier "r_q" is replaced with a very large constant number "$\mathcal{R}$"; it is usually set equal to 10^{20}. From this equation, the penalty term will reach almost infinity (i.e., $P(X) \approx +\infty$) if any violation is detected. Some of death penalty functions are defined by only $\mathcal{R}$ term since $\mathcal{R} >> \sum_{q=1}^{m+l} \hat{g}_q(X)$. Thus, $P(X)$ is expressed as a high constant value, which is independent of the violation magnitudes.

2.1.8.3 Binary Static Penalty Function

It is an additive penalty. The distance function $\hat{g}_q(X)$ given in (2.25) can be expressed as a simple binary value, which equals 1 if the qth constraint is violated. Otherwise, it equals zero [122]. If r_q is taken as a constant value for all the functional constraints,

then the final expression of this penalty function can be formulated as follows:

$$P(X) = r \sum_{q=1}^{l+m} \hat{g}_q(X), \quad \hat{g}_q(X) = \begin{cases} 1, & \text{if } \hat{g}_q(X) \text{ is violated} \\ 0, & \text{if } \hat{g}_q(X) \text{ is satisfied} \end{cases} \tag{2.27}$$

Now, the dimension of the vector r becomes one[12], which is easy to be tuned. In contrast, the other static penalty functions, listed in Figure 2.6, are hard to be tuned. Besides, the binary static penalty can compromise between the speed and accuracy, which are very important winning factors in many engineering applications.

2.1.8.4 Superiority of Feasible Points (SFP) - Type I

In this method, the penalized cost function given in (2.20) is expressed as follows:

$$\begin{aligned} \min_X \phi^{'}(X) &= \phi(X) + \theta(X) \\ \text{where: } \theta(X) &= \begin{cases} 0, & \text{if } \mathcal{F} = \emptyset \text{ or } X \in \mathcal{F} \\ \alpha, & \text{if } \mathcal{F} \neq \emptyset \text{ and } X \notin \mathcal{F} \end{cases} \end{aligned} \tag{2.28}$$

The value of α is the largest feasible individual:

$$\alpha = \max\left[f(Y)\right] \ : \ Y \in \mathcal{F} \tag{2.29}$$

This is done to ensure that the bad feasible individual is better, or at least, equal to the best infeasible individual [390]. The penalty multiplier r_q given in (2.25) can be set as a fixed number [257]. In this dissertation, r_q is set equal to 10.

2.1.8.5 Superiority of Feasible Points (SFP) - Type II

SFP-I does not cover the range when $f(X) < 0$, while SFP-II does. The second type has a similar expression, except for the value of α. Here, it is modified to be equal to the difference between the largest feasible and the smallest infeasible individuals as follows [199, 345]:

$$\alpha = \max\left[0, \max_{Y \in \mathcal{F}} f(Y) - \min_{Y \notin \mathcal{F}} \phi(Y)\right] \tag{2.30}$$

[12] i.e., it becomes a scalar value.

2.1.8.6 Eclectic Evolutionary Algorithm

The bad thing about the extinctive static penalty function shown in Figure 2.6 is that the solution quality is very sensitive to the values of the vector "r_q" [390]. The eclectic EA works in a similar principle of SFP types where the bad feasible individuals are considered to be better than the best infeasible individuals, but in a different way [345]. Instead of using how much the constraints are violated, it uses the number of violated constraints as a basis to penalize the cost function as follows:

$$\phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ K\left[1 - \dfrac{v(X)}{(m+l)}\right], & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.31}$$

where K is a very large positive constant (it is taken as $K = 1 \times 10^9$ [390]), $v(X)$ is the number of satisfied or non-violated constraints, and $(m + l)$ is the total number of functional constraints.

2.1.8.7 Typical Dynamic Penalty Function

If the soft counter, that counts the number of iterations or generations inside the optimization algorithm, is used to simulate the time "t", then the amount of the penalized value is proportional to the number of iterations. This action provides two things:

- In the beginning, the penalization process will forgive the infeasible individuals by penalizing them with small values. Thus, the optimization algorithm is given a chance to collect some useful information about the search space being explored.
- As the number of iterations increases, the penalization level increases. Thus, after collecting enough information about the problem, the penalty function will start forcing the infeasible individuals to go inside the feasible search space.

The penalized cost function is defined as [122, 316, 345]:

$$\phi(X) = f(X) + (c.t)^{\alpha} P(X) \tag{2.32}$$

where c and α are constants. For example, they can be taken as 0.5 and 2, respectively [345]. $P(X)$ is similar to that of (2.25), but with $r = 1$ and $\beta = 2$.

2.1.8.8 Exponential Dynamic Penalty Function

Instead of using the additive form, as in (2.32), this type of EPFs comes in a multiplicative form. The same assumptions are applied here, except that the penalty value grows in an exponential rate rather than a linear rate as in the typical dynamic approach. The new penalization process is defined as follows [122, 316, 345]:

$$\phi(X) = f(X) \times e^{\frac{P(X)}{\tau}} \tag{2.33}$$

where $\tau = \frac{1}{\sqrt{t}}$, which approaches zero as the number of iterations or generations approaches infinity (i.e., $t \to \infty \Rightarrow \tau \to 0 \Rightarrow \phi(X) \to \infty$).

This type of dynamic penalties is valid just for minimization where $f(X) \geqslant 0 \; \forall \; X$. Otherwise, the normalized version, as described in [345], has to be used instead.

2.1.8.9 Adaptive Multiplication Penalty Function

From its name, it comes in a multiplicative form. The cost function is penalized as follows [392]:

$$\phi(X) = f(X) \times \left\{ 1 + \frac{1}{(m+l)} \sum_{q=1}^{m+l} \left[\frac{\hat{g}_q(X)}{\hat{g}_q^{\max}(X)} \right]^{\alpha} \right\} \tag{2.34}$$

Actually, there are many adaptive approaches presented in the literature. The first one was proposed by [159]. After investigating the various types of adaptive EPFs, the adaptive multiplication approach is selected because it works based on the typical feedback and **near feasibility threshold** (**NFT**-based) approaches [390]. The original equation has a subtraction arithmetic operator, while (2.34) has an addition arithmetic operator. This adjustment is essential to make it functional in the minimization mode. The qth constraint $\hat{g}_q(X)$ can be calculated by using (2.25) with $\beta = 1$. The symbol $\hat{g}_q^{\max}(X)$ denotes the biggest functional constraint, which can be obtained as follows:

$$\hat{g}_q^{\max}(X) = \max\left[\varepsilon, \max\left(\hat{g}_q(X)\right)\right] \tag{2.35}$$

The epsilon ε in (2.35) is very important to avoid dividing by zero when all the individuals are feasible. For example, ε can be set equal to 10^{-20}. Thus, if there is no infeasible individual, then the summation term becomes zero instead of getting an error due to dividing zero by zero.

2.1.8.10 Self-Adaptive Penalty Function (SAPF)

The SAPF algorithm works based on the following two conditions [345]:

- If the ratio of the feasible individuals to the entire **population size** N is low, then the penalized cost function "$\phi(X)$" has to be small for infeasible individuals having a few violations.
- If that ratio is high, then only infeasible individuals having low cost "$f(X)$" should be penalized with small penalty terms.

To identify the best individual in the current population, SAPF penalizes infeasible individuals by two terms, $d(X)$ and $P(X)$, as follows [350]:

$$\phi(X) = d(X) + P(X) \tag{2.36}$$

where $d(X)$ is called the distance value. SAPF needs special care for constructing (2.36). The steps can be summarized as follows [345, 350]:

- Firstly, normalize $f(X)$ of each individual as:

$$N(X) = \frac{f(X) - f_{\min}}{f_{\max} - f_{\min}} \tag{2.37}$$

  where $N(X) \in [0, 1] \; \forall \; X$; the best individual occurs when $N(X) = 0$, and vice versa when $N(X) = 1$.

- Secondly, compute the normalized violation magnitude of the qth constraint as:

$$M(X) = \frac{1}{m + l} \sum_{q=1}^{m+l} \left(\frac{\hat{g}_q(X)}{\hat{g}_q^{\max}(X)} \right) \tag{2.38}$$

  where $\hat{g}_q(X)$ and $\hat{g}_q^{\max}(X)$ can be obtained by using the expressions presented in the preceding penalty functions. Thus, $M(X) = 0 \; \forall \; X \in \mathcal{F}$ and $M(\bar{X}) > 0 \; \forall \; \bar{X} \notin \mathcal{F}$.

- Thirdly, for each individual, compute the distance $d(X)$ as:

$$d(X) = \begin{cases} M(X), & \text{if } \mathcal{F} = \emptyset \\ \sqrt{N^2(X) + M^2(X)}, & \text{if } \mathcal{F} \neq \emptyset \end{cases} \tag{2.39}$$

  where $(\mathcal{F} = \emptyset)$ means that all the population individuals are infeasible.

- Fourthly, the penalty term in (2.36) can be calculated as:

$$\begin{aligned} P(X) &= (1-r)Y_1(X) + rY_2(X) \\ \text{where: } Y_1(X) &= \begin{cases} 0, & \text{if } \mathcal{F} = \emptyset \\ M(X), & \text{if } \mathcal{F} \neq \emptyset \end{cases} \\ Y_2(X) &= \begin{cases} 0, & \text{if } X \in \mathcal{F} \\ N(X), & \text{if } X \notin \mathcal{F} \end{cases} \\ r &= \frac{number\ of\ feasible\ individuals}{population\ size} \end{aligned} \tag{2.40}$$

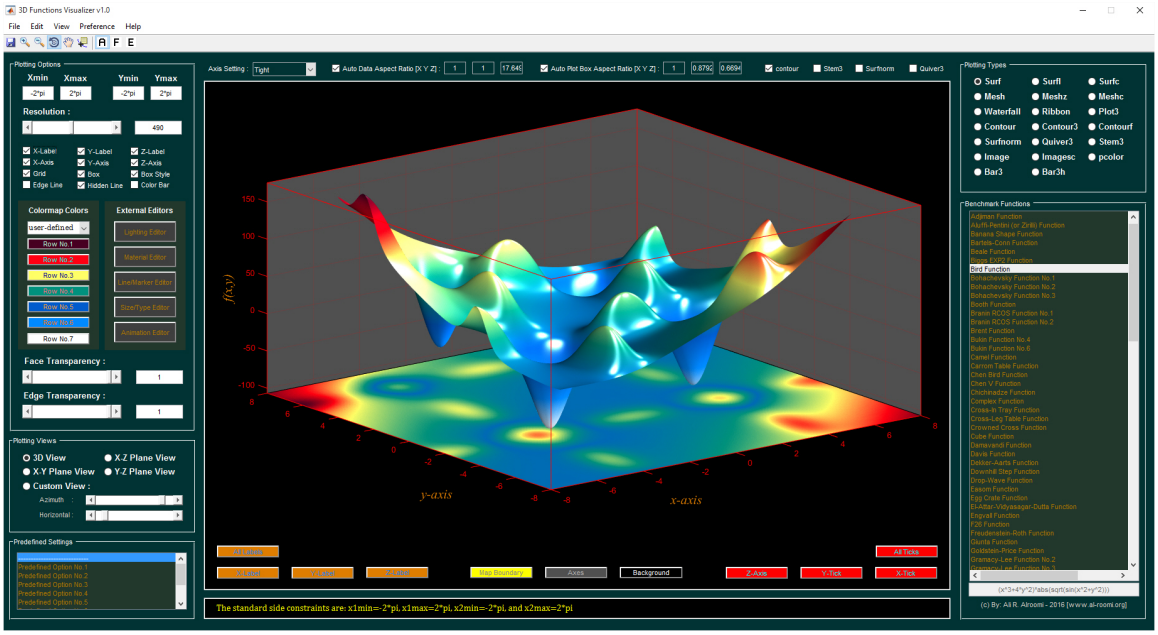
As can be clearly seen from all these equations, (2.36)-(2.40), one of the main disadvantages of SAPF is that it consumes more CPU time compared with other penalty functions.

2.1.9 Performance Criteria Used to Evaluate Algorithms

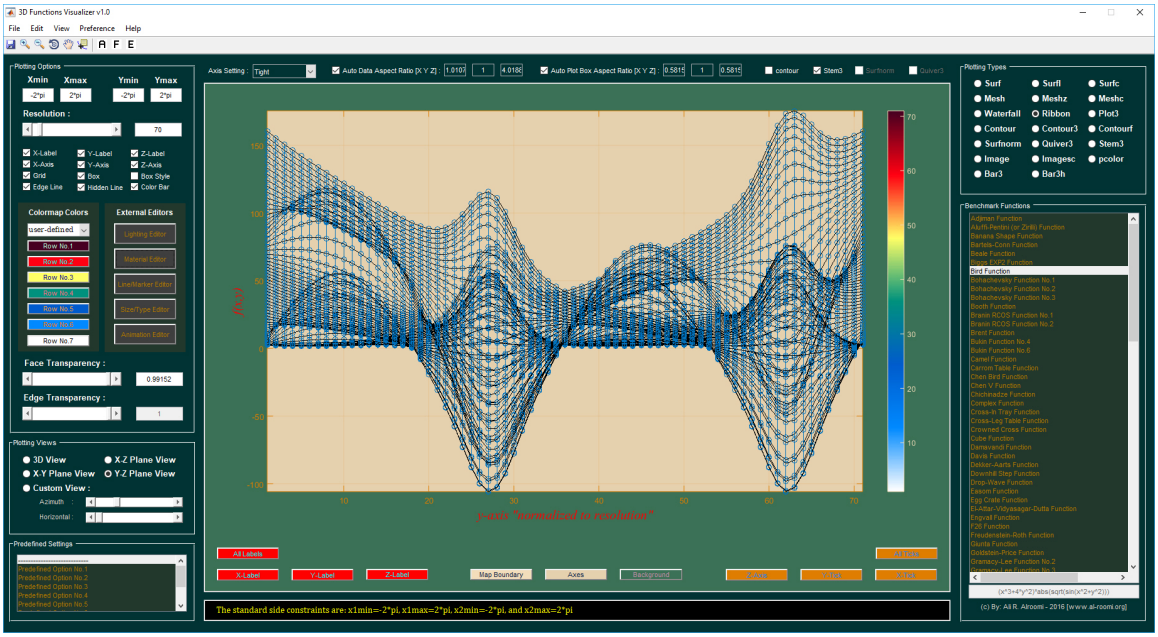
To evaluate the performance of any optimization algorithm, it is necessary to define the performance criteria based on which the algorithm performance can be assessed and compared with other algorithms. The first stage is to run or execute the algorithm to solve some standard **benchmark functions**[13]. These functions are classified as: **unimodal** (*having one optimum solution*) and **multimodal** (*having multiple optimum solutions*). Also, they can be classified as: **unconstrained/constrained**, **static/dynamic**, **convex/non-convex**, **smooth/non-smooth**, **non-noisy/noisy**, **non-shifted/shifted**, **non-rotated/rotated**, **single-objective/multi-objective**, etc. The other properties that might be considered are: **continuity**, **separability**, **differentiability**, **scalability**, etc. The software shown in Figure 2.7 is developed to study the 3D structure of any 2-dimensional function in detail.

The next step is to let the optimization algorithm to run multiple times and collecting the fitness obtained from each random run. Then, the **best**, **worst**, **mean**, and **standard deviation**[14] of the final solutions are extracted. By these fundamental records, someone can start comparing with other results reported in the literature. It is important to ensure that the initialization stage has the same parameters used

[13] Also known as **test functions**.
[14] Sometimes, the **median** metric is also computed.



(a) 3D View of a Surface Plot of the Bird Benchmark Function with Contour and Modified Lighting and Surface Material



(b) Y-Z Plane View of a Ribbon Plot of the Bird Benchmark Function with Contour and Stem Entries

Figure 2.7: Our Benchmark Function Visualizer Software

in other studies (i.e., population size N, number of generations G, etc) to get a fair comparison. For the minimization mode, the best of the best and the worst of the best-obtained solutions per certain number of simulation runs or trials can be respectively computed as follows:

$$f_{\text{best\_ever}} = \min\left(f_{\text{best},1}, f_{\text{best},2}, \cdots, f_{\text{best},i}, \cdots, f_{\text{best},T_r}\right) \tag{2.41}$$

$$f_{\text{worst\_ever}} = \max\left(f_{\text{best},1}, f_{\text{best},2}, \cdots, f_{\text{best},i}, \cdots, f_{\text{best},T_r}\right) \tag{2.42}$$

where T_r is the total number of trials and $f_{\text{best},i}$ represents the best solution obtained in the ith trial. Also, the mean and standard deviation can be respectively calculated as follows [194]:

$$f_{\text{mean}} = \frac{\sum_{i=1}^{T_r} f_{\text{best},i}}{T_r} \tag{2.43}$$

$$f_{\text{std\_dev}} = \sqrt{\frac{\sum_{i=1}^{T_r} \left(f_{\text{best},i} - f_{\text{mean}}\right)^2}{T_r - 1}} \tag{2.44}$$

For standard benchmark functions, the global optimal solution (f^*) is known and given. Based on this, the absolute error (Err_{abs}) between the estimated global optimal solution $f_{\text{best},i}$ and the exact solution f^* can be calculated as follows:

$$\text{Err}_{\text{abs},i} = |f_{\text{best},i} - f^*| \tag{2.45}$$

Thus, (2.41)-(2.44) can be respectively replaced by:

$$\text{Smallest error :} \quad \text{Err}_{\text{smallest\_ever}} = \min\left(\text{Err}_{\text{abs},1}, \text{Err}_{\text{abs},2}, \cdots, \text{Err}_{\text{abs},T_r}\right) \tag{2.46}$$

$$\text{Largest error :} \quad \text{Err}_{\text{largest\_ever}} = \max\left(\text{Err}_{\text{abs},1}, \text{Err}_{\text{abs},2}, \cdots, \text{Err}_{\text{abs},T_r}\right) \tag{2.47}$$

$$\text{Mean error :} \quad \text{Err}_{\text{mean}} = \frac{\sum_{i=1}^{T_r} \text{Err}_{\text{abs},i}}{T_r} \tag{2.48}$$

$$\text{Standard deviation :} \quad \text{Err}_{\text{std\_dev}} = \sqrt{\frac{\sum_{i=1}^{T_r} \left(\text{Err}_{\text{abs},i} - \text{Err}_{\text{mean}}\right)^2}{T_r - 1}} \tag{2.49}$$

In addition to the above criteria, the algorithm computational speed[15] can also be used as one performance criterion. Advanced performance evaluations can be done by conducting some statistical and sensitivity tests; as reported in [31]. Also, some additional performance evaluations are shown in [67] with an extensive description and new ideas.

[15] Also known in the literature as the processing speed (CPU time).

2.1.10 Types of Optimization Techniques

In general, the optimization techniques are classified into two main categories, called classical (or traditional) and modern (or non-traditional) techniques. It is hard to collect them all in one tree-diagram. Instead, the most popular types are briefly summarized in Figure 2.8 [95, 107, 133, 136, 218, 260, 279, 316, 345, 374].

2.2 Classical Optimization Algorithms

Traditional optimization algorithms are the most known methods. They need no special knowledge from other fields of science, like biology and its branches. These algorithms are straight-forward, which follow systematic mathematical steps, like finding some derivatives, constructing matrices, tracing the error deviation between two iterations, etc.

Some advantages of these optimization techniques are summarized as follows [95, 133, 136, 218, 260, 279]:

- They are very fast optimization algorithms. If the initial guess is good, then these techniques become very useful to be embedded in systems that need fast decisions, such as power system protection. The high computational speed performance comes from their simple structures and also because they are **single-point algorithms**. Thus, dealing with just one individual per each iteration will definitely accelerate the computational speed and save part of the memory[16].

- These techniques are very old and thus they are well-established and available everywhere in thousands of books, which make them easy to be reached. Add to that, there are many ready-made codes with different programming formats and languages such as **Fortran**, **MATLAB**, **JAVA**, **C/C++**, **Python**, **Julia**, etc.

- They provide one unique solution every time they are executed with the same initialization parameters.

[16]This term (i.e., the computational speed) may not be correct in some conditions. Suppose, there is a very *high*-dimensional problem ($n = 10^4 \rightarrow 10^6$) and it needs to be optimized. Solving it by classical methods may take tens of hours or even multiple days. On the opposite side, modern optimization algorithms could find acceptable near-optimal solutions within just a few iterations.

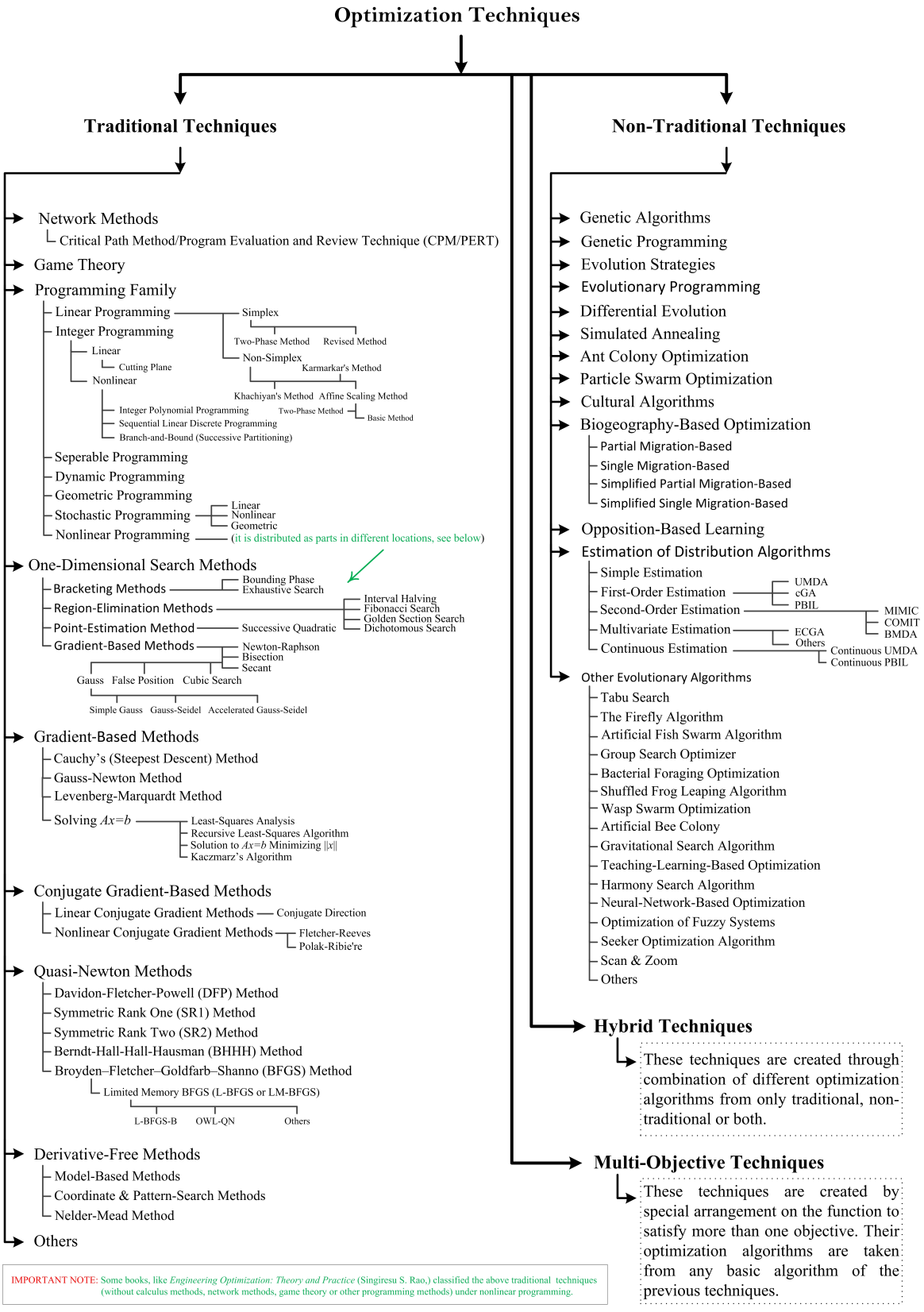


Figure 2.8: Summary of the Most Popular Optimization Techniques

- They have a solid mathematical foundation and principles.
- These **gradient-based methods** can be used as **fine-tuning** sub-algorithms in hybrid global optimization techniques, as will be covered later.

Unfortunately, there are many limitations associated with this category of optimization algorithms. Some of these weaknesses are summarized as follows:

- Some of them are very complicated algorithms, hard to be implemented, and require lots of advanced mathematical expressions.
- Some of them are restricted to *one*-dimensional problems.
- Some of them have matrices and/or require derivatives.
- Most of them are prone to easily trap into local optima, especially if the initial guess is not good.
- This field is very old and thus it is hard to get a creditable contribution

In this dissertation, some classical optimization methods are used[17]. However, they are not individually applied. Instead, they are hybridized with **meta-heuristic optimization algorithms** as fine tuners. It has to be mentioned that the Python's and MATLAB's built-in libraries are used for these classical optimization algorithms.

2.2.1 Linear Programming (LP)

Linear optimization or **linear programming** (**LP**) is one of **operations research** (**OR**) tools used to find the best or optimal solution of linear **mathematical models**. In real-world applications, the nonlinearity enters in different parts of the problem, such as in its objective function as well as equality and inequality constraints. Thus, LP is considered as a special type of **mathematical programming** that requires the following conditions [136, 246, 316, 374]:

- Linear objective function
- Linear constraints
- Non-negative decision variables

[17] Please note that the Newton-Raphson algorithm and its modified versions, which are classified as classical optimization algorithms, are used in many studies of electric power systems engineering.

2.2.1.1 Historical Time-Line

Based on the proverb "***necessity is the mother of invention***", the origin of LP can be traced back to the World War II[18] when the armies tried to find a proper way to deal with the military problems effectively and economically; especially when the army A wants to increase the losses of the enemy B with some limited and scarce resources. The history told us that the first one is the Russian mathematician Leonid Kantorovich who in 1939 proposed a method to solve linear problems. However, the work was published in 1959 [334]. Also, at the same time, the Dutch-American mathematician and economist Tjalling Charles Koopmans independently proposed a method to solve linear economic problem [341]. Three years later (i.e., in 1941), the American mathematician and physicist Frank Lauren Hitchcock successfully formulated a method that can solve linear transportation problems [334]. After around five years (i.e., between 1946 and 1947), George Bernard Dantzig independently generalized the formulation of linear programming as a suitable tool to solve planning problems in US Air Force. His first paper was titled "***Programming in a Linear Structure***" [104]. The term "linear programming" is coined by Koopmans in 1948, and one year later Dantzig published the simplex method.

Nowadays, LP is a very popular linear problem solver for many applications in mathematics, science, engineering, and business. LP is considered as an entry-level to understand more complicated programming methods. As can be clearly seen in Figure 2.8, LP is just one type of **programming family**, and it is available in two main categories called the **simplex** and **non-simplex methods**. Simplex LP problems can be solved by two ways called **tabular** and **algebraic** forms. The latter is an extension of the conventional algebraic method, which is used to overcome some weaknesses such as requiring many fixed steps and non-smart enough to jump from infeasible sets. Also, LP problems can be solved by searching within the plot boundaries of the constraints, which is known as the **graphical method**.

[18] Some authors said that the origin is dated back to the beginning of the nineteenth century when Fourier suggested a method to solve linear inequalities problem in 1827, and that method is known as ***Fourier–Motzkin elimination*** (**FME**) [341].

2.2.1.2 Mathematical Formulation of LP Problems

By referring to the preceding three conditions of LP and the standard formulations given in Subsection 2.1.7, the mathematical model can be expressed in many forms. For example, it can be as follows:

$$\begin{aligned} \text{Maximize:} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{Subjected to:} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_n \\ \text{and:} \quad & x_1 \geqslant 0 \ ; \ x_2 \geqslant 0 \ ; \ \cdots \ ; x_n \geqslant 0 \end{aligned} \tag{2.50}$$

Or, it can be formulated in vector notation as follows:

$$\begin{aligned} \text{Maximize:} \quad & [C]\,[X] \\ \text{Subjected to:} \quad & [A]\,[X] \leqslant [b] \\ \text{and:} \quad & [X] \geqslant [0] \end{aligned} \tag{2.51}$$

where the vectors of n variables and m constraints are defined as follows:

$$[C]_{1\times n} = \langle c_1, c_2, \cdots, c_n \rangle$$

$$[X]_{n\times 1} = \langle x_1, x_2, \cdots, x_n \rangle^T$$

$$[A]_{m\times n} = [a_{ij}]$$

$$[b]_{m\times 1} = \langle b_1, b_2, \cdots, b_m \rangle^T$$

2.2.1.3 Linear Programming Solvers

Because LP is used since a long time ago, so it is not surprising that many free and commercial software and programming languages have special packages and/or libraries to solve LP problems by **simplex**, **revised simplex**, **interior-points**, etc. For instance, MATLAB, Python, **Mathematica**, **Maple**, **Octave**, **LINGO**, and **MS Excel**.

2.2.2 Sequential Quadratic Programming (SQP)

Sequential quadratic programming (**SQP**) is one of the most effective optimization methods used to solve constrained nonlinear optimization problems. The SQP method is applicable if the objective function and the constraints are twice continuously differentiable. This optimization technique generates steps by solving quadratic subproblems. That is, it solves a sequence of optimization subproblems. The SQP can be used in both line search and trust-region strategies [273]. SQP is appropriate for small and large problems and it is well-suited to solve problems with significant nonlinearities [75]. A full description with detailed mathematical modeling of SQP can be found in [75, 273, 279], and a good practical introduction with MATLAB and illustrative examples are introduced in [153].

This classical optimization algorithm will be hybridized later, with one of our purely hybrid meta-heuristic optimization algorithm, as a fine tuner.

2.3 Meta-Heuristic Optimization Algorithms

Meta-heuristic optimization algorithms are also called **modern optimization algorithms**, **non-classical optimization algorithms**, and **non-traditional optimization algorithms**. They are classified into **probabilistic** and **stochastic algorithms** [107, 122, 134, 228, 316, 345, 374]. The disadvantages of classical optimization algorithms motivate many researchers to think about other innovative approaches that can solve all the headache problems of the classical optimization algorithms. To think about approaches that can converge accurately and quickly to the area where the global optima are located without using any derivatives, matrices, or even initial points. Many mathematical theories, principles, and foundations taken from different disciplines have been utilized to design the mechanism of these unfamiliar algorithms. For example[19]:

- Based on the branches of physics science: the **simulated annealing** (**SA**) algorithm from the physics of matter, and the **gravitational search algorithm**

[19] It has to be said that there are different sub-categories and sub-sub-categories of these algorithms. For instance, some of them are called **evolutionary algorithms** (**EAs**), which are part of a more broad sub-category called the **nature-inspired algorithms**. Also, each EA has many versions. For example, some versions of GA are binary GA, real GA, stud GA (sGA), micro GA (μGA), etc.

(**GSA**) from the law of gravity.

- Based on the branches of biology science: the **genetic algorithm** (**GA**) from the genetic science, and the **biogeography based-optimization** (**BBO**) from the biogeography science.

With these inspirations, many researchers have successfully applied these models to solve highly complicated problems by just plugging them into the design function and then pressing the run button.

These modern optimization techniques can also be classified according to their number of generated individuals or candidate solutions per each iteration or generation. Algorithms having only one individual per each iteration (like traditional techniques) are called **single-point** or **single-solution algorithms**. Such algorithms are the **tabu search** (**TS**) and SA algorithms. Whereas, the term **population-based** (**PB**) **algorithm** is used to identify the algorithms that generate multiple individuals per each generation. Such algorithms are the **particle-swarm optimization** (**PSO**) and the **ant colony optimization** (**ACO**) algorithms. Of course, the preceding non-traditional single-solution algorithms also can escape from trapping into local optima, and they are considered as **global optimizers**.

Also, it is essential to differentiate between **heuristic**, **meta-heuristic** and **hyper-heuristic** terms, because many references just mentioned them as "heuristic" methods without any clear definition. It has to be known that the heuristic-based techniques are problem-dependent algorithms that can learn from the given information about the design function and then adapting with that to provide good results. Such techniques could fail due to the chance to trap into local optima. Now, suppose that no useful information is given about the optimization problem; like the proper path that can guide the algorithm to reach the optimum solution or what the optima looks like. Then the techniques that can reach the space, where the global optima are located without knowing how and from where to start, are called meta-heuristic-based techniques[20]. These algorithms benefit from the data of the previously obtained solutions to determine the location of the best solution within the search space. Therefore, they might not guarantee to settle exactly on the global optimum solution. Instead, they might provide some approximate and **near-global** solutions.

[20]Some references called them **black-box optimization algorithms**.

Hyper-heuristic is the most advanced technique compared with the preceding two techniques. It depends on the heuristic or meta-heuristic technique to create its own search space. Therefore, the word "hyper-heuristic" can be translated as a "heuristic search for heuristic" [146, 228, 359].

Some advantages of modern optimization algorithms are [107, 122, 228, 316, 345, 374]:

- No need anymore to find derivatives or constructing matrices.
- Very robust and can converge to the space of the global optima.
- They are relatively easier to be understood.
- The designers need less time and limited libraries to create their programs.
- The designed programs can be used as general-purpose optimizers for any plugged-in design function.
- This branch is new, and thus the door is widely open to conduct many types of research in this field.

On the other hand, some of the main disadvantages of modern optimization algorithms are:

- The population-based techniques are time-consuming. The processing time will increase significantly as the population size increases.
- The final solution is tuned after completing many generations because the algorithms are **probabilistic-/stochastic-based methods**.
- Still, the number of available references and codes is smaller than that of the traditional techniques, especially for the most recent invented techniques.
- To understand their principles and how they work, the researcher needs to study some special topics in physics, biology, or/and other branches of science. This may become hard for those people who have just pure background in the field of mathematics, engineering, computer science, or economics.

In the next subsections, some detailed information about the meta-heuristic optimization algorithms, used in this dissertation, is given.

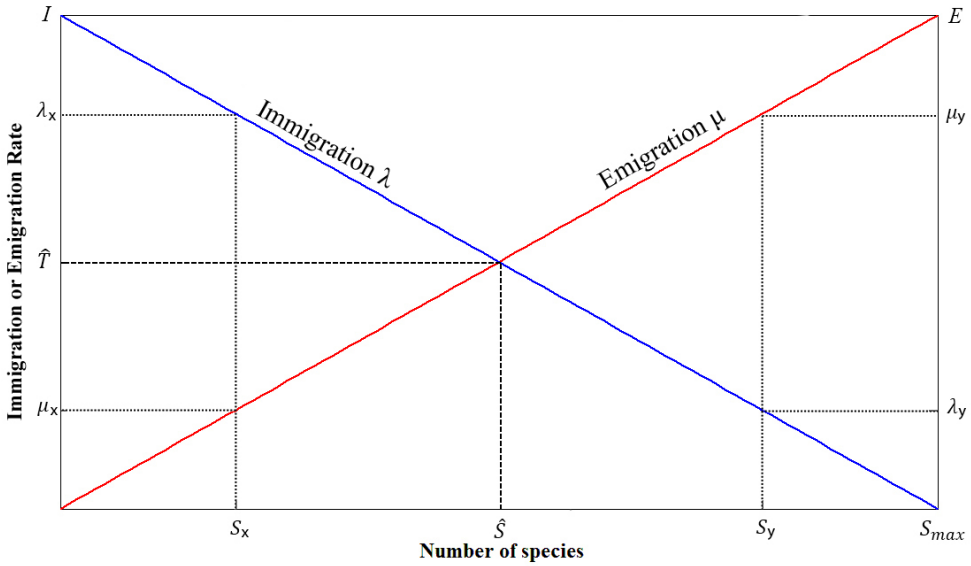


Figure 2.9: Simplified Equilibrium Model of Biota in a Single Island

2.3.1 Biogeography-Based Optimization (BBO)

The mechanism of this new population-based evolutionary algorithm is inspired by an old scientific study conducted in biogeography[21] by the ecologists Robert H. MacArthur and Edward O. Wilson in the period between 1960-1967 [239, 240]. This study is known as "***The Theory of Island Biogeography***". The theory proposes that the dynamic equilibrium between immigrated and extinct species controls the endemic species on isolated islands[22].

The **immigration rate** λ and the **emigration rate** μ can be set in many ways[23]. To simplify the mathematical process, MacArthur and Wilson used a simplified linear migration model with equal maximum immigration and emigration rates (i.e., $I =$

[21] Biogeography is a branch of biological science. It heavily relies on theories and data taken from ecology, population biology, systematics, evolutionary biology, and earth science [226]. Biogeography seeks to describe, analyze and explain the geographic patterns and changes in the distribution of ecosystems and fossil species of plants (**flora**) and animals (**fauna**) through geological space and time [101, 195].

[22] In island biogeography, the word "***island***" could be an aquatic island, desert oasis, lakes or ponds, mountain-tops (sky-islands), caves, individual plants, microcosms or even patches of terrestrial ecosystems [43, 266, 276].

[23] The emigration and immigration rates can be modeled as exponential, logistic, linear, etc [98, 226, 238]. Also, the maximum emigration and immigration rates can be unequal (i.e., $I \neq E$) [238, 240]. Moreover, the equilibrium location $\hat{S}$ can be shifted to the left-side or to the right-side based on the type of rate functions, the area of island and/or the distance or isolation between the recipient island and the source island or mainland [30, 226, 227, 240].

$\lambda_{\max} = E = \mu_{\max}$) as shown in Figure 2.9. The symbol $\hat{T}$ denotes the species turnover rate, which happens when the species density settles on the equilibrium state $\hat{S}$. The symbol $S_{\max}$ denotes the maximum number of endemic species on that island [238–240]. Thus, $\lambda_{\max}$ or I happens when there is no available species on the ith island, and $\mu_{\max}$ or E happens when all the nests are occupied by the emigrated species from the mainland and/or other islands [237].

In BBO, the population size N is simplified to be equal to the maximum number of species $S_{\max}$. Therefore, λ_i and μ_i of the basic migration model, depicted in Figure 2.9, can be determined as follows:

$$\mu_i = \left(\frac{E}{N}\right) i \tag{2.52}$$

$$\lambda_i = 1 - \mu_i = I\left(1 - \frac{i}{N}\right) \tag{2.53}$$

Suppose that at time t the island contains i species with probability $Pr_i(t)$, then the variation of the probability from t to $(t+\Delta t)$ can be described as follows [239,240]:

$$Pr_i(t+\Delta t) = Pr_i(t)(1 - \lambda_i \Delta t - \mu_i \Delta t) + Pr_{i-1}(t)\lambda_{i-1}\Delta t + Pr_{i+1}(t)\mu_{i+1}\Delta t \tag{2.54}$$

Considering (2.54), to have i species at time $(t + \Delta t)$, one of the following three conditions should be satisfied [239, 240]:

1. i species at time t, and no migrated species during the interval Δt;

2. $(i - 1)$ species at time t, and one species immigrated;

3. $(i + 1)$ species at time t, and one species emigrated.

From calculus, it is known that the ratio $\left(\frac{\Delta Pr_i}{\Delta t}\right)$ approaches $\dot{Pr}_i(t)$ as $\Delta t \to 0$:

$$\begin{aligned} \dot{Pr}_i(t) &\cong \lim_{\Delta t \to 0} \frac{Pr_i(t+\Delta t) - Pr_i(t)}{\Delta t} \\ &\cong -(\lambda_i + \mu_i)Pr_i(t) + \lambda_{i-1}Pr_{i-1}(t) + \mu_{i+1}Pr_{i+1}(t) \end{aligned} \tag{2.55}$$

By considering the preceding three conditions, (2.55) can be re-expressed with the following three cases:

$$\dot{Pr}_i(t) = \begin{cases} -(\lambda_i + \mu_i)Pr_i(t) + \mu_{i+1}Pr_{i+1}(t), & \text{if } i = 0 \\ -(\lambda_i + \mu_i)Pr_i(t) + \lambda_{i-1}Pr_{i-1}(t) + \mu_{i+1}Pr_{i+1}(t), & \text{if } 1 \leqslant i \leqslant N-1 \\ -(\lambda_i + \mu_i)Pr_i(t) + \lambda_{i-1}Pr_{i-1}(t)), & \text{if } i = N \end{cases} \tag{2.56}$$

The value of $\dot{Pr}_i(t)$ can also be determined by using the matrix technique presented in [342], which is successfully proved in [181]. Thus, using the known values of $Pr_i(t)$ and $\dot{Pr}_i(t)$, the value of $Pr_i(t+\Delta t)$ given in (2.54) can be approximated as follows:

$$Pr_i(t+\Delta t) \cong Pr_i(t) + \dot{Pr}_i(t)\Delta t \tag{2.57}$$

Equation (2.57) is the final form that should be used in the BBO program. To find $Pr_i(t)$, two methods have been used by Simon in [342]. The first one is by solving (2.56) iteratively, while the other one can be directly applied through the following theorem:

Theorem 1

The steady-state value for the probability of the number of each species is given by:

$$Pr(\infty) = \frac{v}{\sum_{w=1}^{N+1} v_w} \tag{2.58}$$

The eigenvector v can be computed as follows:

$$\begin{aligned} v &= [v_1, v_2, \cdots, v_{N+1}]^{\intercal} \quad , \quad {}^{\intercal} \text{ means transpose} \\ v_w &= \begin{cases} \frac{N!}{(N+1\text{-}w)!(w\text{-}1)!}, & \text{for } w = 1, 2, \cdots, w' \\ v_{N+2\text{-}w}, & \text{for } w = w'+1, \cdots, N+1 \end{cases} \\ & \text{where: } w' = \left\lceil \frac{N+1}{2} \right\rceil \end{aligned} \tag{2.59}$$

Although the second method is easier and $Pr_i(t)$ can be directly computed without any iterations, this approach is not preferred in many numerical programming languages, because they set $N = \infty$ when $N > 170$. This infinity issue can be resolved if an additional sub-algorithm is used. However, dealing with long product operations requires extra CPU time [44]. Based on this, the iterative method is more flexible and more convenient. Thus, it is adopted in this dissertation.

In BBO, the objective function can be optimized if each island is considered as one individual and the independent variables of each individual are dealt as features.

The solutions can be enhanced if these features are distributed between the source and recipient islands. The source island could become a recipient island for other better islands [342]. That is, the richness of species on an island is decided through a probabilistic process. If many good **biotic** and **abiotic** features[24] are available on an island, then it will be good land for immigrants. Each feature is called a **suitability index variable** (SIV), which represents one independent variable of such a problem in BBO. The **island suitability index** (ISI)[25] is the dependent variable, which varies with any change in any element of the vector SIV. Because BBO is a population-based algorithm, so optimizing n-dimensional problem with N-individuals can be mathematically represented as follows:

$$ISI_i = f_i(SIV_{i,1}, SIV_{i,2}, \cdots, SIV_{i,n}) \quad , \quad i = 1, 2, \cdots, N \tag{2.60}$$

Once the initialization stage is completed, the BBO algorithm should pass some sub-algorithms:

2.3.1.1 Migration Stage

The main idea of this stage is to share the good features of rich islands to modify poor islands. Because the selection is done through a probabilistic process, so the ith island is likely to be selected as a source of modification if ISI_i is high, and vice versa for the jth recipient island. From Figure 2.9, low λ_i and high μ_i indications mean a large number of endemic species are available on the ith island. Thus, the solution ISI_i is high. As an example, point S_x is located before $\hat{S}$, so λ_x is high and μ_x is low, and thus ISI_x is considered as a poor solution. On the opposite side, point S_y is located after $\hat{S}$, so λ_y is low and μ_y is high, and thus ISI_y is considered as a good solution. Based on this, μ_i and λ_i are used as metrics to know the solution quality of each island.

Through the migration process, the islands with low ISI could be improved per each new generation, and at the same time, the solution quality of the best islands are kept away from any corruption.

[24] Biotic factors: predation, competition, interactions, etc. Abiotic factors: wind, water, sunlight, temperature, pressure, soil, etc [241].

[25] It is also called the **habitat suitability index** (HSI).

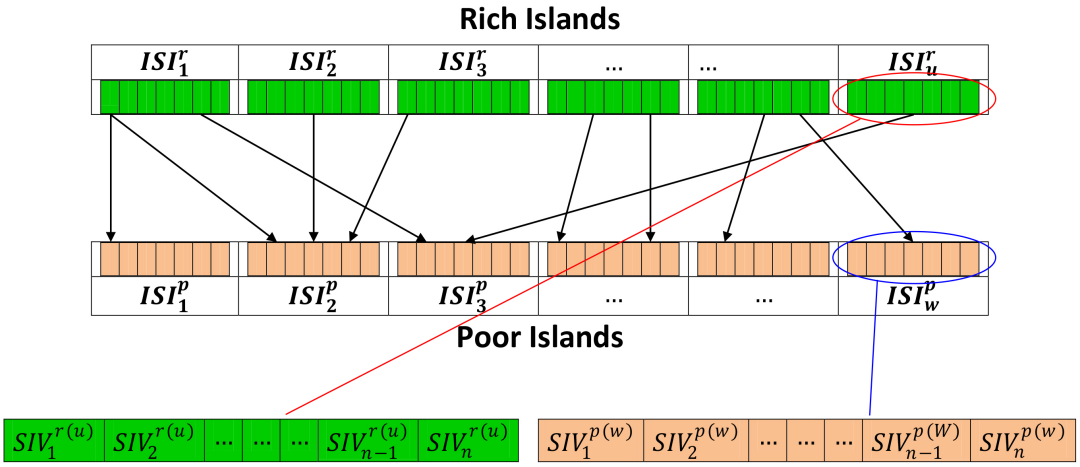


Figure 2.10: Migration Process Between Different Islands in BBO

The original BBO algorithm comes with four migration forms, as described in [43, 344], and called partial, simplified partial, single, and simplified single migration based (PMB, SPMB, SMB, and SSMB) models. The first published BBO paper used the PMB model [342], which is graphically described in Figure 2.10. As can be clearly seen from this figure, these rich and poor islands act as sources and recipients of those migrated n SIV. Each SIV s of a poor island is updated by SIV σ that is probabilistically selected from one rich island. For the SPMB model, the n SIV of poor islands are updated from the first best island(s), which in turn increases the probability to trap into local optimum solutions. The migration process of the SMB and SSMB models are respectively similar to those of the PMB and SPMB models with one main difference: only one randomly selected SIV s of each poor island is modified. The last two models are faster, but with low convergence rates. This dissertation will consider all the essential modifications presented in [44] as a basis for the proposed MpBBO-SQP algorithm. Thus, the original BBO algorithm (before being hybridized with SA and SQP) can save around 32.32% of its total CPU time and with better performance than that of the PMB-BBO model. The migration process used in this dissertation is described by Algorithm 2.

Algorithm 2 Partial Migration Pseudocode

Require: Let ISI_i denote the ith population member and contains n features

Require: Define emigration rate μ_i and immigration rate λ_i for each member

for $i \leftarrow 1$ *to* N **do** {where N = number of islands or individuals, see (2.60)}
  for $s \leftarrow 1$ *to* n **do** {where n = number of features "SIV" or design variables}
    Use λ_i to probabilistically select the immigrating island ISI_i
    for $j \leftarrow 1$ *to* N **do** {Break once ISI_j is selected}
      Use μ_j to probabilistically decide whether to emigrate to ISI_i
      if ISI_j is selected **then** {where $ISI_i \neq ISI_j$}
        Randomly select an SIV σ from ISI_j
        Replace a random SIV s in ISI_i with SIV σ
      end if
    end for
    next SIV
  end for
  next island
end for

2.3.1.2 Mutation Stage

As with many nature-inspired algorithms, this stage is very essential to increase the exploration level. The mutation process of the BBO algorithm can be defined as random natural events that affect the availability of the biotic and abiotic features on an isolated island, which in turn reflected on the total endemic species on that island. These events could be positive (like shipwrecks and wind pollination) that increase the species density, or they could be negative (like volcanoes, diseases, and earthquakes).

In BBO, the species count probability Pr is used exclusively to find the mutation rate [342]. Thus, many choices are available to researchers to select their preferable mutation rate, such as Gaussian, Cauchy and Lèvy mutation operators reported

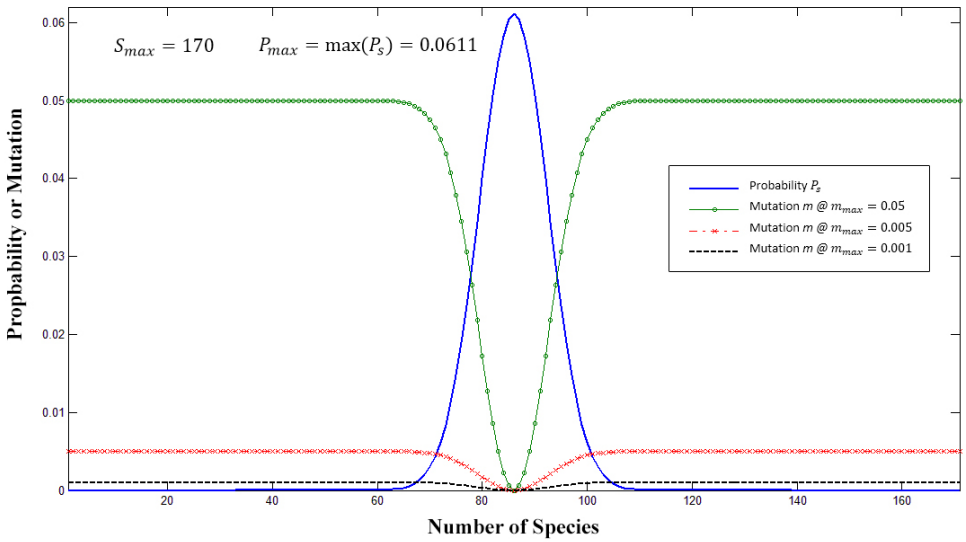


Figure 2.11: Comparison Between Pr and m at Different $m_{\max}$

in [152]. The original mutation rate, which is also used in this dissertation, is described as follows [342]:

$$m_i = m_{\max}\left(1 - \frac{Pr_i}{Pr_{\max}}\right) \tag{2.61}$$

where $Pr_{\max}$ is the largest element of the vector Pr, and $m_{\max}$ is a user-defined maximum allowable value that m_i can reach.

As can be seen from (2.61), the mutation rate is inversely proportional to the probability rate (i.e., $m_i \propto^{-1} Pr_i$). This equation forces m_i to be equal to $m_{\max}$ at ($Pr_i = 0$), and equal to 0 at the largest element of Pr. It can be graphically represented as shown in Figure 2.11.

The mutation rate will flip the bell-shape graph of the probability rate. The main objective of using m_i rather than Pr_i is to have better control over the islands targeted for the mutation stage. That is, the islands located at or near the equilibrium point $\hat{S}$ will be preserved, while the other islands sorted on both sides will have a higher chance to be mutated and hence could be improved. The mutation process is described by Algorithm 3.

Algorithm 3 Original Mutation Pseudocode

1: **for** $i \leftarrow 1$ *to* N **do** {where N = number of islands or individuals, see (2.60)}
2: Calculate probability Pr_i based on λ_i and μ_i {by iterative or eigenvector method}
3: Calculate mutation rate m_i {using (2.61)}
4: **if** rand $< m_i$ and $i \geqslant R_m$ **then** {R_m is a user-defined mutation range}
5: Replace n SIV vector of ISI_i with a randomly generated n SIV vector
6: **end if**
7: **end for**

2.3.1.3 Clear Duplication Stage

If this optional stage is used in BBO, then the diversity of the problem features could increase. The reason behind this is that the emigrated SIV σ will take the same value and place in other island(s), so these duplicated features may have an insignificant impact on their ISI. For *one*-dimensional problems, duplicated SIV will give duplicated islands. In this situation, the exploration level will decrease and the algorithm may quickly settle on a non-global optimum solution[26]. The main purpose of this stage is to check all n SIV of all N ISI whether they are duplicated or not. If any duplicated feature is detected, then it is replaced by a new randomly generated feature. This process is described by Algorithm 4 [343].

Algorithm 4 Clear Duplication Pseudocode

Require: Check all n SIV on all N ISI
1: **while** there is a duplicated SIV **do**
2: **for** $i \leftarrow 1$ *to* N **do** {where N = number of islands or individuals, see (2.60)}
3: **if** any duplicated SIV s is detected **then**
4: Replace the duplicated SIV s in ISI_i with a randomly generated SIV σ
5: **end if**
6: **end for**
7: **end while**

[26]The blended BBO, given in [236], is immune to this duplication phenomenon.

It is important to know that this sub-algorithm must be partially deactivated for discrete features of **mixed-integer optimization problems**, and completely deactivated for **combinatorial optimization problems**. Based on the step-size resolution and the side constraints of discrete variables, ignoring this vital step could lead to trapping into infinite loops.

2.3.1.4 Elitism Stage

Suppose that the good individuals obtained in the last generation are ruined by the previous BBO stages (i.e., migration, mutation, and/or clear duplication). Then, those good solutions will be lost forever if this optional stage is not activated in the BBO program. This stage can provide a rollback option to rescue the last state of the corrupted best islands, or **elites**, and then recycle them back into the population of the next generation [346].

The overall mechanism of the BBO algorithm is depicted by the flowchart shown in Figure 2.12. The software shown in Figure 2.13 is developed to create many possible BBO structures flexibly without knowing any programming skills. It gives the user the ability to hybridize BBO with many other sub-algorithms and many options.

2.3.2 Differential Evolution (DE)

The DE algorithm is known as one of the most popular and simplest population-based evolutionary algorithms. It was presented by R. Storn and K. V. Price in 1995 [355,356]. This algorithm can be programmed easily and quickly without facing that much of challenges, and thus it gets high attention from many researchers.

Because DE is a population-based algorithm, N candidate solutions are generated at the initialization stage. Each individual contains n independent variables. Suppose that the parameters N and n are respectively used to represent the population size and the dimension of each individual. In DE, each new individual is generated by selecting three different individuals from the same population.

Referring to the literature, someone may note that there are many versions of DE. The simplest DE algorithm is called classic DE or DE/rand/1/bin[27]. The mechanism

[27]The other variations are: DE/rand/1/either-or, DE/rand/1/L, DE/rand/2/bin, DE/rand/2/L,

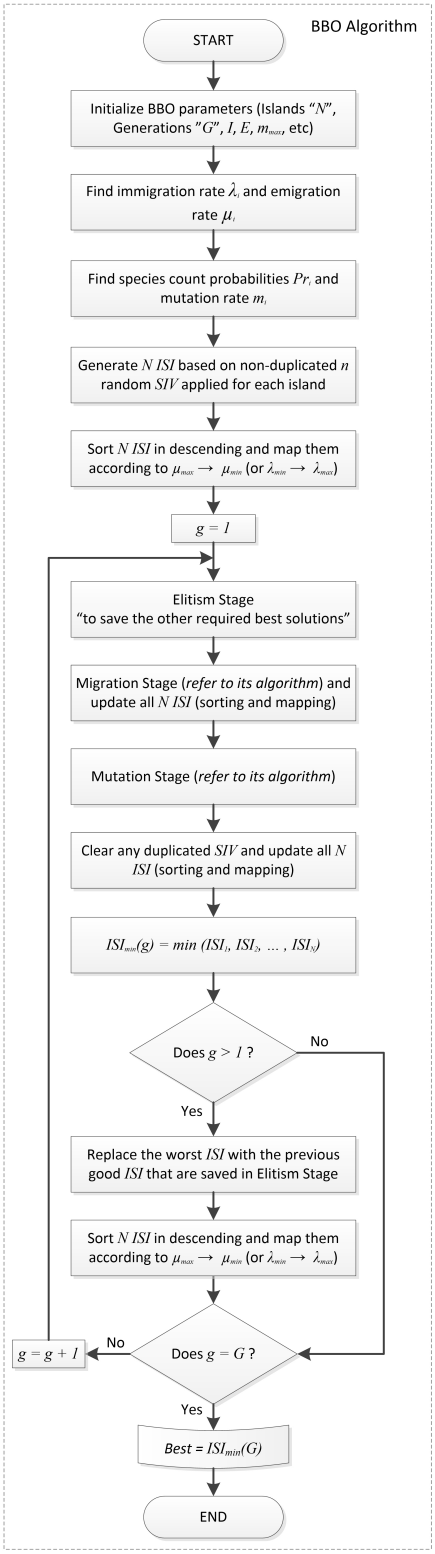


Figure 2.12: Flowchart of the BBO Algorithm

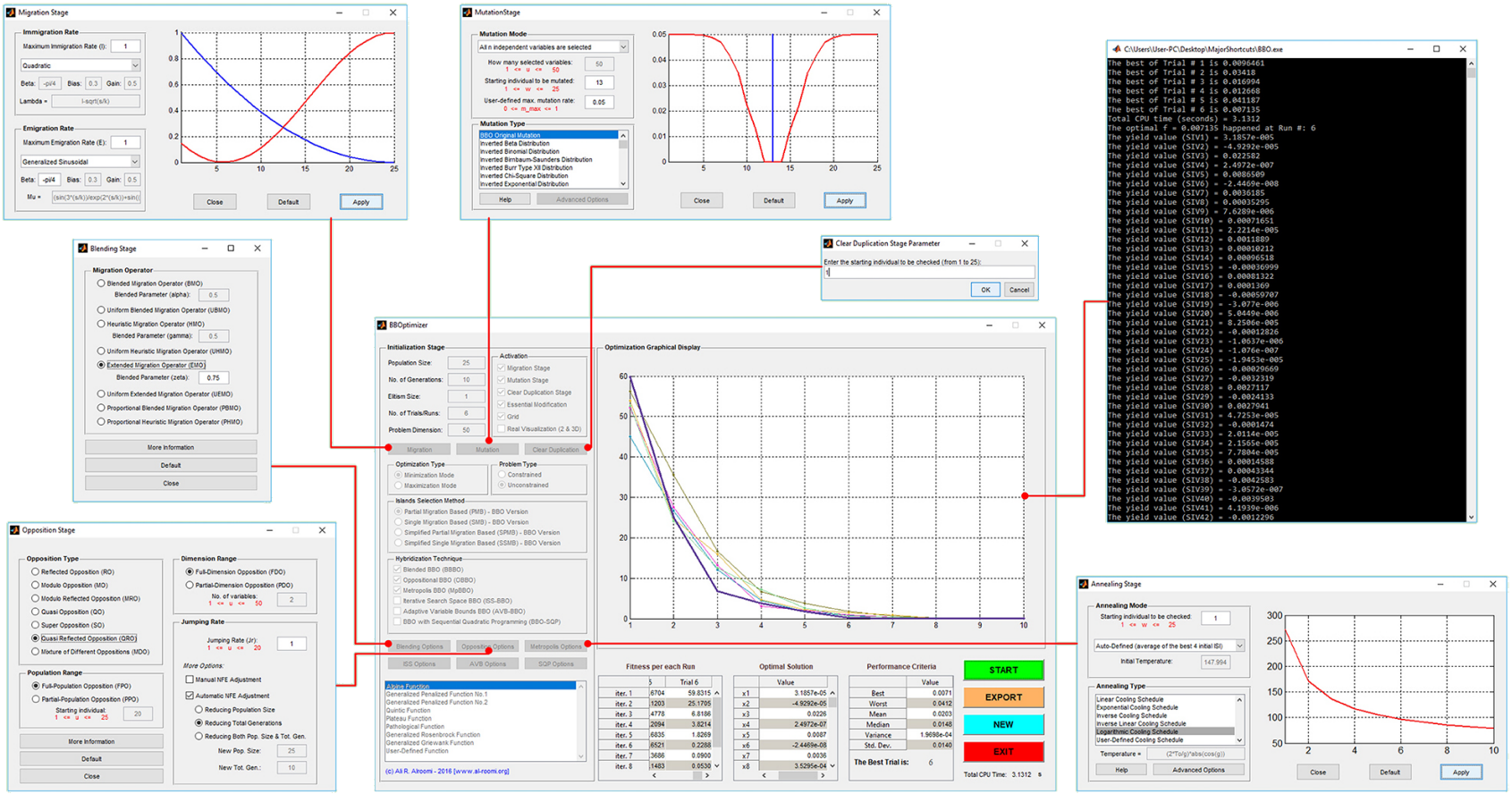


Figure 2.13: Our Intergraded Biogeography-Based Optimization Algorithm Software

of the classic DE algorithm is described by Algorithm 5. More details about the other DE versions can be found in [345].

2.3.3 Simulated Annealing (SA)

In **metallurgy** and **materials science**, the word "***annealing***" means a heat process that controls the properties of the metal (such as ductility, strength, and hardness) by heating it up to a specific temperature (above the **recrystallization** temperature), maintaining that temperature for a certain period, and then allowing it to cool slowly. By this approach, the crystals will be formed in a good shape with the lowest internal energy, and hence the metal will settle on a **crystalline state**. In case the cooling rate is very fast, the metal will be on a **polycrystalline state** so that the high internal energy will deform the structure of the crystals [316].

This slow cooling process inspired Kirkpatrick et al. [206] to design their novel single-point global optimization algorithm, which was presented in 1983 as the "simulated annealing (SA)" algorithm. After two years, similar independent work was presented by Vlado Černý [373]. The core of this derivative-free nature-inspired optimization algorithm is built based on the "**statistical mechanics**" that was demonstrated by Metropolis et al. in 1953 [256] using the concept of **Boltzmann's probability distribution**. It states that if a system is maintained in thermal equilibrium at temperature $\breve{T}$, then the probabilistic distribution $\breve{P}$ of its energy $\breve{E}$ can be achieved by [316]:

$$\breve{P}(\breve{E}) = e^{\frac{-\Delta\breve{E}}{k_B\breve{T}}} \tag{2.62}$$

where k_B is **Boltzmann's constant**. For simplicity, it is set to one. $\Delta\breve{E}$ is the difference in energy, which is translated in SA as the difference in the cost function. It can be calculated as follows:

$$\Delta\breve{E} = f(X) - f(X_0) \tag{2.63}$$

For minimization problems, the new design point X is directly accepted if the following condition is satisfied:

$$f(X) \leqslant f(X_0) \tag{2.64}$$

DE/best/1/bin, DE/best/1/L, DE/best/2/bin, DE/best/2/L, DE/target-to-best/1/bin, DE/target/1/bin, DE/target/1/L, DE/target/2/bin, and DE/target/2/L.

Algorithm 5 Classic Differential Evolution Pseudocode

Require: F = step-size parameter $\in\ [0.4, 0.9]$

Require: C_r = crossover rate $\in\ [0.1, 1]$

Require: Initialize a population of candidate solutions $\{X_i\}$ for $i\ \in\ [1, N]$

while not(termination criterion) **do**
  for each individual $X_i,\ i\ \in\ [1, N]$ **do**
    $r_1 \leftarrow$ random integer $\in\ [1, N]\ :\ r_1 \neq i$
    $r_2 \leftarrow$ random integer $\in\ [1, N]\ :\ r_2 \notin\ \{i, r_1\}$
    $r_3 \leftarrow$ random integer $\in\ [1, N]\ :\ r_3 \notin\ \{i, r_1, r_2\}$
    $V_i \leftarrow X_{r_1} + F\,(X_{r_2} - X_{r_3})$ (mutant vector)
    $\zeta_r \leftarrow$ random integer $\in\ [1, n]$
    for each dimension $j\ \in\ [1, n]$ **do**
      $r_{c_j} \leftarrow$ random integer $\in\ [0, 1]$
      if $(r_{c_j} < C_r)$ or $(j = \zeta_r)$ **then**
        $U_{i,j} \leftarrow V_{i,j}$
      else
        $U_{i,j} \leftarrow X_{i,j}$
      end if
    end for
  end for
  for each population index $i\ \in\ [1, N]$ **do**
    if $f(U_i) < f(X_i)$ **then**
      $X_i \leftarrow U_i$
    end if
  end for
end while

In case (2.64) is not satisfied, then the new design point X will not be directly rejected. Rather, it has another chance to be accepted if it passes the Metropolis criterion.

From (2.62), the probability to accept X will increase as the molten metal is heated to a very high temperature $\breve{T}$, and that chance gradually decreases during the annealing or slow cooling process. This process will avoid trapping into local optimum points when $\breve{T}$ is high. This means that the exploration level is high at the beginning of the annealing process, and the exploitation level is high at the end of that process.

Based on this fact, it is very important to initialize SA at a high temperature $\breve{T}$. But the main question here is: *how much?*. Actually, it has been found that the good initial temperature $\breve{T}_o$ for some objective functions could be unsuitable for other objective functions. Thus, the determination of $\breve{T}_o$ itself becomes a challenge. To solve this issue, many researchers estimate $\breve{T}_o$ by taking the average cost of some randomly generated individuals[28].

The classical SA algorithm is constructed with two searching loops. The external one is coded as a number of cycles or stages z, while the internal one is set as a number of iterations l. That is, SA is initialized with high $\breve{T}_o$ for the first cycle and then decreases by a specific cooling rate[29]. During each cycle, the fitness is enhanced by a certain number of iterations. Therefore, the convergence speed and solution quality highly depend on the setting of $\breve{T}_o$ and its cooling rate. Moreover, the maximum limits of the external and internal loops (i.e., the number of cycles z and iterations l) are also important settings in SA. Small z and l lead to fast computation but with low performance. In contrast, large z and l improve the performance but at the cost of a huge amount of CPU time. It is like compromising between the solution quality and the processing speed. The CPU time can be reduced if these z and l loops are terminated once the solution tolerance is satisfied (i.e., $|\Delta\breve{E}| \leqslant \varepsilon$). However, this option should be turned off in order to have a fair comparison with other algorithms when the processing speed is also considered as one performance criterion. Algorithm 6 depicts how the basic SA algorithm works [316, 345].

[28] This will be discussed later in the next section, which covers our hybrid optimization algorithms.
[29] In the literature, different cooling strategies are proposed for this purpose.

Algorithm 6 Basic Simulated Annealing Pseudocode

Require: $\breve{T}_0$ = initial temperature > 0

Require: $\alpha(\breve{T})$ = cooling function: $\alpha(\breve{T}) \in [0, \breve{T}] \; \forall \; \breve{T}$

Require: Initialize a candidate solution $\{X_0\}$ to the minimization problem f

1: **while** not(termination criterion) **do**
2: **if** $f(X) < f(X_0)$ **then**
3: $X_0 \leftarrow X$
4: **else**
5: $r \leftarrow U[0, 1]$
6: **if** $r < \exp\left[\dfrac{f(X_0) - f(X)}{\breve{T}}\right]$ **then**
7: $X_0 \leftarrow X$
8: **end if**
9: **end if**
10: $\breve{T} \leftarrow \alpha(\breve{T})$
11: **end while**

2.4 Hybrid Optimization Algorithms

In general, the main objective of these highly advanced non-traditional techniques is to accumulate the strengths and eliminate or minimize the weaknesses of individual optimization algorithms. The hybridization could be done by using at least two optimization algorithms. These algorithms can be taken from classical and/or meta-heuristic categories. In addition, the designers have a large space to maneuver and they can apply their hybridizations and/or modifications in many locations of the new algorithm. Therefore, the overall algorithm will integrate all of these improvements.

Minimizing the disadvantages associated with each individual optimization algorithm does not mean that the overall hybrid optimization algorithm will have zero disadvantages. Each additional sub-algorithm added to the new optimization algorithm will consume an extra CPU time and, at the same time, will complicate the programming code. Thus, the new optimization algorithm resulted from this hybridization will be slower[30] and hard to be modified by other programmers. Moreover, because

[30]Unless modifying/bypassing some stages of the main algorithm or/and reducing the simulation

the hybridization phase can be done anywhere on the main algorithm, so there is no standard format to follow. Further, if there is insufficient information about the proposed hybrid optimization algorithm, then this algorithm will stay lonely and no body, except the programmer(s), knows its mechanism. The goal of the following subsections is to reveal the mechanism of four hybrid optimization algorithms used in this dissertation.

2.4.1 BBO-LP

Unlike gradient-based algorithms, EAs are probabilistic-based single/multi-point search techniques, the fact that makes them very slow convergence and time consuming algorithms. Thus, many researchers prefer to use only LP and NLP techniques or, at least, hybridizing them with EAs [70, 280].

To accelerate the convergence speed and accuracy, a hybrid BBO-LP algorithm is designed. LP is selected instead of NLP for the sake of speed and simplicity. However, the LP algorithm can be incorporated with EAs only if the objective function and its design constraints are expressed in a linear form or if they are linearized as will be seen later in Chapter 4. The framework of this hybrid optimization algorithm is illustrated by the flowchart shown in Figure 2.14.

The mechanism of this hybrid optimization algorithm can be briefly described in the following steps:

- First, BBO is executed normally, and the fitness per generation is selected. It could be one or more solutions based on the elitism parameter and user preference.

- Then, the independent variables of the preceding fitness (i.e., n SIV) are set as initial point to the linearized problem in the LP sub-algorithm.

- After that, LP is executed for each elite solution to fine-tune the independent variables and exploit the solutions as much as possible.

- Finally, the worst solutions in the population are replaced with the fine-tuned

parameters.

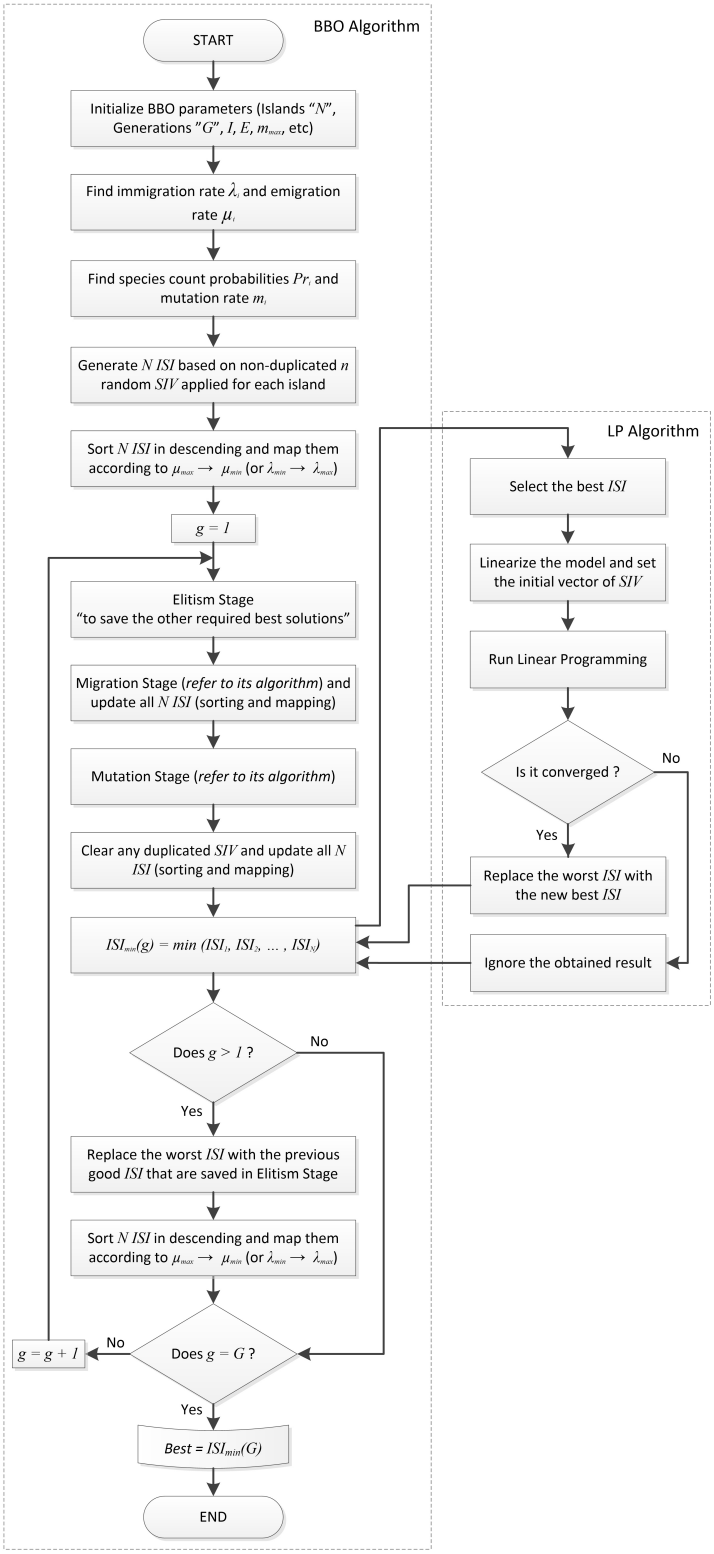


Figure 2.14: Flowchart of the BBO-LP Algorithm

solutions obtained by the LP stage[31].

- Repeat all the preceding steps again in the next generation.

This hybrid algorithm acts like a **multi-start** LP algorithm, but with the power of EAs to explore the entire search space quickly and effectively. The speed can be saved here as a result of being using just a few iterations and a small population size compared with conventional EAs. Of course, if BBO-LP is initiated with the same parameters of BBO, then the former algorithm will be slower.

2.4.2 BBO/DE

As a global optimizer, BBO has a good exploitation level, but it lacks exploration level [151,152,225,300]. In contrast, DE has a good exploration level, and it can reach the space where the global optimal solution is located in [151]. Therefore, merging the strength of DE (i.e., the good exploration level) with the strength of BBO (i.e., the good exploitation level) together in one superior optimization algorithm has been suggested by many researchers [8]. In this dissertation, a new fully discretized hybrid optimizer built based on the BBO and DE algorithms is proposed. Here, the BBO mutation stage given in Algorithm 3 is completely replaced with the mutant process of the classic DE algorithm given in Algorithm 5. To match DE with BBO, DE should have N candidate solutions and n independent variables, so each new individual injected into the existing population of BBO is generated by DE from three randomly selected and unduplicated individuals. The mechanism of this new hybrid BBO/DE optimizer is depicted in Algorithm 7. The DE algorithm can be described through lines 18 to 37 of Algorithm 7. This fully discretized hybrid optimization algorithm will be used later in Chapter 8 to solve some real-world relay coordination problems.

2.4.3 MpBBO

The idea behind this proposed hybrid optimization algorithm is to let BBO work normally, but when the migration stage is completed the modified features will not be selected unless they show better performance. If the new design point SIV^{new}

[31]This is a very important step to guarantee that the best solutions, obtained by BBO before initiating LP, are not ruined.

Algorithm 7 Pseudocode of the Proposed Hybrid BBO/DE Algorithm

Require: Initialization stage: step-size parameter F, crossover rate C_r, problem dimension n, population size N, number of generations G, etc

for $g \leftarrow 1$ *to* G **do** {where G = number of generations}
  Elitism stage (*optional*)
  for $p \leftarrow 1$ *to* N **do** {where N = number of islands or individuals}
    for $s \leftarrow 1$ *to* n **do** {where n = number of features "SIV" or design variables}
      Use λ_p to probabilistically select the immigrating island ISI_p
      for $q \leftarrow 1$ *to* N **do** {Break once ISI_q is selected}
        Use μ_q to probabilistically decide whether to emigrate to ISI_p
        if ISI_q is selected **then** {where $ISI_p \neq ISI_q$}
          Randomly select an SIV σ from ISI_q
          Randomly replace an SIV s in ISI_p with SIV σ
        end if
      end for
      next SIV
    end for
    next island
  end for
  Update all N ISI $\rightarrow$ sorting and mapping
  for $i \leftarrow 1$ *to* N **do** {where N = population size}
    Generate three random integers $\{r_1, r_2, r_3\} \in [1, N]$, where $r_1 \neq r_2 \neq r_3 \neq i$
    Create mutant vector $V_i = X_{r_1} + F \times (X_{r_2} - X_{r_3})$
    Generate a uniform random integer $\zeta \in [1, n]$
    for $j \leftarrow 1$ *to* n **do** {where n = number of independent variables, or dimension of the problem}
      Generate random number $r_{c,j} \in [0, 1]$
      if $(r_{c,j} < C_r)$ or $(j = \zeta)$ **then**
        Generate a trial element from the mutant vector $U_{i,j} = V_{i,j}$
      else
        Keep the jth element of the ith individual $U_{i,j} = X_{i,j}$
      end if

29: **end for**
30: **end for**
31: **for** $i \leftarrow 1$ *to* N **do** {where N = population size}
32: **if** $f(U_i) \geqslant f(X_i)$ **then**
33: Accept U_i as an updated individual
34: **else**
35: Reject U_i and keep the previous individual X_i
36: **end if**
37: **end for**
38: Clear duplication stage (*optional*)
39: Update all N $ISI \rightarrow$ sorting and mapping
40: Replace the worst ISI with the past best ISI stored in the elitism stage (*optional*)
41: Update all N $ISI \rightarrow$ sorting and mapping (*optional*)
42: **end for**

is worse than the previous design point SIV^{old} (i.e., $ISI(SIV^{new}) \leqslant ISI(SIV^{old})$), then the Metropolis criterion will decide whether or not to accept it. This is one of the main differences between MpBBO and the modified BBO with the immigration refusal stage, reported in [116], where the poor islands still have the chance to be selected as a source of modification, which in turn increases the exploration level.

2.4.3.1 Cooling Strategies

Many types of cooling strategies can be used in SA. The initial temperature $\breve{T}_o$ of all the strategies can be determined in many ways. This study uses the average of four random individuals generated by BBO as follows [316]:

$$\breve{T}_o = \frac{\text{sum}\,(ISI_1 \rightarrow ISI_4)}{4} \tag{2.65}$$

where $ISI_1 \rightarrow ISI_4$ are the 1^{st} four best individuals obtained from the BBO initialization stage.

Each cooling strategy has its inherent strengths and weaknesses, like scheduling

simplicity, solution quality, processing time, etc. This study focuses on five common types, which are briefly described as follows [284, 316, 345]:

Linear Cooling Rate

This is one of the simplest cooling strategies available in the literature. For the gth generation of BBO, the cooling temperature $\breve{T}_g$ of SA can be calculated as:

$$\breve{T}_g = \max\left(\breve{T}_o - \alpha \times g, \breve{T}_{\min}\right) \tag{2.66}$$

where $\breve{T}_{\min}$ is a very small temperature that is used to avoid reaching zero or a negative value [345]. It is taken as 10^{-5}. α is the slope of the linear decreasing rate and should be selected properly. We take it equal to:

$$\alpha = \frac{\breve{T}_o}{G} \tag{2.67}$$

where G is the total number of generations assigned to the BBO sub-algorithm.

Exponential Cooling Rate

It is one of the most popular cooling strategies. This type can balance between the simplicity, speed, and solution quality. It can be expressed as follows:

$$\breve{T}_g = c \times \breve{T}_{g-1} \tag{2.68}$$

where $\breve{T}_{g-1}$ equals $\breve{T}_o$ when $(g = 1)$, and the temperature reduction factor c is set in a range between 0.4 and 0.8 [316, 345].

Inverse Cooling Rate

M. Lundy and A. Mees in [229] proposed this strategy. The past temperature $\breve{T}_{g-1}$ is used to calculate the new temperature $\breve{T}_g$ as follows:

$$\breve{T}_g = \frac{\breve{T}_{g-1}}{[1 + \beta \times \breve{T}_{g-1}]} \tag{2.69}$$

where β is a small constant (typically, $0.001 \rightarrow 0.01$ [345]). We take this constant to be $\beta = 0.005$.

Inverse Linear Cooling Rate

This cooling strategy is easy to code. It was implemented to build fast simulated annealing (FSA) [358]. The rate is inversely linear in time, which can be mathematically expressed as follows:

$$\breve{T}_g = \frac{\breve{T}_o}{g} \tag{2.70}$$

Logarithmic Cooling Rate

This cooling strategy was introduced by Geman and Geman in [145]. Its mathematical formula is expressed as follows:

$$\breve{T}_g = \frac{\rho}{\ln(g+d)} \tag{2.71}$$

where d is a constant, which usually set equal to one. ρ is also a constant, which typically set equal to or greater than the largest energy barrier in the problem [284]. In MpBBO, it is taken as the largest ISI_i of the initial population:

$$\rho = \max(ISI_1 \rightarrow ISI_N) \tag{2.72}$$

Of course, there are other more advanced and complicated cooling strategies, such as the adaptive version presented in [284], but they are not covered in this dissertation. Figure 2.15 graphically shows these five cooling strategies.

2.4.3.2 Metropolis BBO

The original SA algorithm comes with only one individual[32], while BBO is a population-based algorithm. Both are probabilistic meta-heuristic optimizers that can converge to the space where the global optima is located in, but each algorithm has its own strengths and weaknesses. As stated before, the original BBO algorithm lacks exploration, while SA has a high probability to explore the other sides of the search space if the temperature $\breve{T}$ is still high [135]. In contrast, the initial temperature $\breve{T}_o$ and its cooling strategy can markedly affect the performance of SA, while BBO has less dependency on its own parameters.

[32] Because SA is a single-solution algorithm.

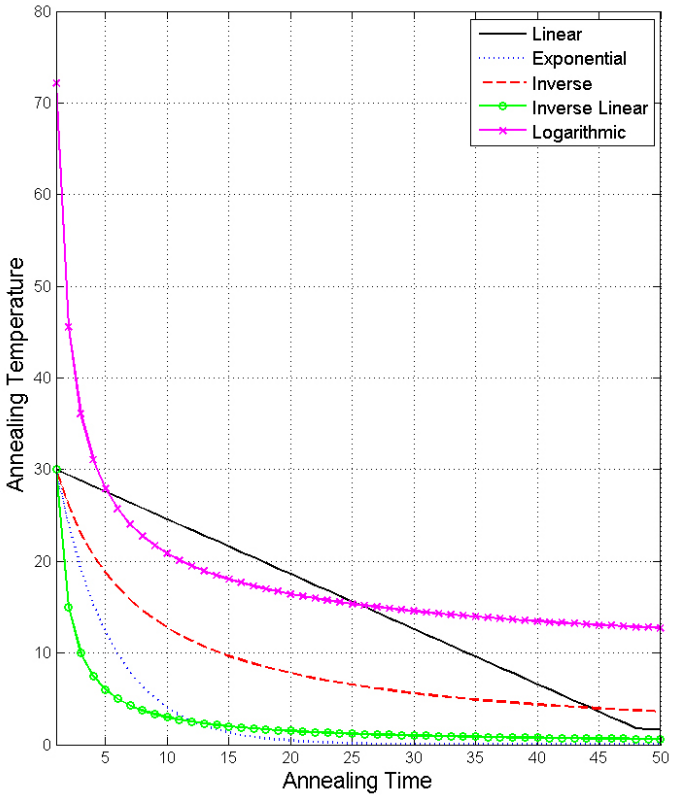


Figure 2.15: Five Different Cooling Strategies of SA

Based on these weaknesses and strengthes, instead of running SA with only one individual, it can be executed with multiple design points supplied from BBO. Thus, the internal searching loops l, within each cycle of z, can be disabled without affecting the solution quality of MpBBO. This approach can save a significant amount of CPU time.

The flowchart of this hybrid MpBBO algorithm is shown in Figure 2.16, and the pseudocode is given in Algorithm 8. It can be seen that the cooling rate of SA is updated at each new generation g of BBO until reaching G. Once the migration stage is completed, the features (n SIV) of the N islands will not be directly overwritten with the new values supplied from the probabilistically selected source islands. Instead, these n SIV of the N islands (before and after the migration stage) are saved in two temporary matrices (M_1 and M_2) with size $N \times n$. Each row of these two matrices represents one individual (before and after being modified). The old independent variables are used again if and only if the modified individual shows lower solution

quality and does not satisfy the Metropolis criterion. With this restriction on the migration stage, the overall performance can be enhanced.

This algorithm is the core of the most highly advanced hybrid optimization algorithm designed in this dissertation by employing the SQP algorithm as a fine tuner[33]. To justify this selection, the following extensive performance analysis is covered [31].

2.4.3.3 Performance Evaluations

The original BBO [43, 342], BBO-EM [44] and the proposed MpBBO models have been extensively tested using 36 benchmark functions to cover various types of difficulties, including unimodal/multimodal functions, few/many local optima, narrow/wide search spaces, problems with different dimensions, etc. Although there are many other benchmark functions available in the literature, this special set is carefully selected to have a fair performance comparison with BBO and BBO-EM as well as other BBO versions. Table 2.2 gives more details about these benchmark functions. Further information can be found in [27, 152, 187, 235, 389]. Also, for a fair processing time performance comparison, the three BBO algorithms are coded using MATLAB R2011a and the simulations are carried out using Intel Pentium E5300 2.60 GHz and 4GB RAM with 32-bit Windows XP SP3 operating system.

Table 2.3 shows the parameters used in the simulations of this comparison. These parameters are similar to those used in [233, 235], but with some restrictions on the generation limit. These restrictions are tabulated in Table 2.4.

Tables 2.5-2.6 show the best and mean values of these 36 benchmark functions when the algorithm is initiated with and without mutation stage, respectively. In addition, the actual and normalized average CPU time of the 30-dimensional problems are also shown at the bottom of each table. The bolded numbers indicate the best results achieved for each benchmark function.

As an overall observation, the first four fastest models are respectively BBO-EM, MpBBO (logarithmic type), MpBBO (linear type) and BBO. The other three MpBBO types are the slowest, but they show better performance in terms of solution quality (i.e., best and mean).

[33] This will be covered in the next subsection.

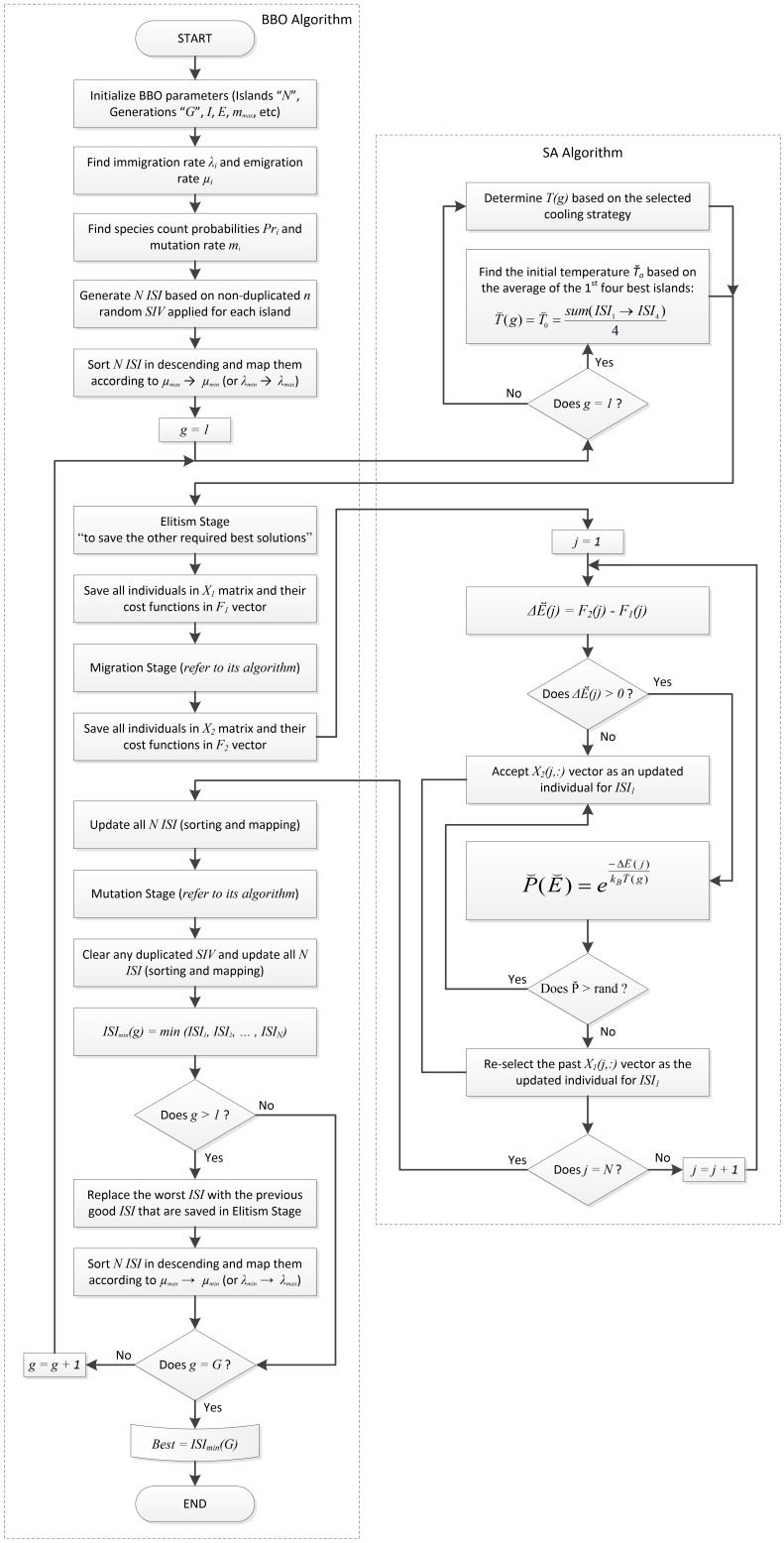


Figure 2.16: Flowchart of the Proposed Hybrid MpBBO Algorithm

Algorithm 8 Hybrid MpBBO Algorithm Pseudocode

Require: Initialization stage with all the parameters

Find λ_i and μ_i rates, then Pr_i and m_i rates
Generate N islands with unduplicated n SIV
Sort and map the population to the species count (i.e., $ISI_1 = ISI_{best}$ is coupled with $\mu_{\max}$ or $\lambda_{\min}$, and continue until reaching $ISI_N = ISI_{worst}$)
for $g \leftarrow 1$ *to* G **do** {where G = number of generations}
  if $g = 1$ **then**
    Find the initial temperature $\breve{T}_o$ based on the average of the 1^{st} four best solutions using (2.65)
  else
    Updated the temperature for the gth generation
  end if
  Save the required best solutions "elites" to be recycled again in the next generation
  Save the vectors of all the individuals (before migration) in a temporary matrix M_1 with size $N \times n$ and their cost functions in a temporary vector V_1 with length N
  Do migration (refer to Algorithm 2)
  Save the vectors of all the individuals (after migration) in a temporary matrix M_2 with size $N \times n$ and their cost functions in a temporary vector V_2 with length N
  for $i \leftarrow 1$ *to* N **do** {where N = number of islands or individuals}
    Calculate $\Delta\breve{E}_i = V_2(i) - V_1(i)$
    if $\Delta\breve{E}_i > 0$ **then**
      Apply the Metropolis criterion $\breve{P}(\breve{E}) = e^{\frac{-\Delta\breve{E}_i}{k_B \breve{T}_g}}$
      if $\breve{P}(\breve{E}) >$ rand **then**
        Accept $M_2(i, 1 \rightarrow n)$ vector of matrix M_2 as an updated individual for ISI_i
      else
        Re-select the past $M_1(i, 1 \rightarrow n)$ vector of matrix M_1 as an updated individual for ISI_i

Table 2.2: Some Characteristics of the Used Benchmark Functions (for more details, refer to [27, 152, 187, 235, 389])

$f\#$	Function Name	Dimension	Variables Bounds	Characteristics	Global Optimum
f01	Shekel's Function No. 1	1	$0 \le x \le 10$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	-10.1532
f02	Shekel's Function No. 2	1	$0 \le x \le 10$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	-10.4029
f03	Shekel's Function No. 3	1	$0 \le x \le 10$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	-10.5364
f04	Branin's RCOS Function	2	$-5 \le x_1 \le 10$ $0 \le x_2 \le 15$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	0.3978873
f05	Bukin's Function No. 6	2	$-15 \le x_i \le 5$	Continuous, Non-Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f06	Davis's Function	2	$-100 \le x_i \le 100$	Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal	0
f07	Goldstein-Price's Function	2	$-2 \le x_i \le 2$	Continuous, Differentiable, Non-separable, Non-Scalable, Multimodal	3
f08	Schaffer's Function No. 6	2	$-100 \le x_i \le 100$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f09	Schaffer's Function No. 7	2	$-100 \le x_i \le 100$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f10	Six-Hump Camel-Back Function	2	$-5 \le x_i \le 5$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	-1.0316285
f11	Box-Betts' Exponential Quadratic Sum Function	3	$0.9 \le x_1, x_3 \le 1.2$ $9 \le x_2 \le 11.2$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f12	Fletcher-Powell's Helical Valley Function	3	$-100 \le x_i \le 100$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f13	Meyer-Roth's Function	3	$-20 \le x_i \le 20$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0.00004
F14	Perm Function No. 1	3	$-n \le x_i \le n+1$	Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal	0
f15	Neumaier's Function No. 2	4	$0 \le x_i \le n$	Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal	0
f16	Wood's (or Colville's) Function	4	$-10 \le x_i \le 10$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f17	Perm Function No. 2	6	$-1 \le x_i \le 1$	Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal	0
f18	Price's Transistor Modelling Problem	9	$-10 \le x_i \le 10$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f19	Storn's Chebyshev Function	9	$-2^n \le x_i \le 2^n$	Continuous, Non-Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f20	Epistatic-Michalewicz's Function	10	$0 \le x_i \le \pi$	Continuous, Non-Differentiable, Non-Separable, Non-Scalable, Multimodal	-9.660152
f21	Neumaier's (or Trid's) Function No. 3	15	$-n^2 \le x_i \le n^2$	Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal	$-665\ (for\ n = 15)$
f22	Normalized Rana's Function with Diagonal Wrap	15	$-520 \le x_i \le 520$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	-512.753162426235 910068636786193
f23	Bent Cigar Function	17	$-100 \le x_i \le 100$	Continuous, Differentiable, Separable, Non-Scalable, Unimodal	0
f24	Alpine Function No. 1	20	$-10 \le x_i \le 10$	Continuous, Non-Differentiable, Separable, Non-Scalable, Multimodal	0
f25	Ackley's Function No. 1	30	$-32 \le x_i \le 32$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f26	Generalized Griewank's Function	30	$-600 \le x_i \le 600$	Continuous, Differentiable, Non-Separable, Scalable, Multimodal	0
f27	Generalized Penalized Function No. 1	30	$-50 \le x_i \le 50$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f28	Generalized Penalized Function No. 2	30	$-50 \le x_i \le 50$	Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal	0
f29	Generalized Rastrigin's Function	30	$-5.12 \le x_i \le 5.12$	Continuous, Differentiable, Separable, Scalable, Multimodal	0
f30	Generalized Rosenbrock's (or 2nd De Jong's) Function	30	$-30 \le x_i \le 30$	Continuous, Differentiable, Non-Separable, Scalable, Unimodal	0
f31	Generalized Schwefel's Function No. 2.26	30	$-500 \le x_i \le 500$	Continuous, Differentiable, Separable, Scalable, Multimodal	-12569.5
f32	Schwefel's Function No. 1.2	30	$-100 \le x_i \le 100$	Continuous, Differentiable, Non-Separable, Scalable, Unimodal	0
f33	Schwefel's Function No. 2.21	30	$-100 \le x_i \le 100$	Continuous, Non-Differentiable, Separable, Scalable, Unimodal	0
f34	Schwefel's Function No. 2.22	30	$-10 \le x_i \le 10$	Continuous, Differentiable, Non-Separable, Scalable, Unimodal	0
f35	Sphere Model (Square Sum, Harmonic or 1st De Jong's Function)	30	$-100 \le x_i \le 100$	Continuous, Differentiable, Separable, Scalable, Multimodal	0
f36	Step Function	30	$-100 \le x_i \le 100$	Discontinuous, Non-Differentiable, Separable, Scalable, Unimodal	0

```
        end if
      end if
    end for
    Update the population with sorting and mapping
    Do mutation (refer to Algorithm 3)
    Clear any duplicated SIV (refer to Algorithm 4)
    Update the population with sorting and mapping
    if g > 1 then
      Replace the worst ISI with the good ISI saved in the elitism stage
      Update the population with sorting and mapping
    end if
  end for
  Display the best individual
```

Table 2.3: Simulation Parameters of BBO (for more details, refer to [233, 235])

Parameter	**Value**
Population size "N"	50
$\lambda_{\max}$, or I	1
$\mu_{\max}$, or E	1
$m_{\max}$	0.01
Elitism "*elit*"	1
Mutation range "R_m"	$\text{round}\left[\left(\frac{N}{2}\right) : N\right]$
Number of trials "T_r"	30

Table 2.4: Total Number of Generations Assigned for Each n-Dimensional Problem (for more details, refer to [233, 235])

Dimension "n"	**Generations "G"**
1, 2, 3, 4 or 6	1,000
9 or 10	5,000
15, 17 or 20	10,000
30	20,000

Among these three powerful cooling strategies, Table 2.5 shows that MpBBO (inverse type) is the slowest type but it wins with the 15 lowest errors out of 36, and all these are multivariate problems with the following characteristics: continuous, differentiable, non-separable, and non-scalable. MpBBO (exponential type) is the fastest type and wins with the 12 best means out of 36, and all these are multivariate problems with the following characteristics: continuous, differentiable, and non-separable (especially for *low*-dimensional problems).

Table 2.6 presents the results obtained when the mutation stage is completely disabled. It shows that MpBBO (exponential type) is the second fastest type, but it does not win in any competition. In contrast, MpBBO (inverse linear type) is the fastest model and it occupies the best position in terms of the lowest errors and best means, especially when the problem dimensions are high. Because relatively few available problems are non-differentiable, separable, and unimodal, the inverse linear cooling strategy performs better in most of them.

2.4.3.4 Sensitivity Analysis

Similar to other EAs, the performance of MpBBO can be affected by changing its parameters, such as population size N, elitism $elit$, maximum mutation rate $m_{\max}$, mutation range R_m, cooling strategy type $\alpha(\breve{T})$ and its setting, etc. The effects of problem dimension n and search domain $SIV \in [SIV^{\min}, SIV^{\max}]$ have been partially covered through using different benchmark functions, as presented in the previous tables. However, selecting specific benchmark functions and changing their dimensions and domains have not been covered yet.

In this study, part of the sensitivity analysis is carried out by changing some parameters and then recording the results for some selected *high*-dimensional benchmark functions. For that, some selected 30-dimensional functions are reduced down to $n = 20$, and the other parameters are set as follows: $E = I = 1$, $N = 50$, $m_{\max} = 0.1$, $G = 100$, and $T_r = 50$. These new settings are essential to match the results presented in [234]. In Table 2.7, the effect of the temperature reduction factor c of the exponential cooling strategy is studied by incrementally increasing it from 0.1 to 0.9. It can be clearly seen that MpBBO with $c = 0.6$ scores better solutions than the others. Also, it can be noticed that the CPU time decreases as c increases because the acceptance probability of the Metropolis criterion slowly decreases and thus the old stored individuals will have less chance to be selected again. That is, if c is very large, then the algorithm will require high computational effort for convergence. Similarly, small c will have a quick reduction in $\breve{T}$, and so it will not have enough chance to explore the search space where the global optimum point is located in [316].

Optimal-trials plots shown in Figure 2.17 represent the good performance of $c = 0.6$ as compared with $c = 0.9$ and the original BBO algorithm. These plots show all the 50 simulation runs with the known optima of Schwefel F2.22 "f34", which is zero.

Based on this analysis, the same recommendation, which is applied to the original SA algorithm to select the good setting of c, is also applied here for the hybrid MpBBO algorithm where the recommended $c \in [0.4, 0.8]$.

Table 2.8 pertains to the performance of the most popular user-defined $m_{\max}$. Through this analysis, it has been found that MpBBO performs better when $m_{\max}$ is not very small (i.e., 0.005 or 0.001). The best-achieved results are those with 0.01, 0.05 and 0.1. The reason behind this phenomenon is that the mutation stage will compensate for the limited exploration of the algorithm due to small G. Also, the mutation stage is very useful when the algorithm is initiated with small N [234]. If $m_{\max}$ is very small, then the islands will have less chance to be mutated. Based on the last row of Table 2.8, it can be observed that $m_{\max}$ has less effect on the processing speed of MpBBO.

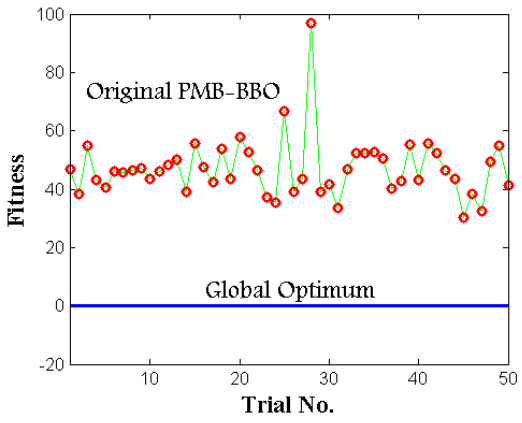
Moreover, the total number of good solutions stored in the elitism stage can affect the overall performance of EAs. Here, *elit* is set to 1, 2, 4 and 6. Two obvious

Table 2.5: Comparison of Results Over 30 Trials of BBO, BBO-EM, and MpBBO (with All Cooling Strategies), where "Best" Indicates the Smallest Error and "Mean" Indicates the Mean Smallest Error | Results Obtained with Mutation Stage

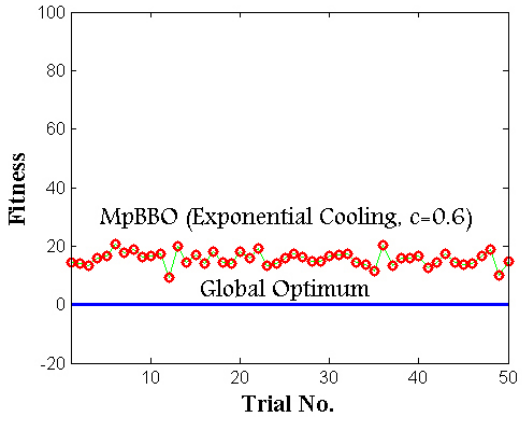
f #	Function Name	*n*	**Models of Biogeography Based Optimization (BBO)**													
			BBO (*Original Algorithm*)		**BBO-EM** (*Essential Mod.*)		**MpBBO** (*Lin. Cooling*)		**MpBBO** (*Exp. Cooling*)		**MpBBO** (*Inv. Cooling*)		**MpBBO** (*Inv. Lin. Cooling*)		**MpBBO** (*Log. Cooling*)	
			Best	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**
f01	**Shekel's F1**	**1**	9.8030E-08	**2.8723E-05**	6.7958E-09	1.2670E-04	1.2792E-07	9.3175E-05	**4.5658E-09**	7.1717E-05	5.4927E-08	**3.7442E-05**	4.6444E-08	9.8464E-05	8.2239E-07	6.5901E-05
f02	**Shekel's F2**	**1**	8.1958E-10	**2.1546E-05**	8.7502E-09	5.4862E-05	2.9993E-08	4.6136E-05	4.5054E-08	7.5239E-05	8.5946E-10	4.0305E-05	1.9153E-09	6.2271E-05	**2.9566E-11**	1.0450E-04
f03	**Shekel's F3**	**1**	1.7454E-08	7.4624E-05	**1.4736E-10**	5.1717E-05	1.3222E-09	5.0442E-05	7.1525E-08	4.5888E-05	2.9842E-08	7.7065E-05	1.2942E-07	8.3159E-05	3.2923E-10	**3.8964E-05**
f04	**Branin RCOS**	**2**	**1.3479E-08**	1.4782E-04	7.2838E-07	3.9066E-05	1.7225E-07	2.8790E-05	8.9146E-08	**1.0156E-05**	1.5827E-07	1.3605E-05	5.8539E-08	2.7602E-05	1.2918E-07	6.0161E-05
f05	**Bukin F6**	**2**	1.6136E-01	**5.6796E-01**	1.3459E-01	6.0400E-01	1.2796E-01	6.2086E-01	1.6539E-01	6.6671E-01	**5.6794E-02**	6.1200E-01	1.9105E-01	7.4028E-01	3.3047E-01	8.3166E-01
f06	**Davis**	**2**	1.1933E-01	4.2600E-01	9.6224E-02	2.6059E-01	9.3331E-02	2.6426E-01	9.4311E-02	**2.5336E-01**	8.6200E-02	2.6475E-01	**8.2447E-02**	2.6714E-01	8.6573E-02	2.6177E-01
f07	**Goldstein-Price**	**2**	2.4171E-05	2.0539E-03	2.9026E-06	2.8458E-04	3.2687E-06	**7.0741E-05**	5.8218E-06	1.2481E-04	**7.6000E-07**	1.2680E-04	9.0645E-07	1.2402E-04	1.5668E-06	1.5711E-04
f08	**Schaffer's F1**	**2**	1.6986E-04	8.8047E-03	2.2290E-04	8.8963E-03	4.1647E-04	1.0332E-02	1.4439E-04	**8.2780E-03**	7.7830E-05	9.0833E-03	6.0724E-05	8.7883E-03	**4.1241E-05**	9.0486E-03
f09	**Schaffer's F2**	**2**	7.9070E-01	7.1617E+00	4.8424E-01	3.5541E+00	4.2504E-01	2.8633E+00	**2.4840E-01**	**2.7785E+00**	6.7256E-01	3.1900E+00	5.1477E-01	2.9872E+00	8.1821E-01	3.1857E+00
f10	**Six-Hump Camel-Back**	**2**	2.4565E-06	9.0249E-05	2.4226E-07	1.2621E-05	**2.3588E-09**	9.4335E-06	5.3001E-09	**7.1553E-06**	9.6023E-09	7.5944E-06	8.7682E-08	1.2308E-05	3.7546E-08	1.5989E-05
f11	**Box-Betts**	**3**	1.0983E-07	1.7811E-06	2.9705E-09	1.0045E-06	**8.3837E-10**	7.7080E-07	5.9157E-09	6.1565E-07	4.3097E-09	**4.5058E-07**	2.8186E-09	1.1693E-06	1.0842E-08	9.7063E-07
f12	**Fletcher-Powell**	**3**	3.4313E-02	9.5005E-01	4.3329E-02	6.7905E-01	4.5686E-03	5.8831E-01	**3.7190E-05**	5.5891E-01	1.3868E-03	**3.6868E-01**	3.2031E-03	4.0727E-01	1.2490E-02	6.3267E-01
f13	**Meyer-Roth**	**3**	1.6692E-05	1.1214E-04	3.8931E-06	5.7958E-05	4.8535E-06	4.9632E-05	**3.6228E-06**	3.1875E-05	3.6505E-06	4.1105E-05	3.9714E-06	**2.6115E-05**	8.2571E-06	6.0700E-05
f14	**Perm F1**	**3**	1.5078E-01	1.5101E+00	7.3013E-03	7.1294E-01	1.0011E-02	**2.9541E-01**	8.2502E-03	5.2292E-01	4.2894E-02	6.0202E-01	2.3265E-02	5.4293E-01	**4.8219E-03**	5.7654E-01
f15	**Neumaier F2**	**4**	8.2399E-04	4.6832E-02	5.5432E-03	3.7898E-02	6.8348E-04	**1.7225E-02**	1.9563E-03	1.9806E-02	**4.4320E-04**	2.6200E-02	6.3128E-04	2.7324E-02	8.2366E-04	2.8112E-02
f16	**Wood**	**4**	4.1174E-01	5.8646E+00	7.3063E-02	2.1807E+00	1.1238E-01	1.2566E+00	2.3763E-02	9.5017E-01	**3.4534E-03**	7.4769E-01	2.3123E-02	**6.5929E-01**	7.8028E-02	1.6814E+00
f17	**Perm F2**	**6**	8.1747E-02	1.1744E+00	2.6251E-02	4.5568E-01	7.0748E-03	2.2006E-01	6.2589E-03	**9.7313E-02**	2.6861E-03	1.4004E-01	**2.0117E-03**	1.1422E-01	8.7497E-03	4.8628E-01
f18	**Price's Transistor**	**9**	4.3225E+01	1.3169E+02	1.9152E+01	8.7268E+01	1.3647E+01	7.4791E+01	3.6287E+00	**3.7751E+01**	**3.5512E+00**	6.1102E+01	4.5631E+00	5.8234E+01	1.1937E+01	7.8802E+01
f19	**Storn's Chebyshev**	**9**	5.5986E+03	5.3834E+04	3.8343E+03	2.8105E+04	6.3970E+02	1.1658E+04	**3.7883E+02**	9.0081E+03	1.7795E+03	9.2786E+03	1.2831E+03	**8.5535E+03**	3.2508E+03	1.7771E+04
f20	**Epistatic Michalewicz**	**10**	3.0295E-01	1.4400E+00	1.6545E-01	7.1945E-01	1.1310E-01	4.0222E-01	1.4659E-01	6.5636E-01	**1.1911E-02**	3.3862E-01	2.2429E-01	6.7894E-01	4.7910E-02	**3.1856E-01**
f21	**Neumaier's F3**	**15**	9.6796E+01	7.4986E+02	6.8391E+01	6.4874E+02	2.7514E+01	4.5419E+02	2.3930E+01	3.9175E+02	**2.2544E+01**	3.3659E+02	3.7623E+01	**3.2320E+02**	5.7175E+01	5.2766E+02
f22	**Normalized Rana**	**15**	3.5229E+01	5.7506E+01	2.7209E+01	4.7915E+01	1.5022E+01	3.7871E+01	2.9219E+01	4.9152E+01	1.8431E+01	**3.3275E+01**	3.1452E+01	4.8610E+01	**1.2628E+01**	3.6421E+01
f23	**Bent Cigar**	**17**	1.0365E+06	2.8255E+06	4.5193E+05	1.6921E+06	1.6289E+05	6.9259E+05	2.7386E+04	**8.2069E+04**	**2.4297E+04**	8.3220E+04	2.8597E+04	1.0680E+05	2.7901E+05	7.6936E+05
f24	**Alpine F1**	**20**	1.7996E-02	3.5168E-02	1.6133E-02	3.0236E-02	1.0680E-02	1.7567E-02	4.3505E-03	7.4793E-03	5.2893E-03	7.7997E-03	**4.2145E-03**	**7.1635E-03**	1.2933E-02	1.8451E-02
f25	**Ackley F1**	**30**	5.0440E-01	9.5527E-01	5.4005E-01	8.7225E-01	2.4359E-01	4.1283E-01	**1.1522E-01**	2.4615E-01	1.6084E-01	**2.4186E-01**	1.2501E-01	2.5120E-01	1.2823E-01	2.5494E-01
f26	**Gen. Griewank**	**30**	1.0171E+00	1.0359E+00	9.6698E-01	1.0283E+00	6.9734E-01	9.8975E-01	2.4499E-01	5.4691E-01	**2.3992E-01**	**4.8687E-01**	2.6890E-01	5.0443E-01	7.0725E-01	9.7785E-01
f27	**Gen. Penalized F1**	**30**	2.4176E-03	2.1480E-02	3.7749E-03	1.9135E-02	1.8012E-03	1.3496E-02	4.9452E-04	**2.3224E-03**	**3.2535E-04**	3.3505E-03	4.3558E-04	3.8194E-03	1.5888E-03	1.5323E-02
f28	**Gen. Penalized F2**	**30**	8.9980E-02	1.9327E-01	7.0905E-02	1.5673E-01	5.7889E-02	1.0227E-01	1.4730E-02	2.8000E-02	**8.1362E-03**	**2.2819E-02**	9.3627E-03	2.7055E-02	4.6061E-02	1.2621E-01
f29	**Gen. Rastrigin**	**30**	6.5128E-01	1.8602E+00	3.1743E-01	1.6697E+00	5.6380E-01	1.2139E+00	**8.5799E-02**	2.7572E-01	1.3166E-01	3.0427E-01	1.0359E-01	**2.7312E-01**	4.1063E-01	9.8147E-01
f30	**Gen. Rosenbrock**	**30**	1.1917E+02	3.6900E+02	1.2684E+02	2.7991E+02	1.2731E+02	2.2709E+02	5.7755E+01	1.5947E+02	**4.9396E+01**	**1.3532E+02**	8.1867E+01	1.3700E+02	9.5496E+01	2.2052E+02
f31	**Gen. Schwefel F2.26**	**30**	4.7161E-06	1.3331E-05	5.8240E-06	1.0469E-05	**9.1197E-07**	**1.8097E-06**	5.5940E-06	1.0189E-05	9.7074E-07	2.2755E-06	5.5268E-06	1.1937E-05	4.3762E-06	1.1737E-05
f32	**Schwefel F1.2**	**30**	2.1099E+01	2.2380E+04	2.6721E-02	5.8748E+02	2.8464E-04	1.2659E+00	2.7293E-03	2.1017E-01	**4.4868E-05**	**2.0580E-01**	1.8337E-03	4.4117E-01	1.9550E-04	1.2775E+01
f33	**Schwefel F2.21**	**30**	4.2917E+00	6.2916E+00	4.3242E+00	5.9236E+00	2.1375E+00	3.0114E+00	1.6843E+00	**2.3277E+00**	**1.4924E+00**	2.3430E+00	1.6288E+00	2.3397E+00	1.9004E+00	2.3942E+00
f34	**Schwefel F2.22**	**30**	4.3794E-01	7.2348E-01	4.7622E-01	6.9053E-01	4.5948E-01	6.9143E-01	**1.6844E-01**	**2.4473E-01**	1.8137E-01	2.5768E-01	4.2496E-01	6.3531E-01	3.7673E-01	6.7230E-01
f35	**Sphere**	**30**	1.3828E+00	3.9501E+00	1.9249E+00	3.3150E+00	1.1898E+00	2.1108E+00	2.2796E-01	**5.2130E-01**	2.6098E-01	5.9174E-01	**1.6561E-01**	5.6390E-01	1.0482E+00	2.4038E+00
f36	**Step**	**30**	**0.0000E+00**	5.0333E+00	2.0000E+00	4.4000E+00	**0.0000E+00**	2.4333E+00	**0.0000E+00**	1.6667E-01	**0.0000E+00**	2.6667E-01	**0.0000E+00**	**6.6667E-02**	**0.0000E+00**	2.4333E+00
Total Winning out of 36 Benchmarks			2/36	3/36	1/36	0/36	4/36	4/36	9/36	**12/36**	**15/36**	8/36	5/36	7/36	5/36	2/36
Sorted Best Algorithms			5	5	6	7	4	4	2	**1**	**1**	2	3	3	3	6
Avg. CPU Time of 30-dimen. Functions (sec)			7064.690161		**5413.311339**		6226.92755		7251.164848		7549.948712		7067.67142		6060.828061	
Normalized Avg. CPU Time			1.305058904		**1.000000000**		1.150299172		1.339506338		1.394700626		1.305609631		1.119615644	
Sorted Fastest Algorithms			4		**1**		3		5		7		6		2	

Table 2.6: Comparison of Results Over 30 Trials of BBO, BBO-EM, and MpBBO (with All Cooling Strategies), where "Best" Indicates the Smallest Error and "Mean" Indicates the Mean Smallest Error | Results Obtained without Mutation Stage

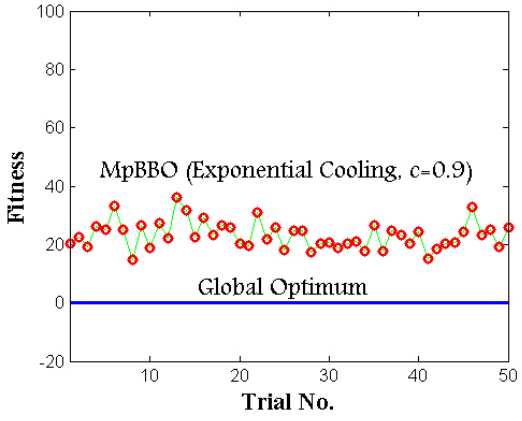
f #	Function Name	*n*	**Models of Biogeography Based Optimization (BBO)**													
			BBO (*Original Algorithm*)		**BBO-EM** (*Essential Mod.*)		**MpBBO** (*Lin. Cooling*)		**MpBBO** (*Exp. Cooling*)		**MpBBO** (*Inv. Cooling*)		**MpBBO** (*Inv. Lin. Cooling*)		**MpBBO** (*Log. Cooling*)	
			Best	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**	**Best**	**Mean**
f01	**Shekel's F1**	**1**	1.2792E-06	6.2952E-05	4.5057E-10	6.5586E-05	3.1393E-08	5.9747E-05	2.7764E-08	**1.9745E-05**	1.2353E-08	5.8074E-05	1.2470E-09	6.9317E-05	**1.3120E-11**	5.6035E-05
f02	**Shekel's F2**	**1**	7.7958E-05	1.7073E-04	6.3663E-08	**4.0937E-05**	4.3003E-08	8.5795E-05	**1.6675E-10**	4.3676E-05	7.2713E-09	5.2726E-05	1.4189E-08	7.1035E-05	1.6937E-10	5.6087E-05
f03	**Shekel's F3**	**1**	1.0971E-04	1.4089E-04	2.4156E-08	**5.6657E-05**	9.5185E-09	9.2273E-05	5.4983E-09	7.5254E-05	3.3648E-09	8.6176E-05	1.5519E-08	6.7668E-05	**2.3705E-11**	6.4634E-05
f04	**Branin RCOS**	**2**	1.0244E-05	1.0348E-03	3.6317E-07	1.1647E-04	1.3276E-07	1.3557E-04	4.1900E-07	**5.8606E-05**	5.9487E-07	8.3328E-05	1.8052E-07	6.9156E-05	**1.0788E-07**	8.9971E-05
f05	**Bukin F6**	**2**	2.1359E-01	7.1971E-01	1.6522E-01	**5.9739E-01**	1.1840E-01	8.2443E-01	**6.8858E-02**	9.6474E-01	1.6341E-01	9.2343E-01	2.1570E-01	7.3037E-01	1.1991E-01	8.2943E-01
f06	**Davis**	**2**	2.9895E-01	7.8963E-01	1.3955E-01	3.3957E-01	**1.0146E-01**	**2.8033E-01**	1.2791E-01	3.4486E-01	1.2788E-01	3.1197E-01	1.1339E-01	3.4690E-01	1.4343E-01	3.3791E-01
f07	**Goldstein-Price**	**2**	7.3064E-04	3.2783E-02	8.2171E-06	8.4074E-04	5.5960E-07	3.7902E-04	3.2724E-06	6.5812E-04	1.2188E-05	5.4835E-04	3.1667E-05	**3.6463E-04**	**2.6082E-06**	6.8238E-04
f08	**Schaffer's F1**	**2**	9.7184E-03	1.1890E-02	4.2863E-05	**8.3314E-03**	8.1000E-04	1.0135E-02	**8.2397E-06**	9.0989E-03	2.3053E-03	9.3572E-03	2.5060E-04	8.6719E-03	1.0433E-03	9.1775E-03
f09	**Schaffer's F2**	**2**	6.6683E+00	2.7006E+01	1.3900E+00	5.9639E+00	1.7104E+00	4.7635E+00	1.3854E+00	**4.6045E+00**	1.3064E+00	4.7721E+00	1.4663E+00	5.2466E+00	**1.2722E+00**	5.1761E+00
f10	**Six-Hump Camel-Back**	**2**	3.5677E-05	1.7114E-03	3.3394E-08	4.7417E-05	5.0113E-07	2.7885E-05	9.5790E-07	**2.7460E-05**	**1.1476E-08**	4.2043E-05	2.7822E-07	3.9947E-05	1.7341E-07	2.9394E-05
f11	**Box-Betts**	**3**	5.7383E-07	4.8765E-06	5.0070E-08	1.0132E-06	1.2444E-08	9.7841E-07	1.0131E-08	9.0830E-07	4.0329E-09	9.6947E-07	3.7406E-09	**8.2585E-07**	**2.9823E-10**	1.2139E-06
f12	**Fletcher-Powell**	**3**	1.3013E-01	2.3318E+00	3.0329E-02	1.2599E+00	1.8164E-03	5.9488E-01	8.6890E-03	4.3786E-01	6.5240E-03	**4.0291E-01**	**9.7299E-04**	4.3605E-01	1.1576E-02	5.9521E-01
f13	**Meyer-Roth**	**3**	1.1476E-05	1.7993E-04	8.6944E-06	9.3672E-05	4.2117E-06	6.0840E-05	7.2533E-06	7.3098E-05	5.0438E-06	**4.3049E-05**	3.7845E-06	7.2549E-05	**3.6998E-06**	6.3515E-05
f14	**Perm F1**	**3**	2.4008E-01	1.8853E+00	4.1883E-02	6.5424E-01	4.9392E-03	**6.2194E-01**	1.6447E-02	8.3016E-01	7.2341E-04	7.7941E-01	**5.7667E-04**	8.5173E-01	6.4282E-02	1.0509E+00
f15	**Neumaier F2**	**4**	1.5871E-03	7.2832E-02	5.4291E-03	4.0537E-02	2.0018E-03	2.6573E-02	**6.0366E-06**	2.8166E-02	9.9638E-04	**2.6019E-02**	1.4924E-04	2.8332E-02	3.0819E-03	4.3248E-02
f16	**Wood**	**4**	2.3861E+00	9.5390E+00	1.0681E-01	2.7161E+00	1.3611E-01	**8.2357E-01**	5.3312E-02	9.1372E-01	3.9603E-02	8.6874E-01	**3.5311E-02**	8.7164E-01	1.7260E-01	1.6285E+00
f17	**Perm F2**	**6**	1.7856E-01	1.3030E+00	3.3215E-02	5.3748E-01	9.6604E-03	1.8420E-01	1.6404E-02	1.8594E-01	**3.9880E-03**	1.6018E-01	4.0767E-03	**1.0176E-01**	6.2241E-03	4.2440E-01
f18	**Price's Transistor**	**9**	5.1972E+01	1.5102E+02	1.4909E+01	7.3385E+01	1.5169E+01	6.3543E+01	3.2464E+00	4.0687E+01	2.1382E+00	4.7764E+01	**2.1201E+00**	**3.9843E+01**	1.0938E+01	8.3679E+01
f19	**Storn's Chebyshev**	**9**	1.3977E+04	7.0027E+04	1.3640E+03	3.1546E+04	9.7372E+02	1.2455E+04	6.7711E+02	1.4081E+04	8.7437E+02	**9.6805E+03**	**1.4060E+02**	1.1564E+04	1.2578E+03	2.2427E+04
f20	**Epistatic Michalewicz**	**10**	7.3670E-01	1.8089E+00	2.2486E-01	6.9678E-01	4.5582E-02	4.0743E-01	1.1730E-01	7.1714E-01	1.3924E-02	3.7426E-01	1.7023E-01	6.3890E-01	**1.7326E-03**	**3.6518E-01**
f21	**Neumaier's F3**	**15**	1.5497E+02	1.6674E+03	2.6501E+01	6.6435E+02	4.2489E+01	5.9656E+02	2.3360E+01	2.9062E+02	1.6354E+01	3.3416E+02	**1.0095E+01**	**2.7414E+02**	4.8035E+01	7.2274E+02
f22	**Normalized Rana**	**15**	6.4485E+01	8.4083E+01	3.2995E+01	4.9462E+01	**1.6996E+01**	**3.3146E+01**	2.3546E+01	4.7175E+01	1.9504E+01	3.4756E+01	2.9521E+01	4.8818E+01	1.9983E+01	3.5992E+01
f23	**Bent Cigar**	**17**	2.2232E+06	1.2807E+07	6.0622E+05	1.7363E+06	1.9250E+05	5.6584E+05	**2.9735E+04**	**9.1113E+04**	3.1485E+04	1.0528E+05	4.1024E+04	9.8092E+04	2.0747E+05	7.8675E+05
f24	**Alpine F1**	**20**	3.3147E-02	1.1369E-01	1.3182E-02	2.4450E-02	1.1316E-02	1.9888E-02	4.9591E-03	7.9083E-03	4.6964E-03	**7.6481E-03**	**4.2689E-03**	7.8347E-03	1.0720E-02	1.8811E-02
f25	**Ackley F1**	**30**	1.1313E+00	1.7229E+00	5.5410E-01	8.8601E-01	2.7181E-01	4.4747E-01	1.5863E-01	2.4741E-01	**1.2950E-01**	**2.3865E-01**	1.7568E-01	2.6694E-01	1.7993E-01	2.6056E-01
f26	**Gen. Griewank**	**30**	1.0526E+00	1.1071E+00	9.8945E-01	1.0300E+00	8.3920E-01	9.9064E-01	**2.1883E-01**	5.4254E-01	3.2631E-01	5.4866E-01	2.5750E-01	**5.2317E-01**	8.2874E-01	9.7670E-01
f27	**Gen. Penalized F1**	**30**	1.6841E-02	8.4679E-02	2.6843E-03	1.5289E-02	2.4511E-03	1.5895E-02	3.7031E-04	3.2949E-03	3.8704E-04	2.9407E-03	**2.3954E-04**	**2.4352E-03**	2.6132E-03	1.7918E-02
f28	**Gen. Penalized F2**	**30**	2.2517E-01	5.0591E-01	7.1096E-02	1.7506E-01	4.1930E-02	1.0537E-01	**5.9819E-03**	2.5890E-02	8.8468E-03	2.8301E-02	6.0067E-03	**2.3626E-02**	3.9681E-02	1.2496E-01
f29	**Gen. Rastrigin**	**30**	2.3320E+00	4.5176E+00	6.9811E-01	1.6054E+00	5.5688E-01	1.1110E+00	1.2048E-01	2.4790E-01	**9.1460E-02**	**2.4250E-01**	1.1436E-01	2.5245E-01	4.2503E-01	1.0369E+00
f30	**Gen. Rosenbrock**	**30**	1.7036E+02	6.0492E+02	1.5761E+02	3.7445E+02	1.0882E+02	2.4272E+02	6.6832E+01	1.3282E+02	4.1739E+01	1.5396E+02	**2.9987E+01**	**1.3224E+02**	5.8013E+01	2.2552E+02
f31	**Gen. Schwefel F2.26**	**30**	8.1593E-06	2.9257E-05	5.0031E-06	1.0870E-05	**6.4893E-07**	**1.6534E-06**	5.4447E-06	1.1479E-05	1.0224E-06	2.2710E-06	3.0467E-06	9.5701E-06	4.8305E-06	1.1704E-05
f32	**Schwefel F1.2**	**30**	2.9496E+04	3.5713E+04	1.8231E-02	2.1922E+02	1.3907E-04	1.0427E+00	1.1061E-03	1.8730E+00	1.8993E-04	**3.7804E-01**	**4.7571E-05**	4.9482E-01	2.3814E-03	1.1927E+01
f33	**Schwefel F2.21**	**30**	7.4583E+00	1.0220E+01	4.1899E+00	5.4466E+00	2.2883E+00	3.1611E+00	1.4267E+00	2.2725E+00	1.6049E+00	2.2490E+00	**1.3503E+00**	**2.2194E+00**	1.5892E+00	2.2875E+00
f34	**Schwefel F2.22**	**30**	8.0283E-01	1.2265E+00	5.0707E-01	6.8179E-01	4.6629E-01	6.8265E-01	1.8565E-01	2.6015E-01	**1.6793E-01**	**2.5792E-01**	4.5310E-01	6.8001E-01	3.5526E-01	7.0024E-01
f35	**Sphere**	**30**	4.1094E+00	1.1390E+01	1.8662E+00	3.4693E+00	7.8744E-01	2.5476E+00	2.6718E-01	6.3357E-01	2.7671E-01	5.7382E-01	**2.5718E-01**	**5.1227E-01**	9.1562E-01	2.1552E+00
f36	**Step**	**30**	4.0000E+00	1.3533E+01	1.0000E+00	4.1333E+00	**0.0000E+00**	2.8667E+00	**0.0000E+00**	2.3333E-01	**0.0000E+00**	**1.3333E-01**	**0.0000E+00**	1.6667E-01	**0.0000E+00**	2.3000E+00
Total Winning out of 36 Benchmarks			0/36	0/36	0/36	4/36	4/36	5/36	8/36	5/36	6/36	10/36	**13/36**	**11/36**	9/36	1/36
Sorted Best Algorithms			6	6	6	4	5	3	3	3	4	2	**1**	**1**	2	5
Avg. CPU Time of 30-dimen. Functions (sec)			6607.419593		**4716.098121**		5573.137696		6794.500931		6913.270273		6634.153229		5531.929964	
Normalized Avg. CPU Time			1.401035225		**1.000000000**		1.181726409		1.440703895		1.46588771		1.406703817		1.172988734	
Sorted Fastest Algorithms			4		**1**		3		6		7		5		2	



(a)



(b)



(c)

Figure 2.17: Optimal-Trials Plots of 50 Independent Simulation Runs of Schwefel Function No. 2.22 "f34". (a) Original PMB-BBO, (b) MpBBO - Exponential Cooling Rate with $c = 60\%$, (c) MpBBO - Exponential Cooling Rate with $c = 90\%$

Table 2.7: Effect of the Temperature Reduction Factor c on the Best Results of MpBBO (Exponential Type)

f #	Function Name	n	c = 0.1	c = 0.2	c = 0.3	c = 0.4	c = 0.5	c = 0.6	c = 0.7	c = 0.8	c = 0.9
f25	**Ackley F1**	**20**	9.96E+00	9.67E+00	9.92E+00	9.32E+00	1.02E+01	9.37E+00	1.03E+01	**8.59E+00**	9.22E+00
f26	**Gen. Griewank**	**20**	1.40E+01	1.66E+01	1.74E+01	1.42E+01	1.47E+01	**8.01E+00**	1.18E+01	1.41E+01	1.38E+01
f29	**Gen. Rastrigin**	**20**	5.08E+01	4.84E+01	4.70E+01	4.37E+01	5.11E+01	5.59E+01	**4.05E+01**	4.81E+01	4.82E+01
f30	**Gen. Rosenbrock**	**20**	8.62E+01	8.59E+01	1.03E+02	9.21E+01	9.44E+01	**7.94E+01**	1.02E+02	1.18E+02	1.04E+02
f32	**Schwefel F1.2**	**20**	4.09E-01	4.13E-01	**4.71E-04**	1.25E+00	2.44E-01	5.42E-01	9.64E-01	5.40E+00	1.04E+00
f33	**Schwefel F2.21**	**20**	2.69E+01	2.57E+01	2.51E+01	2.56E+01	2.54E+01	**2.29E+01**	2.36E+01	3.22E+01	2.88E+01
f34	**Schwefel F2.22**	**20**	9.93E+00	1.15E+01	9.54E+00	1.03E+01	1.12E+01	**9.43E+00**	1.28E+01	1.50E+01	1.48E+01
f35	**Sphere**	**20**	4.39E+00	4.31E+00	**3.02E+00**	3.37E+00	3.68E+00	4.00E+00	3.15E+00	3.66E+00	4.33E+00
f36	**Step**	**20**	1.03E+03	1.43E+03	1.70E+03	1.51E+03	1.17E+03	9.64E+02	**8.52E+02**	1.74E+03	1.81E+03
Avg. CPU Time (sec)			40.94	40.88	40.98	40.78	40.81	40.79	40.48	40.14	**39.50**

Table 2.8: Effect of the User-Defined Maximum Mutation Rate $m_{\max}$ on the Best Results of MpBBO

f #	**0.1**	**0.05**	**0.01**	**0.005**	**0.001**
f25	9.37E+00	1.00E+01	**8.04E+00**	1.00E+01	1.01E+01
f26	**8.01E+00**	1.20E+01	1.24E+01	1.50E+01	1.42E+01
f29	5.59E+01	5.13E+01	4.74E+01	5.23E+01	**4.73E+01**
f30	7.94E+01	8.21E+01	9.50E+01	**4.80E+01**	6.26E+01
f32	5.42E-01	5.56E-01	**4.04E-02**	7.30E-01	6.01E+00
f33	2.29E+01	2.52E+01	**2.24E+01**	2.63E+01	3.02E+01
f34	**9.43E+00**	1.11E+01	1.06E+01	1.10E+01	1.19E+01
f35	4.00E+00	3.78E+00	**2.69E+00**	3.29E+00	3.85E+00
f36	**9.64E+02**	1.16E+03	1.45E+03	1.65E+03	1.32E+03
CPU (s)	40.79	40.42	**40.33**	40.63	40.65

things can be noticed from Table 2.9: the solution quality is enhanced by selecting non-small or large *elit*, and the CPU time is proportional to *elit*. It is known that the good selection of *elit* depends on other parameters, such as n, N, and G. But in general, small *elit* (like 0 or 1) will force optimization algorithms to lose useful information contained in some good individuals that might guide them to settle on better solutions. Similarly, large *elit* will reduce the exploration level by premature convergence and thus the performance will worsen [79, 346].

2.4.3.5 MpBBO vs. Non-Simplified BBO Models

By referring to the literature of BBO, it is known that the types of the immigration rate λ_i and the emigration rate μ_i play an important role in the selection probability

Table 2.9: Effect of the Elitism Parameter *elit* on the Best Results of MpBBO

f #	1	2	4	6
f25	8.04E+00	**7.87E+00**	9.18E+00	8.65E+00
f26	1.24E+01	**7.85E+00**	1.05E+01	9.46E+00
f29	4.74E+01	4.30E+01	**3.85E+01**	4.41E+01
f30	9.49E+01	8.58E+01	8.27E+01	**6.92E+01**
f32	4.04E-02	**1.51E-03**	8.52E-01	1.51E-03
f33	2.24E+01	2.67E+01	**2.15E+01**	2.49E+01
f34	1.06E+01	9.55E+00	8.99E+00	**8.70E+00**
f35	**2.69E+00**	3.24E+00	2.91E+00	2.85E+00
f36	1.45E+03	1.28E+03	**1.06E+03**	8.22E+02
CPU (s)	**40.33**	42.96	44.52	44.63

of the rich and poor islands, which in turn affect the overall performance of any BBO algorithm [233, 234, 236]. Thus, selecting good rate functions for λ_i and μ_i could increase the chance to get better solutions. These non-simplified migration rates are not incorporated with MpBBO[34]. However, it is highly interesting to compare the performance of MpBBO with other BBO models equipped with some advanced migration rates.

Table 2.10 shows the results achieved by MpBBO for some selected benchmark functions. The parameters here are similar to [233] where: $E = I = 1$, $N = 50$, $m_{\max} = 0.01$, $n = 20$, $G = 10,000$ and $T_r = 50$. The actual results of these 6 BBO models are available in [233]. From Table 2.10, the proposed MpBBO algorithm shows better performance compared with the first five models, and it shares the first position with the last model where both achieve the same score for the step function. Also, it can be observed that MpBBO is the winner against all the models for the Schwefel F1.2 "f32" and Sphere "f35" benchmark functions.

2.4.3.6 MpBBO vs. Other EAs

One of the interesting things here is to make a comparison between the proposed MpBBO algorithm and other EAs, especially those hybrid versions that contain SA as an essential part of their structures. From the literature, many hybrid models are

[34]They could be considered in future work.

Table 2.10: Comparison of the Best Results Over 50 Trials of MpBBO and BBO with Six Migration Models. (a) model1: constant λ_i and linear μ_i, (b) model2: linear λ_i and constant μ_i, (c) model3: linear λ_i and μ_i, (d) model4: trapezoidal λ_i and μ_i, (e) model5: quadratic λ_i and μ_i, (f) model6: sinusoidal λ_i and μ_i. The values of these six models can be found in [233]. The 0s and 1s mean whether MpBBO scored better position or not, where the left hand side digit is for MpBBO and the right hand side digit is for the models (i.e., if the result shows 1-0, this means that MpBBO is the winner and vice versa)

f #	Function Name	*n*	MpBBO	vs. mdel1	vs. mdel2	vs. mdel3	vs. mdel4	vs. mdel5	vs. mdel6
f25	**Ackley F1**	**20**	1.0204E-01	**1 - 0**	0 - 1	0 - 1	**1 - 0**	**1 - 0**	0 - 1
f26	**Gen. Griewank**	**20**	2.0654E-01	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**	0 - 1
f29	**Gen. Rastrigin**	**20**	3.2696E-02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**
f30	**Gen. Rosenbrock**	**20**	5.2682E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**
f32	**Schwefel F1.2**	**20**	1.9426E-09	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f33	**Schwefel F2.21**	**20**	9.5348E-01	**1 - 0**	0 - 1	0 - 1	**1 - 0**	0 - 1	0 - 1
f34	**Schwefel F2.22**	**20**	8.1196E-02	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1
f35	**Sphere**	**20**	2.9060E-04	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f36	**Step**	**20**	0.0000E+00	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	1 - 1
Total Winning out of 9 Benchmarks				**9/0**	**7/2**	**5/4**	**7/2**	**5/4**	5/5

presented as effective optimizers for solving some numerical problems. For example, the hybrid CGP/SA algorithm given in [135] is used for the data mining task of a data classification problem, while the hybrid SA/TS algorithm given in [203] is used for the optimal sizing problem of small autonomous power systems (SAPS). However, there are also many other studies conducted based on some selected standard benchmark functions, such as [91, 327, 338].

Table 2.11 shows the comparison between the results of MpBBO and 10 different types of EAs taken from [327]. For this part of the study, some parameters are modified to match [327]. From Table 2.11, it is obvious that MpBBO is the winner in all the performance criteria (best, mean and median) for Rastrigin "f29" and Sphere "f35" (multimodal and separable functions). For Rosenbrock "f30" (unimodal and non-separable function), MpBBO scores the second position for the best and median results and the third position for the mean result.

Also, with the same initialization parameters of [45], the performance of MpBBO

is evaluated against **bat algorithm** (**BA**) and **bacterial foraging optimization** (**BFO**) algorithm using 9 different benchmark functions. As can be observed in Table 2.12, the competition is mainly between BFO and MpBBO. For some benchmark functions, BFO can reach better solutions and sometimes even with better worst and mean errors as with Egg Crate, Beale, and Booth functions. However, BFO is prone to trap into local optima. This main drawback can be easily highlighted in the Rastrigin, Bartels-Conn, and Bohachevsky benchmark functions. In contrast, MpBBO could lose in some benchmark functions, but the gaps between the bests, worsts, and means are very small compared with BFO. For example, BFO scores 4.61E-07 with Bohachevsky Function No.3, but the average is 3.10E+01, which is very big. A similar thing can be seen with Bartels-Conn Function.

Table 2.13 shows another performance comparison between MpBBO and six other types of EAs listed as follows [90]: standard PSO (SPSO), linear decreasing inertia weight PSO (LDWPSO), quantum-behaved PSO (QPSO), **teaching-learning-based optimization** (**TLBO**), elitism-based TLBO (ETLBO), producer-scrounger TLBO (PSTLBO). Among these seven benchmark functions, it can be clearly seen that the competition is mainly between TLBO, PSTLBO, and MpBBO. Both win with the 3 best means out of 7. For the best standard deviation (StDev), it is obvious that MpBBO is the winner. TLBO is superior with non-separable functions, such as Ackley's and Griewank's functions, while MpBBO performs better with separable functions, such as Rastrigin's and Schwefel's No. 2.26 functions. The noise term applied to Quartic function makes PSTLBO win with the best mean and MpBBO win with the best standard deviation.

Lastly, the performance of MpBBO is evaluated using ten benchmark functions, and then compared with the following EAs [397]: PSO, gravitational search algorithm (GSA), BFO, and effective BFO (EBFO). The best and mean values achieved by these optimization algorithms are tabulated in Table 2.14. Except for BFO, most of them perform well. PSO wins with the 10 lowest errors out of 10, while GSA, EBFO, and MpBBO score 9 out of 10. However, with the 8 best means out of 10, MpBBO has the highest stability compared to others.

More information, with some statistical hypothesis tests for this algorithm, is given in [31]. Based on all these significant results, the MpBBO algorithm is selected

Table 2.11: Comparison of the Best Results over 100 Trials of MpBBO and Ten Other Types of EAs [327]

Algorithm	**Rastrigin "f29"**			**Rosenbrock "f30"**			**Sphere "f35"**		
Type	**Best**	**Mean**	**Median**	**Best**	**Mean**	**Median**	**Best**	**Mean**	**Median**
PSO-g-a	64	104	101.5	195	497.1	302.5	238	309.4	303
PSO-l-a	102	185.7	176	326	704.3	497	406	449.4	452
H-PSO-a	95	432.7	291	247	528.3	340.5	301	360	361
PSO-g-b	82	142	132	230	641.2	382.5	289	363	355
PSO-l-b	119	270	220.5	408	798	615	516	563.2	563
H-PSO-b	142	500.9	367.5	295	780.2	471	388	453.9	453
$\wedge$ **H-PSO**	65	151.2	127	238	702	369.5	301	351.5	348
$\vee$ **H-PSO**	52	184.4	117	144	352.7	262.5	173	209.6	205
PSO-B-SA1	98	147	148	203	270.2	260	3	9.6	10
PSO-B-SA2	13	38.8	36	**37**	**52.64**	**54**	5	10	9
MpBBO	**0.61**	**1.66**	**1.58**	69.83	304.21	246.46	**1.54**	**3.82**	**3.47**

as the basis of the next highly advanced triple-hybrid optimization algorithm.

2.4.4 MpBBO-SQP

The goal of this hybridization is to have a good balance between the exploration and exploitation levels. In this proposed algorithm, two phases are suggested. Firstly, modifying the migration stage of BBO by hybridizing it with the Metropolis criterion of SA so that the bad migrated islands can be checked whether they are worth being re-selected again as candidate solutions or not. Secondly, the best-obtained solutions are fine-tuned by recycling them through SQP. If these elites are improved, then they will take the positions of the worst individuals in the population so that the original elites of MpBBO can be kept away from any change. The final name of this hybrid algorithm, which is a combination of three different algorithms, is MpBBO-SQP[35]. It has been tried to collect the main strengths of each algorithm into one

[35]Mp stands for the Metropolis criterion of SA.

Table 2.12: Comparison of the Best Results over 30 Trials of MpBBO and Two Other Types of EAs [45]

Function	BA			BFO			MpBBO		
Name	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean
Rastrigin	4.99E-04	2.30E+00	1.21E+00	**1.69E-08**	9.95E-01	5.30E-01	7.50E-06	**8.60E-04**	**2.03E-04**
Egg Crate	1.27E-04	1.30E+00	3.18E-01	**1.98E-09**	**2.12E-06**	**4.41E-07**	1.53E-07	9.29E-05	1.15E-05
Bartels-Conn	2.94E-01	2.59E+02	9.07E+01	**0.00E+00**	1.09E+04	3.27E+03	4.33E-03	**4.34E-01**	**9.51E-02**
Beale	1.18E-06	8.41E-03	3.74E-03	**2.68E-10**	**2.86E-07**	**1.01E-07**	2.65E-07	1.82E-03	4.57E-04
Bohachevsky F1	8.07E-02	1.30E+01	5.15E+00	2.42E-01	3.15E+02	4.18E+01	**1.09E-04**	**4.13E-02**	**8.39E-03**
Bohachevsky F2	2.4E-03	1.74E+01	5.17E+00	1.29E-02	3.14E+02	3.04E+01	**8.42E-05**	**2.34E-01**	**9.56E-02**
Bohachevsky F3	1.11E-02	2.61E+01	7.05E+00	**4.61E-07**	2.12E+02	3.10E+01	1.77E-04	**1.84E-01**	**5.31E-02**
Booth	8.57E-06	2.98E-01	9.43E-02	**1.34E-10**	**3.33E-07**	**8.14E-08**	1.47E-05	1.30E-03	4.27E-04
Parsopoulos	1.00E-05	2.54E-02	6.54E-03	1.72E-10	**4.23E-08**	**1.27E-08**	**1.33E-13**	1.02E-07	1.46E-08
Total Winning	0/9	0/9	0/9	**6/9**	4/9	4/9	3/9	**5/9**	**5/9**

algorithm and, at the same time, the main weaknesses are rejected. However, one of the inherent problems faced with most hybrid algorithms is that their processing times are higher than that of the classical algorithms. However, MpBBO-SQP is relatively fast, because the internal searching loops l of the SA sub-algorithm are excluded and compensated by receiving multiple design points from the BBO sub-algorithm without getting any big effect on the overall performance of MpBBO-SQP. Also, the essential modifications considered in [44] are taken into account to save an additional amount of CPU time.

The flowchart shown in Figure 2.18 gives a quick overview of the proposed MpBBO-SQP optimization algorithm, while the pseudocode given in Algorithm 9 describes the whole mechanism of the proposed triple-hybrid MpBBO-SQP algorithm. As can be seen from the pseudocode, the migration stage of BBO is extended to include the Metropolis criterion of SA. Thus, the migrated n SIV between N ISI will not take place unless they pass this test. The full description of the heuristic part of this algorithm, i.e. MpBBO, is given in the preceding subsection. Once the migration stage is completed and checked by the Metropolis criterion, the elites are fine-tuned by SQP before starting the next generation. The MpBBO-SQP algorithm with this triple-hybrid approach will be used in the next chapter to solve one of the popular electric power system operation problems.

Algorithm 9 Hybrid MpBBO-SQP Algorithm Pseudocode

Require: Initialization stage with all the parameters

1: Find λ_i and μ_i rates, then Pr_i and m_i rates
2: Generate N islands with unduplicated n SIV
3: Sort and map the population to the species count (i.e., $ISI_1 = ISI_{best}$ is coupled with $\mu_{\max}$ or $\lambda_{\min}$, and continue until reaching $ISI_N = ISI_{worst}$)
4: **for** $g \leftarrow 1$ *to* G **do** {where G = number of generations}
5: **if** $g = 1$ **then**
6: Find the initial temperature $\breve{T}_o$ based on the average of the 1^{st} four best solutions using (2.65)
7: **else**
8: Updated the temperature for the gth generation
9: **end if**
10: Save the required best solutions "elites" to be recycled again in the next generation
11: Save the vectors of all the individuals (before migration) in a temporary matrix M_1 with size $N \times n$ and their cost functions in a temporary vector V_1 with length N
12: Do migration (refer to Algorithm 2)
13: Save the vectors of all the individuals (after migration) in a temporary matrix M_2 with size $N \times n$ and their cost functions in a temporary vector V_2 with length N
14: **for** $i \leftarrow 1$ *to* N **do** {where N = number of islands or individuals}
15: Calculate $\Delta\breve{E}_i = V_2(i) - V_1(i)$
16: **if** $\Delta\breve{E}_i > 0$ **then**
17: Apply the Metropolis criterion $\breve{P}(\breve{E}) = e^{\frac{-\Delta\breve{E}_i}{k_B \breve{T}_g}}$
18: **if** $\breve{P}(\breve{E}) >$ rand **then**
19: Accept $M_2(i, 1 \rightarrow n)$ vector of matrix M_2 as an updated individual for ISI_i
20: **else**
21: Re-select the past $M_1(i, 1 \rightarrow n)$ vector of matrix M_1 as a confirmed individual for ISI_i

22: **end if**
23: **end if**
24: **end for**
25: Update the population with sorting and mapping
26: Select the best individuals (*elit*)
27: **for** $j \leftarrow 1$ *to elit* **do** {where *elit* = the best individuals}
28: Tune the jth elite by SQP
29: **if** $ISI_j^{tuned} \leqslant ISI_j^{untuned}$ **then**
30: Insert ISI_j^{tuned} in the population by taking the place of the worst individuals
31: **else**
32: Neglect ISI_j^{tuned}
33: **end if**
34: **end for**
35: Update the population with sorting and mapping
36: Do mutation (refer to Algorithm 3)
37: Clear any duplicated SIV (refer to Algorithm 4)
38: Update the population with sorting and mapping
39: **if** $g > 1$ **then**
40: Replace the worst ISI with the previous good ISI that are saved in the elitism stage
41: Update the population with sorting and mapping
42: **end if**
43: **end for**
44: Display the best individual

Table 2.13: Comparison of the Best Means and Standard Deviation (StDev) over 50 Trials of MpBBO and Six Other Types of EAs [90]

Fun. Name	Criteria	SPSO	LDWPSO	QPSO	TLBO	ETLBO	PSTLBO	MpBBO
Ackley	**Mean**	8.09E-02	5.52E-04	1.07E+00	**3.55E-15**	1.24E-08	**3.55E-15**	1.94E-02
	StDev	± 7.57E-02	± 5.46E-04	± 8.43E-01	± **0.00E+00**	± 6.79E-08	± **0.00E+00**	± 7.22E-03
Griewank	**Mean**	4.80E-01	1.83E-01	1.67E-01	**4.93E-03**	7.40E-03	2.31E-02	5.08E-02
	StDev	± 1.78E-01	± 1.31E-01	± 1.54E-01	± **7.90E-03**	± 1.24E-02	± 2.75E-02	± 2.56E-02
Quartic (with noise)	**Mean**	5.10E-03	4.81E-03	3.61E-03	1.25E-03	1.11E-03	**1.02E-03**	1.10E-03
	StDev	± 2.41E-03	± 2.33E-03	± 2.43E-03	± 5.87E-04	± 5.26E-04	± 5.20E-04	± **4.63E-04**
Rastrigin	**Mean**	7.21E+00	4.51E+00	9.43E+00	3.20E+00	4.21E+00	3.57E+00	**9.72E-04**
	StDev	± 3.18E+00	± 1.36E+00	± 6.57E+00	± 1.45E+00	± 1.59E+00	± 1.89E+00	± **6.01E-04**
Rosenbrock	**Mean**	3.82E+01	8.39E+00	2.38E+01	2.93E+00	**2.54E+00**	3.09E+00	6.01E+00
	StDev	± 4.29E+01	± 1.25E+01	± 5.34E+01	± 7.74E-01	± **3.14E-01**	± 7.09E-01	± 4.49E+00
Schwefel F2.26	**Mean**	-3.67E+03	-3.64E+03	-2.33E+03	-3.41E+03	-3.38E+03	-3.88E+03	**-4.19E+03**
	StDev	± 1.83E+02	± 2.23E+02	± 1.84E+02	± 3.77E+02	± 3.61E+02	± 1.82E+02	± **4.03E-02**
Step	**Mean**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	StDev	± **0.00E+00**	± **0.00E+00**	± **0.00E+00**	± **0.00E+00**	± **0.00E+00**	± **0.00E+00**	± **0.00E+00**
Total Winning	**Mean**	1/7	1/7	1/7	**3/7**	2/7	**3/7**	**3/7**
	StDev	1/7	1/7	1/7	3/7	2/7	2/7	**4/7**

Table 2.14: Comparison of the Best Results over 50 Trials of MpBBO and Four Other Types of EAs [397]

Function	PSO		GSA		BFO		EBFO		MpBBO	
Name	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
Branin	**0.398**	**0.398**	**0.398**	**0.398**	**0.398**	0.409	**0.398**	**0.398**	**0.398**	**0.398**
Foxholes	**0.998**	**0.998**	**0.998**	4.631	**0.998**	**0.998**	**0.998**	3.482	**0.998**	**0.998**
GoldStein-Price	**3.000**	**3.000**	**3.000**	**3.000**	**3.000**	4.721	**3.000**	**3.000**	**3.000**	**3.000**
Hartman No.3	**-3.863**	**-3.863**	**-3.863**	**-3.863**	-3.862	-3.636	**-3.863**	**-3.863**	**-3.863**	**-3.863**
Hartman No.6	**-3.322**	-3.263	**-3.322**	**-3.322**	-3.252	-2.539	**-3.322**	**-3.322**	**-3.322**	-3.264
Kowalik	**3.07E-04**	**4.74E-04**	5.89E-04	4.02E-03	4.15E-04	4.21E-03	5.78E-04	7.67E-04	3.55E-04	7.83E-04
Shekel No.1	**-10.153**	-7.607	**-10.153**	-5.381	-10.152	-8.567	**-10.153**	-7.677	**-10.153**	**-10.153**
Shekel No.2	**-10.403**	-9.693	**-10.403**	**-10.403**	-10.332	-8.736	**-10.403**	-10.276	**-10.403**	**-10.403**
Shekel No.3	**-10.536**	-9.452	**-10.536**	-10.232	-10.526	-8.829	**-10.536**	-10.294	**-10.536**	**-10.536**
Six-Hump Camel-Back	**-1.032**	**-1.032**	**-1.032**	**-1.032**	**-1.032**	-1.005	**-1.032**	**-1.032**	**-1.032**	**-1.032**
Total Winning	**10/10**	6/10	9/10	6/10	4/10	1/10	9/10	5/10	9/10	**8/10**

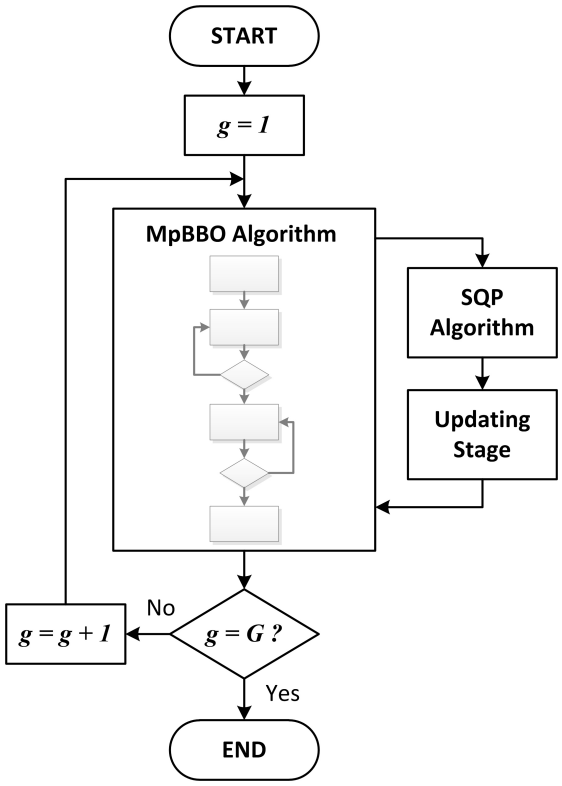


Figure 2.18: Flowchart of the Proposed Hybrid MpBBO-SQP Algorithm

Chapter 3

Economic Load Dispatch Using MpBBO-SQP

3.1 Overview

Optimal Economic operation is considered as one of the most important problems in any power system that has to be solved to satisfy the load required by the end-users at the lowest possible production cost so that the net profit can be maximized. The other goal is to minimize the **emission rates** (oxides of nitrogen "**NOx**", oxides of sulfur "**SOx**", oxides of carbon "**COx**", **soot**, and unburned hydrocarbons "**UHCs**").

Two main strategies can be involved here to achieve economic operation. The first one is based on scheduling the output of generating units to meet the required load demand at the lowest possible fuel consumption. The second strategy is based on minimizing the losses in the network by controlling the flow of power in each branch. The first strategy is called the **economic load dispatch** (**ELD**) problem, while the second strategy is called the **minimum-loss** problem; and both strategies can be optimized by means of the **optimal power-flow** (**OPF**) technique [155].

To solve the ELD problem, there are two possible streams called analytical and numerical techniques. The first one is mainly used if the given system is small and has many simplifications (such as neglecting generator limits and network losses) [326]. The second technique is more advanced and applicable to real-world problems. It is used especially to solve large and highly complex systems. To minimize the objective function of the ELD problem, many traditional and modern optimization algorithms have been suggested [4, 63, 123, 130, 141, 155, 165, 175, 202, 217, 243, 258, 298, 326, 386]. Also, this objective involves many constraints (equality, inequality, and side constraints) that need first to be satisfied to get a feasible solution.

Some of these traditional optimization techniques are: the **lambda-iteration** method, the **base point and participation factors** method, the **gradient** method, **Newton-Raphson** (**NR**) method, and LP/NLP methods. Because these techniques are single-search methods, they are very fast compared to modern population-based

nature-inspired algorithms if both are initialized with the same number of iterations. However, there are many limitations to these conventional techniques, like [141, 165]: converging to non-global solutions, derivative-based approaches[1], the incremental cost curves of generating units are linearized[2]. Practically, solving this power engineering problem requires to consider many obstacles and constraints (such as **power losses**, **ramp rate limits**, **prohibited operating zones**, and **multiple fuel** options), which makes conventional methods insufficient for this type of applications. Although dynamic programming (DP) is successfully used to solve this problem [63, 217], it consumes a large amount of CPU time and it may end up trapping into local optimum solutions.

Based on this, more recent optimization techniques[3][4] are used to overcome the inherent complexities that come with the conventional techniques. Thus, there is no need to simplify the design function, determine derivatives, or even selecting good initial/starting points to ensure searching within the global area of the search space. If penalty functions are selected to deal with these constrained problems, then the optimization algorithms equipped with these constraint-handling techniques can work independently. Thus, the design function and its objective (whether it is minimization or maximization) can be inserted as a plug-in function within the program.

In this chapter, the triple-hybrid optimization algorithm MpBBO-SQP presented in Chapter 2 is used to solve the ELD problem of the 3-, 13- and 40-unit test systems. Although the classical BBO algorithm outperforms many nature-inspired algorithms given in the literature [258], it has been found that our proposed MpBBO-SQP algorithm can win in this competition.

[1] For example, Newton-Raphson method.

[2] For example, linear programming method.

[3] The mechanisms of most algorithms are inspired by nature; or, more specifically, from biology, physics and chemistry science [197]. Please, refer to Chapter 2; particularly Figure 2.8.

[4] For example, genetic algorithm (GA), evolutionary programming (EP), differential evolution (DE), simulated annealing (SA), particle swarm optimization (PSO), ant colony optimization (ACO), artificial bee colony (ABC), and bacterial foraging optimization (BFO) [4, 63, 141, 165, 202, 243, 258, 313, 348].

3.2 Mathematical Formulation of ELD Problems

As extensively discussed in Chapter 2, the first step to optimize any ELD problem is to transform the real-world problem into a mathematical model. The following subsections cover the classical ELD optimization model.

3.2.1 Objective Function

The fuel-cost variation of each generating unit can be represented by many ways based on the type of each unit [298]. Suppose that the ith unit is a conventional type[5]. To achieve an optimal operation of n generating units, the following single-objective function is commonly used:

$$\text{OBJ} = \min \sum_{i=1}^{n} C_i^a \left(P_i\right) \tag{3.1}$$

where C_i^a is the operating cost of the ith unit to supply P_i, which is a function of the active power P_i generated by that unit.

Some studies, especially OPF and **electricity market**, consider both the active and reactive power. Thus, the cost required to generate a specific amount of reactive power can be represented as:

$$\text{OBJ} = \min \sum_{i=1}^{n} C_i^r \left(Q_i\right) \tag{3.2}$$

where C_i^r is the operating cost of the ith unit to supply Q_i, which is a function of the reactive power Q_i generated by that unit.

These two operating costs can be mathematically expressed by the following polynomial-based regression models [171]:

$$C_i^a \left(P_i\right) = \alpha_0 + \sum_{j=1}^{d} \alpha_j P_i^j \tag{3.3}$$

$$C_i^r \left(Q_i\right) = \beta_0 + \sum_{j=1}^{d} \beta_j Q_i^j \tag{3.4}$$

where αs and βs are the polynomial coefficients obtained by fitting C_i^a and C_i^r via linear regression analysis. d is the degree of these two polynomial equations. Thus,

[5] i.e., a thermal unit.

they become linear equations when $d = 1$, quadratic equations when $d = 2$, and cubic equations when $d = 3$ — as follows:

- If $d = 1$:

$$C_i^a(P_i) = \alpha_0 + \alpha_1 P_i \tag{3.5}$$
$$C_i^r(Q_i) = \beta_0 + \beta_1 Q_i \tag{3.6}$$

- If $d = 2$:

$$C_i^a(P_i) = \alpha_0 + \alpha_1 P_i + \alpha_2 P_i^2 \tag{3.7}$$
$$C_i^r(Q_i) = \beta_0 + \beta_1 Q_i + \beta_2 Q_i^2 \tag{3.8}$$

- If $d = 3$:

$$C_i^a(P_i) = \alpha_0 + \alpha_1 P_i + \alpha_2 P_i^2 + \alpha_3 P_i^3 \tag{3.9}$$
$$C_i^r(Q_i) = \beta_0 + \beta_1 Q_i + \beta_2 Q_i^2 + \beta_3 Q_i^3 \tag{3.10}$$

Actually, the best and most popular regression models used in the literature are (3.7) and (3.8), because of two reasons:

1. Equations (3.5)-(3.6) are weak to explain the variability of C_i^a and C_i^r in terms of variance (s), mean squared error (MSE), coefficient of determination (R^2) and its adjusted and predicted versions (R^2_{adj} and R^2_{pre}), etc.

2. Statistically, it is insignificant to increase d from 2 to 3, because the improvements on $\{s, \text{MSE}, R^2, R^2_{\text{adj}}, R^2_{\text{pre}}, \cdots\}$ through using (3.9) and (3.10) are insufficient to replace the preceding quadratic equations in favor of the cubic equations.

Sometimes, (3.5) and (3.6) are modified to be in piecewise forms, so the nonlinear

cost curves C_i^a and C_i^r can be linearized in κ zones as follows[6]:

$$C_i^a = \begin{cases} \alpha_{1,0} + \alpha_{1,1}P_i; & P_i \in \left[P_i^{\min}, P_i^{Z_1^{\min}}\right) \\ \alpha_{2,0} + \alpha_{2,1}P_i; & P_i \in \left[P_i^{Z_1^{\min}}, P_i^{Z_1^{\max}}\right) \\ \alpha_{3,0} + \alpha_{3,1}P_i; & P_i \in \left[P_i^{Z_1^{\max}}, P_i^{Z_2^{\min}}\right) \\ \quad\vdots & \\ \alpha_{\kappa,0} + \alpha_{\kappa,1}P_i; & P_i \in \left[P_i^{Z_\kappa^{\max}}, P_i^{\max}\right] \end{cases} \tag{3.11}$$

$$C_i^r = \begin{cases} \beta_{1,0} + \beta_{1,1}Q_i; & Q_i \in \left[Q_i^{\min}, Q_i^{Z_1^{\min}}\right) \\ \beta_{2,0} + \beta_{2,1}Q_i; & Q_i \in \left[Q_i^{Z_1^{\min}}, Q_i^{Z_1^{\max}}\right) \\ \beta_{3,0} + \beta_{3,1}Q_i; & Q_i \in \left[Q_i^{Z_1^{\max}}, Q_i^{Z_2^{\min}}\right) \\ \quad\vdots & \\ \beta_{\kappa,0} + \beta_{\kappa,1}Q_i; & Q_i \in \left[Q_i^{Z_\kappa^{\max}}, Q_i^{\max}\right] \end{cases} \tag{3.12}$$

where $P_i^{\min}$ and $Q_i^{\min}$ are respectively the lower bounds of the ith active and reactive power. Similarly, $P_i^{\max}$ and $Q_i^{\max}$ are respectively the upper bounds of the ith active and reactive power. $P_i^{Z_\kappa^{\min}}$ and $Q_i^{Z_\kappa^{\min}}$ are respectively the lower bounds of the κth operating zone Z_κ assigned to the ith active and reactive power. Similarly, $P_i^{Z_\kappa^{\max}}$ and $Q_i^{Z_\kappa^{\max}}$ are respectively the upper bounds of the κth operating zone Z_κ assigned to the ith active and reactive power. It has to be said that (3.11) and (3.12) are different than the functional constraints of the prohibited operating zones phenomenon[7].

In this chapter, only the quadratic **fuel-cost function** of the active power is used. Thus, for simplicity, C_i^a is replaced with C_i and (3.7) is expanded and re-expressed as follows:

$$C_i\left(P_i\right) = \alpha_i + \beta_i P_i + \gamma_i P_i^2 \tag{3.13}$$

where α_i, β_i, and γ_i are the constants of the ith unit, and they are defined as follows [175]:

- α_i: cost of crew's salary, interest and depreciation ($/hr)
- β_i: fuel-cost coefficient ($/MWh)

[6]The main goal from this approach is to solve ELD problems by LP.
[7]This phenomenon will be discussed later in this section.

- γ_i: losses measurements in the system ($/MW^{2}h)

Equation (3.13) should be extended if the term of the valve-point loading effect is included as follows [348]:

$$C_i\left(P_i\right) = \alpha_i + \beta_i P_i + \gamma_i P_i^2 + \left|e_i \times \sin\left[f_i \times \left(P_i^{\min} - P_i\right)\right]\right| \tag{3.14}$$

where e_i and f_i are the fuel-cost coefficients of the ith unit modeled with the valve-point loading effect.

Moreover, the fuel-cost function should also be modified if some additional specifications (like emission rates and multiple fuels) are considered[8] [243,298]. Furthermore, for non-conventional generators, the quadratic C_i is replaced with other suitable expressions. For example, a linear fuel-cost function is considered for wind-generated power [166].

Once the objective function is constructed, then some equality, inequality and side constraints should be satisfied in order to have a feasible solution. Detailed information about these constraints can be found in [4, 30, 130, 155, 243, 386]. The next subsections cover the most popular constraints:

3.2.2 Generator Active Power Capacity Constraint

Each generating unit has its own lower and upper active power limits. This side constraint can be expressed as:

$$P_i^{\min} \leqslant P_i \leqslant P_i^{\max} \tag{3.15}$$

where $P_i^{\min}$ and $P_i^{\max}$ are respectively the minimum and maximum active power that can be supplied by the ith generator.

3.2.3 Active Power Balance Constraint

To meet the consumers' power requirement, the total generated power must satisfy the power demand as well as the losses in the power network. This equality constraint can be expressed as:

$$P_T - P_D - P_L = 0 \tag{3.16}$$

[8] This will be extensively explained in Chapter 7.

where P_T is the total power generated by n units, which can be calculated as follows:

$$P_T = \sum_{i=1}^{n} P_i \tag{3.17}$$

and P_D is the total active load demand. The term P_L represents the transmission losses, which can be calculated by using **Kron's loss formula** as follows:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \tag{3.18}$$

where B_{ij}, B_{0i}, and B_{00} are called the **loss coefficients**; and sometimes called the **B-coefficients** [123, 326].

3.2.3.1 Equality to Inequality Constraint Conversion

As described in the last chapter, satisfying equality constraints through a non-gradient iterative process is very hard. Instead, an acceptable amount of tolerance "$\pm\ \varepsilon$" can be used. Thus, (3.16) is satisfied if:

$$-\varepsilon \leqslant h(P) \leqslant \varepsilon \quad , \quad \text{where} \quad h(P) = P_T - P_D - P_L \tag{3.19}$$

Because (3.19) is a functional constraint, so it can be split into two parts of inequality constraints as follows:

$$h(P) - \varepsilon \leqslant 0 \tag{3.20}$$

$$-h(P) - \varepsilon \leqslant 0 \tag{3.21}$$

These new constraints can be satisfied by employing either the penalty functions or the random search method[9].

3.2.3.2 External Optimizer

The other possible approach to iteratively solve (3.16) is by employing a special sub-algorithm to deal with that stiff constraint. For example, the Newton-Raphson technique can be employed as a sub-algorithm within the main optimizer just to satisfy this task.

[9] The full description about these constraint-handling techniques have been covered in the first section of the last chapter; specifically, in Subsection 2.1.8.

3.2.3.3 Slack Generator

The **slack generator** technique can also be used to directly satisfy this equality constraint. Although any generator can be selected as a slack unit, it is preferable to select the unit with the largest capacity to have a better chance to satisfy this constraint in one pass. Neglecting the transmission losses P_L and considering P_1 as a slack generator, the following equation can be used:

$$P_1 = P_D - \sum_{i=2}^{n} P_i \tag{3.22}$$

If the term P_L is taken into account, then (3.22) is converted to:

$$\begin{aligned} P_1 &= P_D + P_L - \sum_{i=2}^{n} P_i \\ &= P_D + \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} - \sum_{i=2}^{n} P_i \end{aligned} \tag{3.23}$$

By separating the slack generator[10] P_1 from the summations and putting all the P_1-terms on the left-hand side, (3.23) becomes:

$$\begin{aligned} P_1 - \sum_{i=1}^{n} P_1 B_{1i} P_i - \sum_{i=2}^{n} P_i B_{i1} P_1 - B_{01} P_1 = P_D + \sum_{i=2}^{n}\sum_{j=2}^{n} P_i B_{ij} P_j \\ + \sum_{i=2}^{n} B_{0i} P_i + B_{00} - \sum_{i=2}^{n} P_i \end{aligned} \tag{3.24}$$

By extracting P_1 from the first summation, (3.24) becomes a quadratic equation as follows:

$$a P_1^2 + b P_1 + c = 0 \tag{3.25}$$

The analytical solution to this 2^{nd} order polynomial equation can be obtained by finding the positive roots of the following general formula[11]:

$$P_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad , \quad \text{and} \quad b^2 - 4ac \geqslant 0 \tag{3.26}$$

[10] P_1 is just as an assumption. Any unit can be used as a slack generator.

[11] It is important to note that the solution of this analytical technique becomes valid only if P_1 passes some restrictions, like: located between the lower and upper limits, positive and real value, satisfies its prohibited operating zones, satisfies its downward and upward ramp rates, etc.

where

$$a = -B_{11} \tag{3.27}$$

$$b = 1 - \sum_{i=2}^{n} B_{1i} P_i - \sum_{i=2}^{n} P_i B_{i1} - B_{01} \tag{3.28}$$

$$c = \sum_{i=2}^{n} P_i - P_D - \sum_{i=2}^{n} \sum_{j=2}^{n} P_i B_{ij} P_j - \sum_{i=2}^{n} B_{0i} P_i - B_{00} \tag{3.29}$$

The slack generator approach is adopted in this study.

3.2.4 Generator Ramp Rate Limits

The power output of the ith unit cannot be adjusted instantaneously. Thus, for a more convenient solution, the ramp rate limit has to be considered where the increasing and decreasing actions should happen within some specific steps as follows:

$$P_i^{\text{now}} - P_i^{\text{new}} \leqslant R_i^{\text{down}} \tag{3.30}$$

$$P_i^{\text{new}} - P_i^{\text{now}} \leqslant R_i^{\text{up}} \tag{3.31}$$

where P_i^{now} and P_i^{new} are the existing and new power outputs of the ith unit, respectively. Also, R_i^{down} and R_i^{up} are respectively the downward and upward ramp rate limits [243]. These two equations, which represent the **ramp rate limits**, can be included within (3.15) as follows [87]:

$$\max\left(P_i^{\min}, P_i^{\text{now}} - R_i^{\text{down}}\right) \leqslant P_i^{\text{new}} \leqslant \min\left(P_i^{\max}, P_i^{\text{now}} + R_i^{\text{up}}\right) \tag{3.32}$$

3.2.5 Prohibited Operating Zones

The phenomenon of the **prohibited operating zones** could happen to the ith unit due to some physical limitations, such as vibrations in the shaft bearing, steam valve opening, etc [243]. This phenomenon causes some discontinuities on the fuel-cost curve. Thus, incorporating this constraint in (3.15) yields:

$$\begin{aligned}
P_i^{\min} &\leqslant P_i \leqslant P_{i,j}^{L} \\
P_{i,j}^{U} &\leqslant P_i \leqslant P_{i,j+1}^{L} \\
&\vdots \\
P_{i,\varkappa_i-1}^{U} &\leqslant P_i \leqslant P_{i,\varkappa_i}^{L} \\
P_{i,\varkappa_i}^{U} &\leqslant P_i \leqslant P_i^{\max}
\end{aligned} \tag{3.33}$$

where $P_{i,j}^{L}$ and $P_{i,j}^{U}$ are respectively the lower and upper limits of the jth prohibited operating zone on the fuel-cost curve of the ith unit, and $\varkappa_i$ stands for the total number of prohibited operating zones associated with the ith unit.

3.2.6 Emission Rates Constraint

Based on rules followed in each jurisdiction, there is an environmental regulation that requires power plants to not exceed the **maximum allowable limits** of the emission rates [302]. This constraint can be expressed as follows [4, 243]:

$$\mathrm{GER}_j\left(P_T\right) \leqslant \mathrm{MAL}_j \tag{3.34}$$

where GER_j and MAL_j stand for the gas emission rate and the maximum allowable limit of the jth gas, respectively.

In addition to the preceding constraints, the objective function could also be subject to many other constraints based on the type and operational philosophy of electric power plants. Some of these constraints are: spinning reserve constraint, line flow constraint, hydro-water discharge limits, reservoir storage limits, water balance equation, network security constraint, etc. More details can be found in [4, 130, 202, 243].

Moreover, optimizing this objective function requires using one or multiple sub-algorithms to satisfy all these constraints to get feasible solutions. For side constraints, the independent or decision variables $\{P_1, P_2, \cdots, P_i, \cdots, P_n\}$ can be easily satisfied by bounding the randomly generated values between the lower and upper limits[12], as shown in (3.15) and programmatically explained in (2.19). The other equality and inequality constraints can be satisfied by using different constraint-handling techniques[13] [122, 316, 345].

3.3 Numerical Experiments

The performance of the conventional and hybrid BBO algorithms are evaluated using three different test systems. The initialization parameters of these optimization

[12]This statement is invalid if the ramp rate limits, prohibited operating zones, and transmission losses P_L are considered.

[13]Please, refer to Chapter 2.

Table 3.1: The Algorithm Parameters Used for Each Test System

Initialization	**3-Unit Test System**		**13-Unit Test System**		**40-Unit Test System**	
Settings[a]	without SQP	with SQP	without SQP	with SQP	without SQP	with SQP
Population Size (N)	20	5	20	15	60	20
Iterations No. (G)	50	20	250	40	1000	500
Total Trials (Tr)	50	50	50	50	50	50
Mutation ($m_{\max}$) [258]	0.007	0.007	0.009	0.009	0.007	0.007
Elitism ($elit$)	1	1	1	4	3	8

[a]The initial population is randomly generated for all the test systems.

algorithms are listed in Table 3.1 for each test system. The fuel-cost curves of the units of all the three test systems are modeled using the quadratic cost function with valve-point loading effects, which is given in (3.14). All the required information about these test systems can be found in [28, 348].

The programs are coded in MATLAB R2011a and simulated on a computing machine having the following specifications: Intel Pentium E5300 2.60 GHz and 4GB RAM with 32-bit Windows XP SP3 operating system.

3.3.1 Test Case I - 3 Generating Units

This system contains three generating units with a load demand of 850 MW. Table 3.2 shows the results obtained by different BBO versions. Based on the mean of MpBBO, the exponential cooling rate is selected for MpBBO-SQP. It can be clearly seen from this table that MpBBO-SQP can converge to the best solution, but it consumes more CPU time than BBO-EM and MpBBO. Also, it can be observed that, due to a few numbers of generators, MpBBO-SQP could trap into local optima. However, this happens just for a few trials as can be concluded from the median. This can be easily avoided by increasing G and/or N. Figure 3.1 shows the fitness curves of this test system, which again proves the superiority of the MpBBO-SQP approach.

Table 3.3 shows an extended comparison with other optimization algorithms presented in the literature. As can be obviously seen from this table, MpBBO-SQP can reach the best known solution with the lowest N and G.

Table 3.2: Comparison Between Different BBO Algorithms (Test Case I)

Generating	**BBO-EM**[a]	**MpBBO**					**MpBBO-SQP**
Unit	PMB	Lin.	Exp.	Inv.	Inv. Lin.	Log.	Exp.
P_1 **(MW)**	300.155	300.084	300.375	299.632	300.331	300.302	300.267
P_2 **(MW)**	149.925	149.982	149.802	150.398	149.780	149.783	149.733
P_3 **(MW)**	399.920	399.934	399.823	399.970	399.889	399.915	400.000
$\sum P_i$ **(MW)**	850.000	850.000	850.000	850.000	850.000	850.000	850.000
Best Cost ($/hr)	8234.22	8234.24	8234.20	8234.46	8234.15	8234.14	**8234.07**
Mean ($/hr)	8252.78	8254.33	**8249.37**	8250.34	8254.91	8254.43	8260.67
Median ($/hr)	8248.36	8252.37	8243.98	8246.37	8251.09	8249.23	**8241.59**
StDev ($/hr)	17.5654	**15.0395**	15.6774	15.8421	16.3213	17.6732	45.2105
Avg. CPU Time (s)	**0.05233**	0.05976	0.06033	0.06143	0.05997	0.05982	0.26762

[a]The acronym BBO-EM stands for BBO with essential modifications. This algorithm has been previously used in [44] and won against the original BBO algorithm. It is used here just to show the superiority of the proposed MpBBO-SQP algorithm.

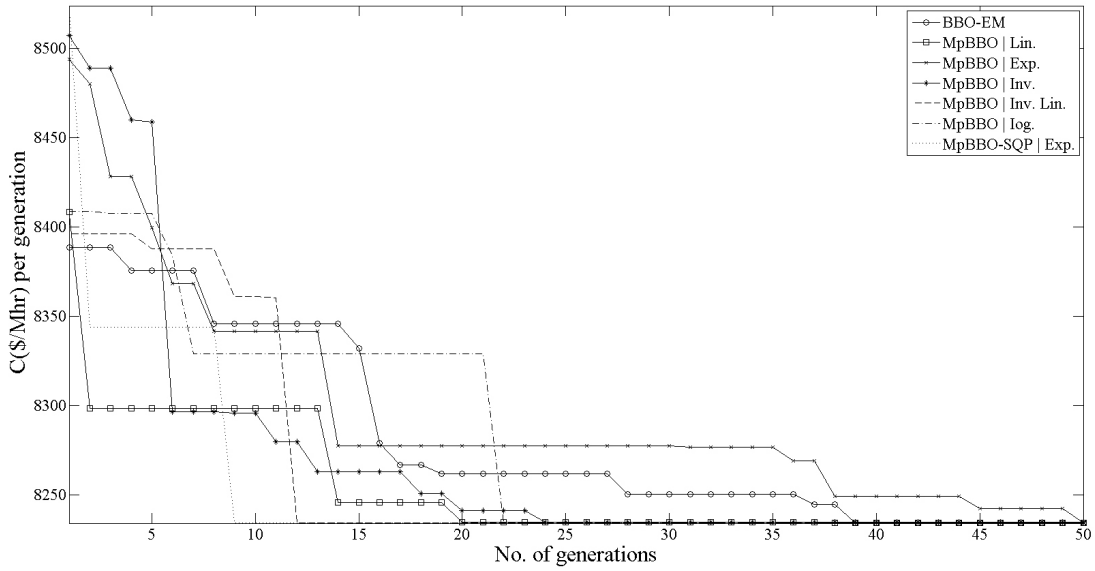


Figure 3.1: The Fitness Curves of Different BBO Versions (Test Case I)

Table 3.3: Comparison Between MpBBO-SQP and Other Algorithms (Test Case I)

Algorithm	**Performance Criteria**		
Type	**Best Cost ($/hr)**	**Population Size (N)**	**Iterations No. (G)**
EP [388]	**8234.07**	30	50
GAB [348]	8234.08	20	50
GAF [348]	**8234.07**	20	50
CEP [348]	**8234.07**	20	50
FEP [348]	**8234.07**	20	50
MFEP [348]	8234.08	20	50
IFEP [348]	**8234.07**	20	50
MPSO [299]	**8234.07**	20	150
GA [258]	8239.20	300	150
PSO [258]	8234.72	300	150
BBO [258]	8234.08	300	150
MpBBO-SQP	**8234.07**	**5**	**20**

3.3.2 Test Case II - 13 Generating Units

This is the second system, which is relatively harder than the preceding one. The load demand that has to be satisfied is 1800 MW. Table 3.4 shows the results obtained by BBO-EM, MpBBO and MpBBO-SQP after 50 trials. Again, MpBBO-SQP is executed with the exponential cooling rate because it shows better mean as compared with the other four cooling schedules. With fair CPU time performance comparison, it has been found that MpBBO-SQP consumes more CPU time. However, it significantly outperforms the other two algorithms; as can be seen in Figure 3.2. This superiority can also be observed in Table 3.5. Among 17 different competitive optimization techniques, MpBBO-SQP can reach 17963.8 $/hr with the lowest G and the second lowest N.

3.3.3 Test Case III - 40 Generating Units

With 40 generating units, this system is considered as one of the biggest ELD test systems available in the literature. The load demand of this ELD problem is 10500 MW. Table 3.6 shows the results obtained by BBO-EM, MpBBO, and MpBBO-SQP

Table 3.4: Comparison Between Different BBO Algorithms (Test Case II)

Generating	**BBO-EM**[a]	**MpBBO**					**MpBBO-SQP**
Unit	PMB	Lin.	Exp.	Inv.	Inv. Lin.	Log.	Exp.
P_1 **(MW)**	449.268	449.099	539.574	538.214	538.804	449.194	628.319
P_2 **(MW)**	305.198	225.160	150.093	299.622	79.921	146.605	149.599
P_3 **(MW)**	145.659	221.228	228.265	70.523	148.985	149.103	222.751
P_4 **(MW)**	109.085	114.141	158.729	111.007	161.286	159.364	109.865
P_5 **(MW)**	111.583	109.271	60.143	62.090	107.561	161.413	109.866
P_6 **(MW)**	109.165	109.234	110.723	60.057	159.572	163.772	109.867
P_7 **(MW)**	66.301	158.037	110.242	110.634	159.795	108.754	109.867
P_8 **(MW)**	109.126	114.962	61.883	160.157	61.197	158.759	60.000
P_9 **(MW)**	159.715	62.576	109.950	155.804	109.449	60.633	109.867
P_{10} **(MW)**	78.143	80.973	42.789	40.405	82.760	42.496	40.000
P_{11} **(MW)**	44.689	42.393	79.005	44.654	40.547	83.474	40.000
P_{12} **(MW)**	56.509	55.606	56.841	91.481	58.033	55.353	55.000
P_{13} **(MW)**	55.557	57.322	91.762	55.352	92.089	61.081	55.000
$\sum P_i$ **(MW)**	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00
Best Cost ($/hr)	18227.5	18226.9	18153.9	18185.0	18226.6	18267.0	**17963.8**
Mean ($/hr)	18393.1	18386.0	18301.6	18308.8	18325.4	18397.7	**18070.0**
Median ($/hr)	18380.4	18389.8	18290.9	18304.3	18321.7	18402.8	**18073.3**
StDev ($/hr)	81.0980	61.6227	74.5877	55.3060	61.9199	58.3879	**40.7717**
Avg. CPU Time (s)	**0.57700**	0.62722	0.82779	0.83726	0.73686	0.63166	4.47230

[a]See the footnote of Table 3.2

after 50 trials. In this case, the inverse linear cooling strategy performs better than the others in terms of average quality. Thus, MpBBO-SQP is executed with this cooling rate. It can be observed from Table 3.6 that the processing time of MpBBO increases as the solution quality increases. This phenomenon is obvious with the exponential, inverse, and inverse linear cooling rates[14]. In this comparison, it can be clearly seen that MpBBO-SQP is much better than the other versions of BBO. However, it consumes around 80% to 148% additional CPU time. In this system, the SQP phase is not activated until it reaches 90% of G; as can be seen in Figure 3.3. This approach gives enough chance to MpBBO phase to explore the search space, and it can also save a significant amount of CPU time. Table 3.7 shows an extended comparison between MpBBO-SQP and 17 different optimization algorithms presented

[14]Please, refer to MpBBO in Chapter 2.

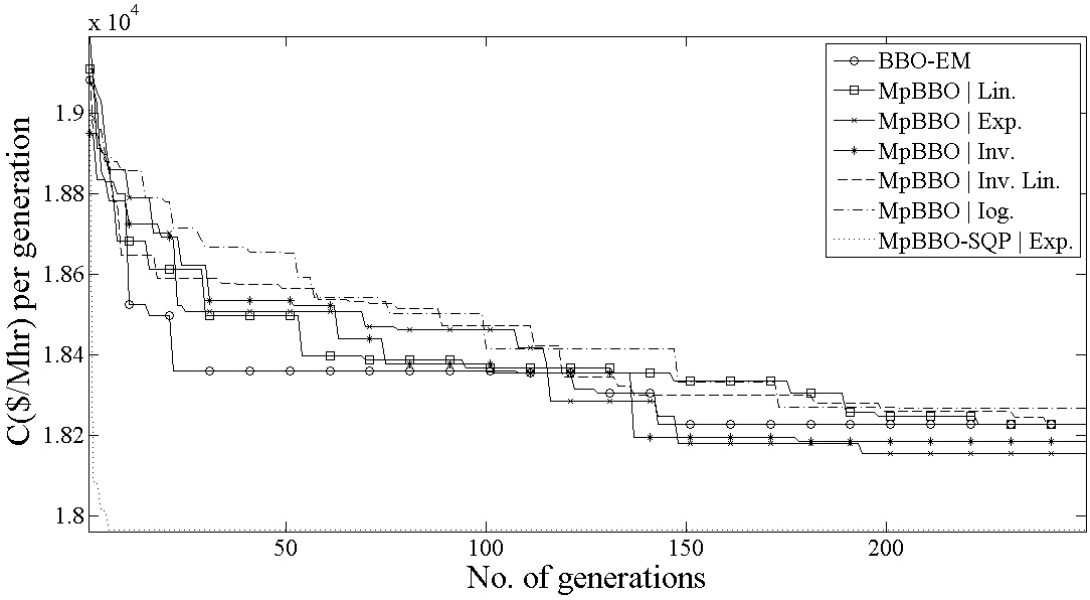


Figure 3.2: The Fitness Curves of Different BBO Versions (Test Case II)

Table 3.5: Comparison Between MpBBO-SQP and Other Algorithms (Test Case II)

Algorithm	**Performance Criteria**		
Type	**Best Cost ($/hr)**	**Population Size (N)**	**Iterations No. (G)**
CEP [348]	18048.2	30	800
FEP [348]	18018.0	30	800
MFEP [348]	18028.1	30	800
IFEP [348]	17994.1	30	800
PSO-SQP [376]	17969.9	100	100
CGA_MU [94]	17975.3	30	30 × 3000
IGA_MU [94]	17964.0	**5**	30 × 3000
PSO [89]	18014.2	20	250
PPSO [89]	17971.0	20	250
MPPSO [89]	17976.2	20	250
APPSO [89]	17978.9	20	250
DPSO [89]	17976.3	20	250
GA–PS–SQP [47]	17964.25	100	1000
ABC [165]	17963.9	300	200
FAPSO-NM [277]	**17963.8**	26	300
FAPSO-VDE [278]	**17963.8**	26	100
BBO [258]	17965.8	50	100
MpBBO-SQP	**17963.8**	15	**40**

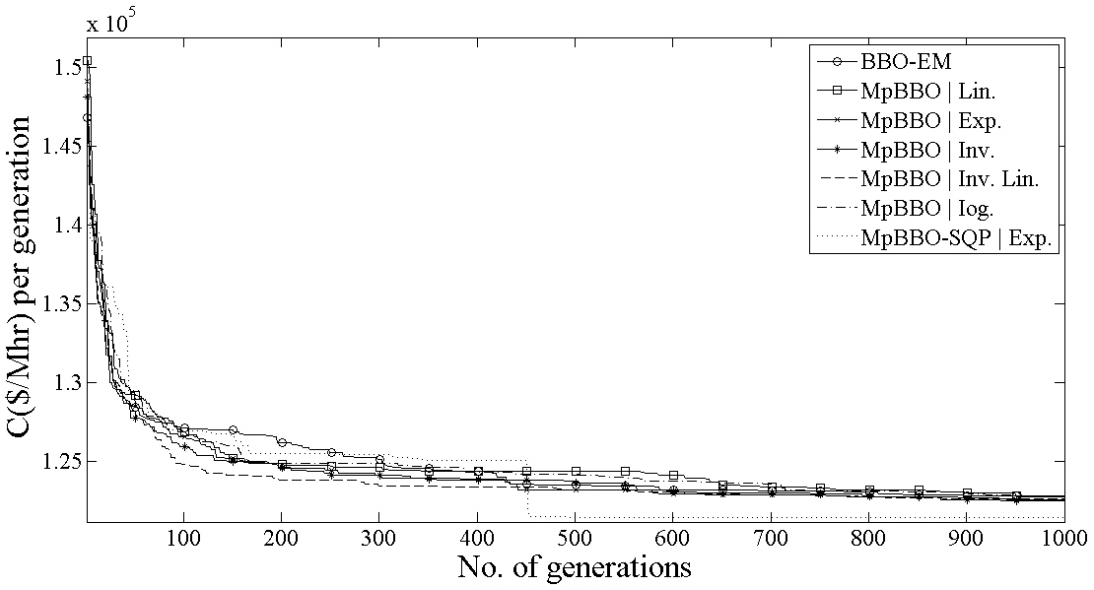


Figure 3.3: The Fitness Curves of Different BBO Versions (Test Case III)

in the literature. Again, MpBBO-SQP can detect better solutions with very small population size N and relatively few numbers of generations G, which proves itself as a superior optimization algorithm.

3.4 Further Discussion

Based on the results presented in the preceding section, it can be concluded that the BBO solution quality can be steeply enhanced by reinforcing the migration stage through the Metropolis criterion of SA and fine-tuning the elites through SQP. With this triple-hybrid approach, both the exploration and exploitation levels can be prettily balanced.

It has been found that the other properties of SA have also been transferred to MpBBO-SQP. Thus, the initial temperature $\breve{T}_o$ and the cooling strategy are very important factors that might affect the overall performance of MpBBO-SQP. However, because BBO does not need many settings in the initialization stage, so MpBBO-SQP has less dependability on the SA parameters.

Also, it has been found that, although the internal searching loops l of SA are deactivated in MpBBO-SQP, the overall hybrid algorithm consumes huge CPU time. This is due to the processing speed of the second hybrid phase. It can be effectively

Table 3.6: Comparison Between Different BBO Algorithms (Test Case III)

Generating	**BBO-EM**[a]	**MpBBO**					**MpBBO-SQP**
Unit	PMB	Lin.	Exp.	Inv.	Inv. Lin.	Log.	Inv. Lin.
P_1 **(MW)**	109.9829	107.1563	110.4708	113.5721	112.3417	110.6294	110.8006
P_2 **(MW)**	112.9480	105.3991	109.8269	110.6250	110.2898	111.5661	110.8009
P_3 **(MW)**	60.8319	101.0059	92.6172	118.7452	96.7447	98.6038	97.4002
P_4 **(MW)**	177.5861	179.2414	177.6211	183.7266	180.1699	180.5318	179.7331
P_5 **(MW)**	89.5334	92.0468	88.5719	89.1711	91.1842	96.0074	90.7899
P_6 **(MW)**	111.2320	109.1939	131.7801	138.4395	106.1829	135.3823	140.0000
P_7 **(MW)**	259.8986	256.5497	292.8111	275.7522	257.3219	280.3666	259.6003
P_8 **(MW)**	283.0137	283.1697	284.6723	291.4189	285.2167	285.0032	284.6004
P_9 **(MW)**	283.6620	285.6290	284.3732	295.5165	283.8322	297.1240	284.5998
P_{10} **(MW)**	131.1614	204.8213	138.3372	134.3401	131.9480	193.0429	130.0000
P_{11} **(MW)**	98.4919	156.3835	176.2985	169.1616	241.4022	94.8783	168.7998
P_{12} **(MW)**	168.5389	172.5566	162.0961	100.6024	154.4301	172.7777	168.7998
P_{13} **(MW)**	214.1733	304.4306	215.7737	215.0265	214.3881	128.7132	214.7599
P_{14} **(MW)**	397.7859	304.6572	298.7209	390.9713	395.1666	395.9701	394.2794
P_{15} **(MW)**	393.1412	299.3851	394.1351	302.5692	390.4244	305.1189	394.2793
P_{16} **(MW)**	387.4290	394.0908	394.4072	392.6898	306.8219	393.6977	304.5197
P_{17} **(MW)**	485.7239	490.0678	490.9882	489.0061	496.8710	489.1214	489.2795
P_{18} **(MW)**	492.6118	493.0292	487.7777	486.1721	489.1898	489.4086	489.2795
P_{19} **(MW)**	512.9207	512.3289	507.5660	511.9301	516.8716	511.3859	511.2795
P_{20} **(MW)**	511.6986	511.2362	511.1204	516.4457	511.3494	511.8786	511.2794
P_{21} **(MW)**	521.4651	523.2986	524.7963	535.3893	520.2215	522.9748	523.2794
P_{22} **(MW)**	524.7502	533.0381	533.9713	533.6980	519.2020	527.9580	523.2794
P_{23} **(MW)**	534.8721	525.2546	521.4333	519.7955	519.9372	516.3175	523.2794
P_{24} **(MW)**	522.3196	538.6236	521.6195	521.9118	525.9079	527.6274	523.2801
P_{25} **(MW)**	521.0463	524.3549	534.4174	528.5863	523.6342	528.7126	523.2800
P_{26} **(MW)**	522.2667	525.3645	529.7122	533.1516	529.7113	528.4108	523.2794
P_{27} **(MW)**	12.9535	11.5957	10.7071	18.0412	12.1148	16.0110	10.0000
P_{28} **(MW)**	16.2587	10.9665	15.4744	10.9164	11.1498	16.1873	10.0000
P_{29} **(MW)**	20.5544	10.1174	10.9433	10.0349	17.1487	14.6821	10.0000
P_{30} **(MW)**	96.5324	89.8871	90.8936	88.2024	91.6684	84.3462	89.7581
P_{31} **(MW)**	189.2199	181.5048	185.5420	185.7671	188.5363	189.4289	190.0000
P_{32} **(MW)**	188.3592	164.5573	189.2287	186.9412	189.1755	184.6063	190.0000
P_{33} **(MW)**	183.6452	169.0938	178.1893	188.7571	179.1483	188.4070	190.0000
P_{34} **(MW)**	171.3132	165.2316	164.1037	166.3106	163.2768	177.2380	164.8006
P_{35} **(MW)**	197.7943	182.2683	165.9807	170.2927	169.1983	171.9220	164.8002
P_{36} **(MW)**	193.9636	164.3957	166.6935	173.5405	170.3869	176.3794	164.8030
P_{37} **(MW)**	107.8912	103.0432	109.0115	109.1962	89.6925	91.7985	110.0000
P_{38} **(MW)**	99.5476	89.1103	98.4373	91.4707	99.2854	106.0449	110.0000
P_{39} **(MW)**	84.6454	99.4453	88.9657	88.6757	87.8194	105.4537	110.0000
P_{40} **(MW)**	508.2361	526.4697	509.9135	513.4386	520.6376	544.2860	511.2794
$\sum P_i$ **(MW)**	10500.00	10500.00	10500.00	10500.00	10500.00	10500.00	10500.00
Best Cost ($/hr)	122762.2	122801.5	122577.9	122509.0	122642.4	122780.1	**121415.3**
Mean ($/hr)	123337.0	123324.5	123080.7	123100.6	123057.2	123269.2	**122025.2**
Median ($/hr)	123326.6	123324.9	123043.9	123023.2	123050.3	123251.5	**121989.1**
StDev ($/hr)	268.1310	262.9107	243.7758	309.6664	**210.4609**	254.9956	343.0694
Avg. CPU Time (s)	**12.53991**	13.68938	17.23036	16.83868	16.71288	13.52386	31.07582

[a] See the footnote of Table 3.2

Table 3.7: Comparison Between MpBBO-SQP and Other Algorithms (Test Case III)

Algorithm	**Performance Criteria**		
Type	**Best Cost ($/hr)**	**Population Size (N)**	**Iterations No. (G)**
CEP [348]	123488.3	100	1000
FEP [348]	122679.7	100	1000
MFEP [348]	122647.6	100	1000
IFEP [348]	122624.4	100	1000
PSO [89]	122324.0	40	500
PPSO [89]	121788.2	40	500
MPPSO [89]	122225.7	40	500
APPSO [89]	122044.6	40	500
DPSO [89]	122160.0	40	500
PSO-LRS [337]	122035.8	**20**	1000
NPSO [337]	121704.7	**20**	1000
NPSO-LRS [337]	121664.4	**20**	1000
CSO [336]	121461.7	30	1000
CDEMD [115]	121423.4	25	1000
ABC [165]	121441.0	800	200
FAPSO-NM [277]	121418.3	60	1000
BBO [258]	121510.8	500	**100**
MpBBO-SQP	**121415.3**	**20**	500

decreased by reducing the number of elites that need to be fine-tuned through SQP and/or by setting a lower accuracy ε for the termination or early stopping criterion[15]. For example, the processing time of the 40-unit test system can be steeply decreased from 31.08 seconds down to 8.13 seconds, with an optimal cost of 121478.0 $/hr, if only the fitness is recycled through SQP instead of all the elites. This means that MpBBO-SQP can get very good solutions with the lowest CPU time[16]. Additionally, it is good to initiate SQP after completing a big portion of the total iterations[17] so that the algorithm can have more chance to explore the search space and at the same time the CPU usage can be effectively reduced.

It has to be said that many options can also be tried here to increase the performance of MpBBO-SQP in terms of solution quality and processing speed. One

[15]In this study, a tolerance of 10^{-7} is used for SQP.
[16]Comparing with BBO-EM and MpBBO.
[17]For example, 90% of G, which is considered during solving the 40-unit test system.

of these options is by finding the optimal algorithm settings[18] by conducting some sensitivity analysis. Moreover, it is good to test MpBBO-SQP with some adaptive cooling strategies. Furthermore, the basic mutation and migration rates of BBO can be replaced with more advanced rates that are presented in the literature. Also, oppositional and blended BBO options (OBBO and BBBO) can be considered as new phases for more advanced hybrid optimization algorithms.

[18] They are called **hyperparameters**, which are covered in Chapters 5, 9, and 10.

Chapter 4

Optimal Coordination of Directional Overcurrent Relays Using BBO-LP

4.1 Overview

Overcurrent protection, as a subject, is very wide. Based on many criteria (applications, input signals, severity levels, etc), the protection can be achieved by many types of protective devices, such as mechanical/thermal devices (Buchholz, bimetallic, and pressure relief relays), fuses, and **overcurrent relays** (**OCRs**).

Compared with other expensive relays, OCRs can compromise between different design criteria (cost, speed, reliability "security vs. dependability", simplicity, adequateness, selectivity, sensitivity, etc), and this is the reason why they are widely used in power system protection [126, 368].

Based on their **time-current characteristic curves** (**TCCCs**), OCRs can be classified into three categories: **definite-current overcurrent relay**[1] (**DCOCR**), **definite-time overcurrent relay** (**DTOCR**), and **inverse-time overcurrent relay** (**ITOCR**). The last type is designed to remedy the inherent weaknesses of the first and second types, and thus it becomes the most popular type of OCRs [147, 292]. To set ITOCRs, there are two independent variables called the **plug setting**[2] (PS) and the **time multiplier setting**[3] (TMS) with three parameters $\{\alpha, \beta, \gamma\}$. These parameters are adjusted to have "short-time inverse", "inverse", "standard inverse", "moderately inverse", "very inverse", "extremely inverse", etc, of TCCCs [56, 147].

In general, OCRs are non-directional devices, which can detect faults based only on their stepped-down current magnitudes $|I|$ measured by **current transformers** (**CTs**) to initiate trip signals to their corresponding **circuit breakers** (**CBs**). To understand the main problem of non-directional OCRs, consider the parallel line

[1] In some references, it is also called **instantaneous overcurrent relay** (**IOCR**) [147, 292].

[2] In some references, it is also known as the **pick-up setting** (PS) and the **tap setting** (TS) [24].

[3] In some references, it is also known as the **time dial setting** (TDS) and the **time lever setting** (TLS) [24].

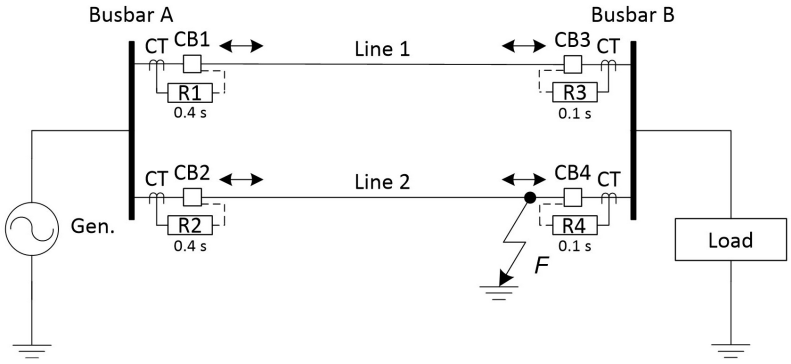


Figure 4.1: Single-End Fed Power System of Parallel Feeders Containing only OCRs

radial circuit shown in Figure 4.1. In this example, assume that there is a fault F on line 2 close to busbar B. If only OCRs are used, then the overcurrent relays R_3 and R_4 will detect the same fault current magnitude and send their trip signals simultaneously to the circuit breakers CB_3 and CB_4, respectively. In addition, CB_2 will be tripped by R_2 after an additional time delay so that the fault can be completely cleared from the system. The problem with this protection scheme is that the load and line 1 are unnecessarily disconnected. Thus, this design is considered unreliable and unselective.

To overcome this problem, an additional unit is combined with each OCR to identify the direction of fault currents. Based on this, both the current magnitude and direction are taken into account for tripping the faulty element as fast and selective as possible, while the remaining parts of the network can operate normally (i.e., the dependability increases). This special protective device is called a **directional overcurrent relay** (**DOCR**), which is the focus of this chapter.

The term **relays coordination** can be shortly defined as "***the quality of selectivity among protective devices*** [52]". Without this crucial stage, the protection design will not satisfy the selectivity criteria, and hence the overall design will become unreliable where there is always a possibility that some backup relays may act before their primary relays.

To achieve that, all protective relays should be adjusted with correct settings to ensure that the primary relays have enough time to act before giving the corresponding backup relays the permission to act. This time delay is known as the **coordination**

time interval[4] (CTI). It can be calculated as follows [21]:

$$CTI = T_{CB} + T_{OS} + T_{SM} \tag{4.1}$$

where T_{CB} is the time delay of the circuit breaker. T_{OS} is called the **overshoot time**[5]. T_{SM} is the safety margin given to the relay to account all the negligible errors (current magnitude measurement error, relay timing error, CT-ratio error, etc [40]). The typical value of CTI lies between 0.2 and 0.5 second [52, 74].

Optimal relay coordination (**ORC**[6]) is simply achieved by satisfying two important things, which are:

1. The primary/backup (P/B) relay pairs are correctly coordinated.
2. The sum of operating times of protective relays, when they act as primary devices, is minimized.

The solution of ORC problems is straightforward for simple radial networks. Moreover, it could be analytically solved by expert protection engineers for some small and uncomplicated networks if all the faults, system contingencies, and abnormalities are analyzed and predetermined. Otherwise, the only available tool is to use optimization techniques where both the solution quality and processing speed are the key factors to determine which algorithm is the best [24, 40, 280]. In general, these problems are considered as highly constrained non-convex **mixed-integer non-linear programming** (**MINLP**) problems where TMS are continuous and PS are discrete[7,8].

The aim of this chapter is to cover: 1. the mathematical formulation of classical ORC problems when only DOCRs are used, 2. the solutions of some popular test systems, 3. the existing approach to enhance the solution quality and the main challenges associated with this approach, and 4. our proposed approach to effectively achieve the last point in an innovative way.

[4] In some references, it is also known as the **selective time interval** (STI) [292] and the **discrimination margin** (DM) [147].

[5] The overshoot time is limited to electromechanical relays. It is also called the **over-travel delay** and the **coasting time** [24].

[6] Please, note that the acronym ORC is different than the acronym OCR. The latter one is used as a short name for overcurrent relays.

[7] In the literature, if numerical relays are used in ORC problems, then both PS and TMS are provided as almost continuous variables [24, 110, 111].

[8] This assumption is not correct in real-world ORC problems [21]; as will be seen later in Chapter 8.

4.2 Mathematical Formulation of ORC Problems

As per extensively discussed in Chapter 2, the design function of any optimization problem should be modeled as an objective function subjected to some design constraints. The following subsections cover all these parts of classical ORC problems:

4.2.1 Objective Function

Suppose that an electrical network contains ß branches, and each branch is protected by two CBs mounted on both ends. If each CB is triggered by one DOCR, then the preceding network should contain $\varrho = 2ß$ DOCRs. Also, it is known that the operating time of each DOCR depends on PS and TMS. Thus, the dimension n of any DOCR-based ORC problem can be calculated as follows:

$$n = 2\varrho = 4ß \tag{4.2}$$

If an **in-zone fault** occurs at the location k, then the operating time of the ith primary relay R_i can be mathematically represented as follows:

$$T_{i,k} = f\left(TMS_i, PS_i, I_{i,k}\right) \quad , \quad i = 1, 2, \cdots, \varrho \tag{4.3}$$

where $I_{i,k}$ is a short-circuit current seen by the ith primary relay R_i for a fault occurring at the kth location.

For $T_{i,k}$, there were many attempts to model ITOCRs mathematically [1, 56, 84, 177, 179, 232, 301, 308]. One of them was modeled based on a polynomial equation, and it was presented by Sachdev et al. in [232]. Until the last decade, many papers have been presented based on that model to solve different ORC problems. In fact, settling on one standard model was a very challenging task. It was deeply discussed by one of the **IEEE-PES** committees in 1989 [1]. Nowadays, the operating time of ITOCRs is mathematically modeled based on two common standards. The first one is known as the **IEC-60255** or **BS142** standard, which is expressed as follows [42, 177]:

$$T_{i,k} = TMS_i \times \frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} \tag{4.4}$$

The ANSI/IEEE standard model can also be used. It is similar to (4.4), but with

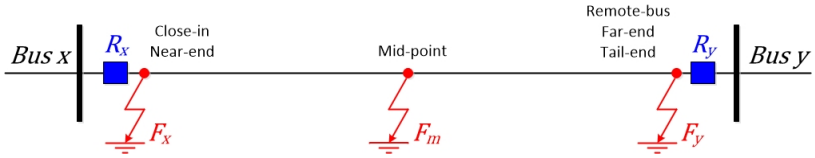


Figure 4.2: Levels of Protection Design Criteria

one extra coefficient called[9] [56, 179] γ:

$$T_{i,k} = TMS_i \times \left[\frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} + \gamma \right] \tag{4.5}$$

It has to be said that the lower and upper bounds of TMS_i depend on the standard TCCC the relay belongs to[10].

The objective of this optimization problem is to find the best values of PS and TMS of all ϱ DOCRs so that the weighted sum of the operating times, when DOCRs act as primary relays, is minimized for l fault locations as follows:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} \sum_{k=1}^{l} w_{i,k} T_{i,k}^{\text{pr}} \tag{4.6}$$

where $w_{i,k}$ represents the fault probability that might happen at the kth location of a branch protected by the ith relay. For the sake of simplicity, all the weights given in (4.6) are considered equal to one [367].

To do short-circuit analysis, two options are available. The first option is to do it manually by hand, which is a weary process and highly unrecommended. The other option is to use one of the known commercial and free-distributed software[11].

For l fault locations on each zone, Damborg et al. in [103] classifies three levels of coordination criteria. By referring to Figure 4.2, the first one is called the **desired design criterion**, which considers two fault locations. The first one is called the **near-end** fault[12] and the other one is called the **far-end** fault[13]. For the near-end

[9] Please, note that there is no subscript i for the coefficients $\{\alpha, \beta, \gamma\}$ because most of the studies consider that all ϱ DOCRs have the same TCCC. If multiple TCCCs are used, then multi-standard coefficients must be applied, and thus the subscript i must be included to these coefficients.

[10] Please, refer to Subsection 4.3.3 for more details about this point.

[11] Such commercial software are: **ETAP**, **DigSILENT PowerFactory**, **PowerWorld Simulator**, **Neplan**, **ASPEN**, **SIEMENS PSS/E**, and **CYME**. Also, there is one popular library in Python called **PandaPower**, which has the ability to do static short-circuit calculation according to IEC 60909.

[12] It is also known as the **close-in** fault.

[13] It is also known as the **remote-bus** fault and the **tail-end** fault.

Table 4.1: Number of Faults Considered for Coordination Criteria

Coordination Criteria	**Number & Names of Fault Locations**
Desired Design Criteria	2 Bolted-Points: F_x and F_y
Minimum Design Criteria	1-Bolted Point: F_x for R_x or F_y for R_y
Enhanced Design Criteria	1-Bolted Point: F_m

location, the fault occurs at the nearest possible point of the line where the primary protective relay is installed, and vice versa for the far-end location [34]. This criterion could be relaxed to what is called the **minimum design criterion** where the relay settings are optimized based on the near-end 3ϕ faults. When larger classes of faults are studied, then one fault at the **mid-point** of each line could be considered. This case is called the **enhanced design criterion** [73], which is applied in [70]. All these three design criteria are covered in Table 4.1 [34]. For the desired criterion, (4.6) is re-expressed as follows:

$$\text{OBJ} = \min \left[\sum_{p=1}^{\varrho^{\text{near}}} T_p^{\text{pr,near}} + \sum_{q=1}^{\varrho^{\text{far}}} T_q^{\text{pr,far}} \right] \tag{4.7}$$

If the same number of relays are considered for both fault locations, then $\varrho^{\text{near}} = \varrho^{\text{far}} = \varrho$. Thus, (4.7) can be simplified to:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} \left(T_i^{\text{pr,near}} + T_i^{\text{pr,far}} \right) \tag{4.8}$$

For other test systems, where the minimum and enhanced criteria are considered, the previous expression is further reduced to:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} T_i^{\text{pr}} \tag{4.9}$$

where T_i^{pr} is calculated either at the near-end or mid-point[14].

Optimizing any one of these objective functions requires to satisfy first the following design constraints:

[14]Please, note that the near-end fault for the xth primary DOCR mentioned in Figure 4.2 is considered as the far-end fault for the yth primary DOCR.

4.2.2 Inequality Constraints on Relay Operating Times

To realize the operation of the ith relay, its speed should be bounded between two limits:

$$T_{i,k}^{\min} \leqslant T_{i,k} \leqslant T_{i,k}^{\max} \tag{4.10}$$

where $T_{i,k}^{\min}$ and $T_{i,k}^{\max}$ are respectively the minimum and maximum operating times of the ith relay R_i for a fault occurring at the kth location. $T_{i,k}^{\min}$ depends on the internal components of R_i, whereas $T_{i,k}^{\max}$ depends on the critical clearing time t_{cr} required to preserve system stability [40, 213, 291].

The above functional constraint can be divided into two inequality constraints as follows:

$$T_{i,k}^{\min} - T_{i,k} \leqslant 0 \tag{4.11}$$

$$T_{i,k} - T_{i,k}^{\max} \leqslant 0 \tag{4.12}$$

4.2.3 Side Constraints on Relay Time Multiplier Settings

Manufacturers of protective relays offer their products with some specifications. One of these specifications is about the lower and upper limits of TMS, which can be expressed as follows:

$$TMS_i^{\min} \leqslant TMS_i \leqslant TMS_i^{\max} \tag{4.13}$$

where $TMS_i^{\min}$ and $TMS_i^{\max}$ are respectively the minimum and maximum values of TMS of the ith DOCR.

4.2.4 Side Constraints on Relay Plug Settings

For the ith relay, the lower limit $PS_i^{\min}$ should be set equal to or greater than the maximum overload current $I_{OL}^{\max}$, and the upper limit $PS_i^{\max}$ should be set equal to or less than the minimum fault current[15]. The term $I_{OL}^{\max}$ can be calculated as follows:

$$I_{OL}^{\max} = \text{OLF} \times I_L^{\max} \tag{4.14}$$

where $I_L^{\max}$ is the maximum rated current. **OLF** is the **overload factor**, which depends on the element being protected, and it is usually set in the range of 1.25-1.5 [147].

[15] In most cases, the minimum fault current is the single-phase 1ϕ short-circuit current [123].

Therefore, the practical side constraints on the plug settings of all ϱ DOCRs depend on the specification of **powerlines**[16] and the short-circuit analysis. Once these two fundamental steps are successfully done, the limits of PS of all ϱ relays can be defined. In general, the side constraint of the ith plug setting PS_i can be mathematically expressed as follows:

$$PS_i^{\min} \leqslant PS_i \leqslant PS_i^{\max} \tag{4.15}$$

To simplify this constraint, most of the studies presented in the literature consider this side constraint as a fixed vector of discrete values, such as [40, 51, 280]. However, some other studies use the practical range associated with each relay, such as [40, 70]. For the sake of clarity, these two bounds can be practically calculated as follows:

$$PS_i^{\min} = \frac{\text{OLF} \times I_{L,i}^{\max}}{CTR_i} \tag{4.16}$$

$$PS_i^{\max} = \frac{2}{3CTR_i} I_{f,i}^{\min} \tag{4.17}$$

where CTR_i is the **CT-ratio** of the ith relay, and $I_{f,i}^{\min}$ is the minimum fault current that must be detected by that relay [147].

4.2.5 Selectivity Constraint Among Primary and Backup Relay Pairs

This inequality constraint ensures that the associated backup DOCRs will not operate before their primary DOCRs. This can be accomplished by selecting the proper PS and TMS so that the backup relay(s) can initiate the trip signal to isolate the kth **out-zone fault** if the ith primary relay exceeds the given chance[17]. The mathematical formulation of this constraint can be expressed as follows[18]:

$$T_{j,k} - T_{i,k} \geqslant CTI \tag{4.18}$$

where $T_{j,k}$ is the operating time of the jth backup relay for an out-zone fault occurred at the kth location.

[16] i.e., **transmission**, **sub-transmission**, and **distribution** lines.

[17] In other words, it is the sum of the ith relay operating time T_i^{pr} plus its coordination time interval CTI_i.

[18] If all ϱ DOCRs have the same coordination time interval, then the subscript i is dropped to have just CTI.

Based on this, the operating time of the jth backup relay must be known to check whether the preceding constraint is satisfied or not. This can be easily calculated by using any one of the previous standard equations. If the ANSI/IEEE standard given in (4.5) is used as the TCCC model, then $T_{j,k}$ can be determined as follows:

$$T_{j,k} = TMS_j \times \left[\frac{\beta}{\left(\frac{I_{j,k}}{PS_j}\right)^{\alpha} - 1} + \gamma \right] \tag{4.19}$$

where $I_{j,k}$ is a short-circuit current seen by the jth backup relay R_j for a fault occurring at the kth location[19].

For having a more realized model, some researchers take into account the case when one of the two-end primary relays operate before the other. At that short-period of time, the network will have a transient topology and thus the fault current will change. Therefore, it is important to ensure that the corresponding backup relays will not operate at that moment [40, 51, 367]. This can be achieved by defining the following inequality constraint:

$$T'_{j,k} - T'_{i,k} \geqslant CTI \tag{4.20}$$

where $T'_{i,k}$ and $T'_{j,k}$ are the operating times of the i-jth P/B relay pair under that transient condition.

Therefore, by collecting all the optimization parts given above, the standard optimization model[20] of classical ORC problems can be expressed as follows:

$$\begin{aligned} \min_{TMS,PS} \; & Z(TMS_1, \cdots, TMS_\varrho, PS_1, \cdots, PS_\varrho) \\ \text{Subjected to: } & T^{\text{pr}}_{i,k} + CTI - T^{\text{bc}}_{j,k} \leqslant 0 \\ & T^{\text{min}}_{i,k} - T^{\text{pr}}_{i,k} \leqslant 0 \\ & T^{\text{pr}}_{i,k} - T^{\text{max}}_{i} \leqslant 0 \\ & TMS^{\text{min}}_i \leqslant TMS_i \leqslant TMS^{\text{max}}_i \\ & PS^{\text{min}}_i \leqslant PS_i \leqslant PS^{\text{max}}_i \end{aligned} \tag{4.21}$$

where i and j notations are respectively used to represent the primary and backup relays, and the notation k represents the fault location.

[19] This location belongs to the ith primary relay R_i. Thus, this fault is considered as an out-zone fault for R_j.

[20] Refer to the first section of Chapter 2.

It has to be remembered that, if the transient selectivity constraint expressed in (4.20) is modeled in the ORC problem, then it should also be included in (4.21).

4.3 Possible Ways to Apply EAs to Solve ORC Problems

Once the mathematical model given in (4.21) is correctly formulated, any n-dimensional optimization algorithm can be applied to find the optimal TMS and PS. Three possible scenarios could be faced during designing any EA program:

- TMS and PS are discrete $\rightarrow$ real-coded EAs: This approach could be used to simulate electromechanical DOCRs.
- TMS is continuous and PS is discrete $\rightarrow$ mixed-integer EAs: This approach could be used to simulate solid-state DOCRs.
- TMS and PS are continuous $\rightarrow$ combinatorial EAs : This approach could be used to simulate state-of-the-art numerical DOCRs.

To do that, different techniques could be applied here, which are categorized and briefly described in the following subsections.

4.3.1 Applying Conventional Meta-Heuristic Optimization Algorithms

The conventional EAs given in Chapter 2 (i.e., DE, SA, and BBO) can be applied to any one of the preceding scenarios. For example, the conventional BBO algorithm shown in Figure 2.12 has been successfully used to solve some popular test systems of this subject [24, 40]. However, the ORC problems are well-known as highly stiff non-convex nonlinear optimization problems where many constraints need to be satisfied. Recently, some researches suggest to incorporating a pre-processing unit called a **feasibility checker** (**FC**). This sub-algorithm checks the selectivity constraint of each P/B relay pair and fixes it through a *while*-loop [40, 201]. Actually, this is a significant enhancement and a big step in developing superior optimizers, because this FC has the ability to detect feasible solutions within just a few iterations compared with hundreds and thousands of iterations as with conventional EAs [70, 280]. To explain it more, suppose that the conventional BBO algorithm shown in Figure 2.12

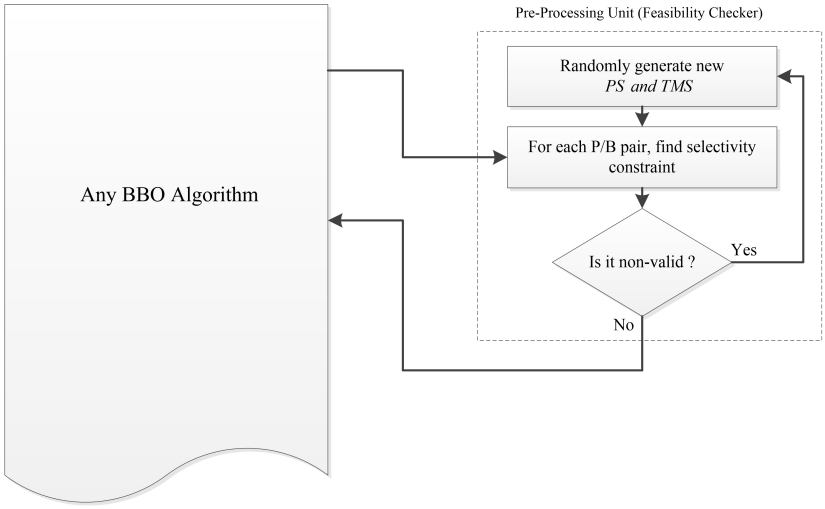


Figure 4.3: Mechanism of the Feasibility Checker within BBO

is used to solve an ORC problem. The mechanism of the FC sub-algorithm can be illustrated in Figure 4.3.

Also, some recent researches suggest to hybridize EAs with LP and NLP to accelerate the convergence rate. Such studies are [14, 70, 280, 297]. However, because the ORC problems are nonlinear, so there is a technical problem when LP is selected as a sub-algorithm in EAs. Detailed information about this problem and how to practically solve it in different ways is given below.

4.3.2 Tuning Relay Settings by Hybridizing EAs with LP

In Chapter 2, it has been said that to be able to apply LP as a sub-algorithm[21], the optimization model should be expressed in a linear form. Referring to the preceding ORC models, it is obvious that the objective function and the functional constraints are nonlinear. Thus, the first essential step to be used, before initiating LP, is to linearize the optimization model. The literature shows one possible way to do that, which is about taking PS supplied from EAs for all ϱ relays as a constant vector, and the TMS vector as an initial point to LP. Actually, there is also another approach that can be applied as well. This novel approach can be achieved by simply linearizing the model as a function of PS instead of TMS. However, a precaution should be given to the PS vector since its ϱ elements are discrete variables [39]. These two

[21] To fine-tune the elites supplied from EAs.

options for both the IEC/BS and ANSI/IEEE standards are covered in the following lines.

4.3.2.1 Fixing Plug Settings and Varying Time Multiplier Settings

To linearize the IEC/BS standard given in (4.4) for the ith relay, the following expression is used:

$$T_{i,k} = TMS_i \times \frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} = \psi_{i,k} TMS_i \; ; \; \psi_{i,k} = \frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} \tag{4.22}$$

where PS_i is constant, which is determined by EAs.

Substituting (4.22) in (4.21) for $T_{i,k}^{\text{pr}}$ and $T_{j,k}^{\text{bc}}$ yields:

$$\begin{aligned} Z &= \min \sum_{i=1}^{\varrho} \psi_{i,k} TMS_i \\ \text{Subjected to: } \psi_{i,k} TMS_i + CTI - \psi_{j,k} TMS_j &\leqslant 0 \\ T_{i,k}^{\min} - \psi_{i,k} TMS_i &\leqslant 0 \\ \psi_{i,k} TMS_i - T_{i,k}^{\max} &\leqslant 0 \\ TMS_i^{\min} \leqslant TMS_i &\leqslant TMS_i^{\max} \end{aligned} \tag{4.23}$$

Similarly, the ANSI/IEEE standard can be linearized as follows:

$$T_{i,k} = TMS_i \times \left[\frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} + \gamma\right] = \varphi_{i,k} TMS_i \; ; \; \varphi_{i,k} = \left[\frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} + \gamma\right] \tag{4.24}$$

and, again, PS_i is held constant for the LP stage.

Thus, by substituting (4.24) in (4.21) for $T_{i,k}^{\text{pr}}$ and $T_{j,k}^{\text{bc}}$, the following standard linear model can be obtained:

$$\begin{aligned} Z &= \min \sum_{i=1}^{\varrho} \varphi_{i,k} TMS_i \\ \text{Subjected to: } \varphi_{i,k} TMS_i + CTI - \varphi_{j,k} TMS_j &\leqslant 0 \\ T_{i,k}^{\min} - \varphi_{i,k} TMS_i &\leqslant 0 \\ \varphi_{i,k} TMS_i - T_{i,k}^{\max} &\leqslant 0 \\ TMS_i^{\min} \leqslant TMS_i &\leqslant TMS_i^{\max} \end{aligned} \tag{4.25}$$

This hybrid approach can be depicted by the flowchart shown in Figure 4.4 when BBO is used as the main algorithm.

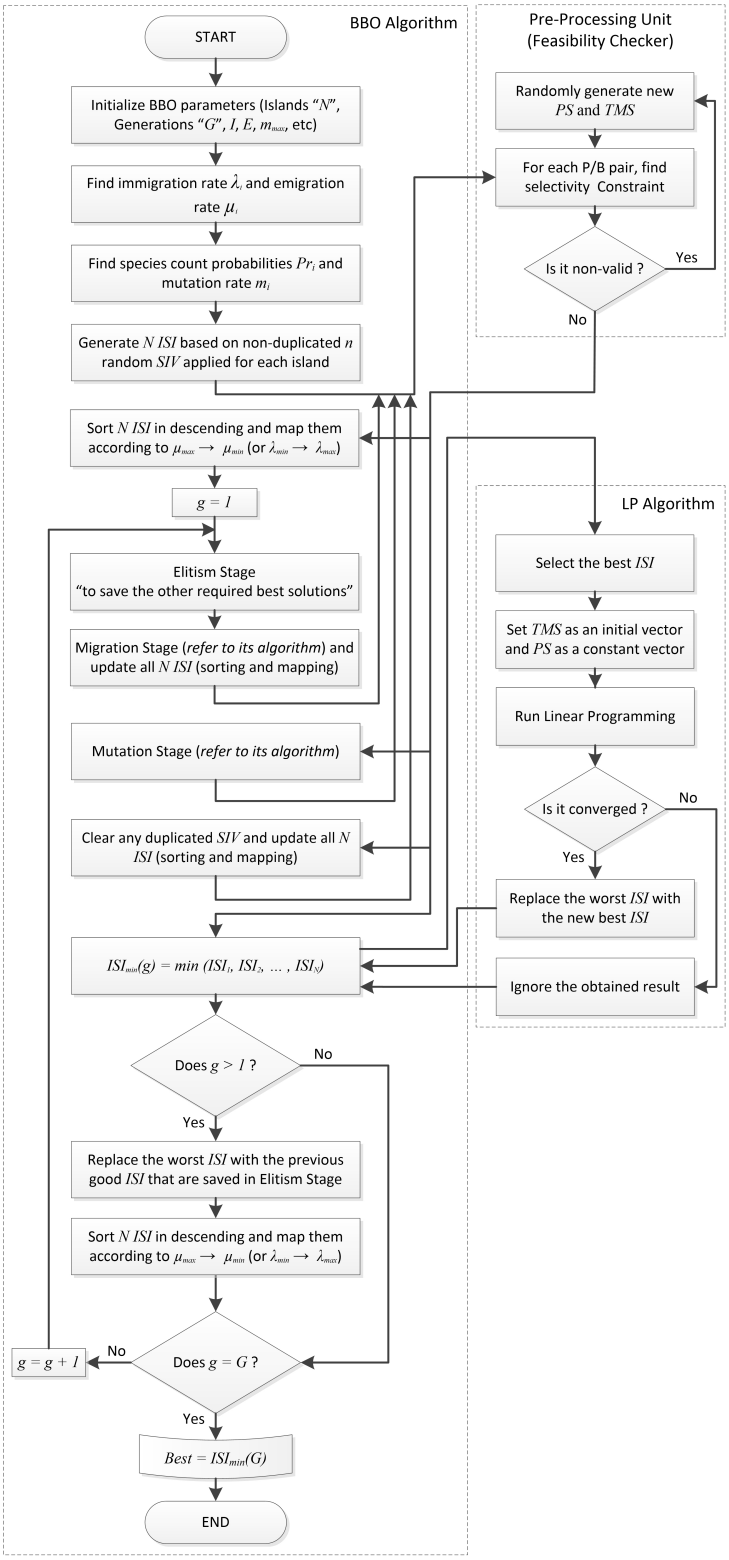


Figure 4.4: Flowchart of the TMS-Based BBO-LP Algorithm Reinforced by the Feasibility Checker Sub-Algorithm

4.3.2.2 Fixing Time Multiplier Settings and Varying Plug Settings

This approach is tricky and requires to involve the **transformation** technique used in linear regression analysis [214]. For the ith relay, the IEC/BS standard given in (4.4) can be linearized as follows:

$$T_{i,k} = \vartheta_i \widetilde{PS}_{i,k} \quad ; \quad \vartheta_i = \beta \; TMS_i \tag{4.26}$$

where TMS_i is constant, which is determined by EAs. $\widetilde{PS}_{i,k}$ is a transformed plug setting, which is built based on:

$$\widetilde{PS}_{i,k} = \frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}} \tag{4.27}$$

Substituting (4.26) in (4.21) for $T_{i,k}^{\text{pr}}$ and $T_{j,k}^{\text{bc}}$ yields:

$$\begin{aligned} Z = \min \sum_{i=1}^{\varrho} \vartheta_i \widetilde{PS}_{i,k} \\ \text{Subjected to: } \vartheta_i \widetilde{PS}_{i,k} + CTI - \vartheta_j \widetilde{PS}_{j,k} &\leqslant 0 \\ T_{i,k}^{\min} - \vartheta_i \widetilde{PS}_{i,k} &\leqslant 0 \\ \vartheta_i \widetilde{PS}_{i,k} - T_{i,k}^{\max} &\leqslant 0 \\ \widetilde{PS}_{i,k}^{\min} \leqslant \widetilde{PS}_{i,k} &\leqslant \widetilde{PS}_{i,k}^{\max} \end{aligned} \tag{4.28}$$

where the lower and upper bounds of $\widetilde{PS}_{i,k}$ can be respectively calculated as follows:

$$\widetilde{PS}_{i,k}^{\min} = \frac{\left(PS_i^{\min}\right)^{\alpha}}{I_{i,k}^{\alpha} - \left(PS_i^{\min}\right)^{\alpha}} \tag{4.29}$$

$$\widetilde{PS}_{i,k}^{\max} = \frac{\left(PS_i^{\max}\right)^{\alpha}}{I_{i,k}^{\alpha} - \left(PS_i^{\max}\right)^{\alpha}} \tag{4.30}$$

After fine-tuning the transformed plug setting $\widetilde{PS}_{i,k}$ by LP for the ith relay and at the kth fault location, the actual quantity PS_i can be retrieved by applying the following de-transformation formula:

$$PS_i = \left[\frac{\widetilde{PS}_{i,k} I_{i,k}^{\alpha}}{1 + \widetilde{PS}_{i,k}}\right]^{1/\alpha} = I_{i,k} \times \sqrt[\alpha]{\frac{\widetilde{PS}_{i,k}}{1 + \widetilde{PS}_{i,k}}} \tag{4.31}$$

For more information about this transformed optimization model, the full derivation is given in Appendix A.

Following the same transformation steps, the ANSI/IEEE standard given in (4.5) can be linearized as follows:

$$T_{i,k} = \vartheta_i \widetilde{PS}_{i,k} + \xi_i \quad ; \quad \vartheta_i = \beta \; TMS_i \quad , \quad \xi_i = \gamma \; TMS_i \tag{4.32}$$

where TMS_i is constant, which is determined by EAs. $\widetilde{PS}_{i,k}$ is a transformed variable, which is expressed in (4.27).

Substituting (4.32) in (4.21) for $T_{i,k}^{\text{pr}}$ and $T_{j,k}^{\text{bc}}$ yields:

$$\begin{aligned} Z = \min & \sum_{i=1}^{\varrho} \vartheta_i \widetilde{PS}_{i,k} + \xi_i \\ \text{Subjected to: } & \left(\vartheta_i \widetilde{PS}_{i,k} + \xi_i\right) + CTI - \left(\vartheta_j \widetilde{PS}_{j,k} + \xi_j\right) \leqslant 0 \\ & T_{i,k}^{\min} - \left(\vartheta_i \widetilde{PS}_{i,k} + \xi_i\right) \leqslant 0 \\ & \vartheta_i \widetilde{PS}_{i,k} + \xi_i - T_{i,k}^{\max} \leqslant 0 \\ & \widetilde{PS}_{i,k}^{\min} \leqslant \widetilde{PS}_{i,k} \leqslant \widetilde{PS}_{i,k}^{\max} \end{aligned} \tag{4.33}$$

where the lower and upper bounds of $\widetilde{PS}_{i,k}$ are already given in (4.29)-(4.30).

Again, once the transformed plug settings $\widetilde{PS}$ of all ϱ DOCRs are fine-tuned by the LP sub-algorithm, the corresponding actual plug settings PS can be retrieved by de-transforming the former settings via (4.31). The full derivation about this transformed optimization model is given in Appendix B. Figure 4.5 graphically explains the major difference with the last BBO-LP flowchart.

4.3.2.3 Fixing/Varying Time Multiplier Settings and Plug Settings

This is the most advanced hybrid approach between EAs and LP. It incorporates the advantages of both algorithms shown in Figures 4.4-4.5. That is, two separated LP sub-algorithms are coded in two different places of BBO for both TMS and PS. Thus, by this hybridization strategy, the user can ensure that both settings are fine-tuned. However, there are some couple questions that are still open and require detailed answers! For example: *Which relay setting should be fine-tuned first, TMS or PS? Is it required to fine-tune both settings in every iteration? What is about the exploration level of the overall algorithm? Is it useful to embed the* ***jumping rate (Jr)*** *technique used in the* ***opposition-based learning (OBL)*** *algorithm? What is the suitable value of Jr? How fast is it with/without Jr?*

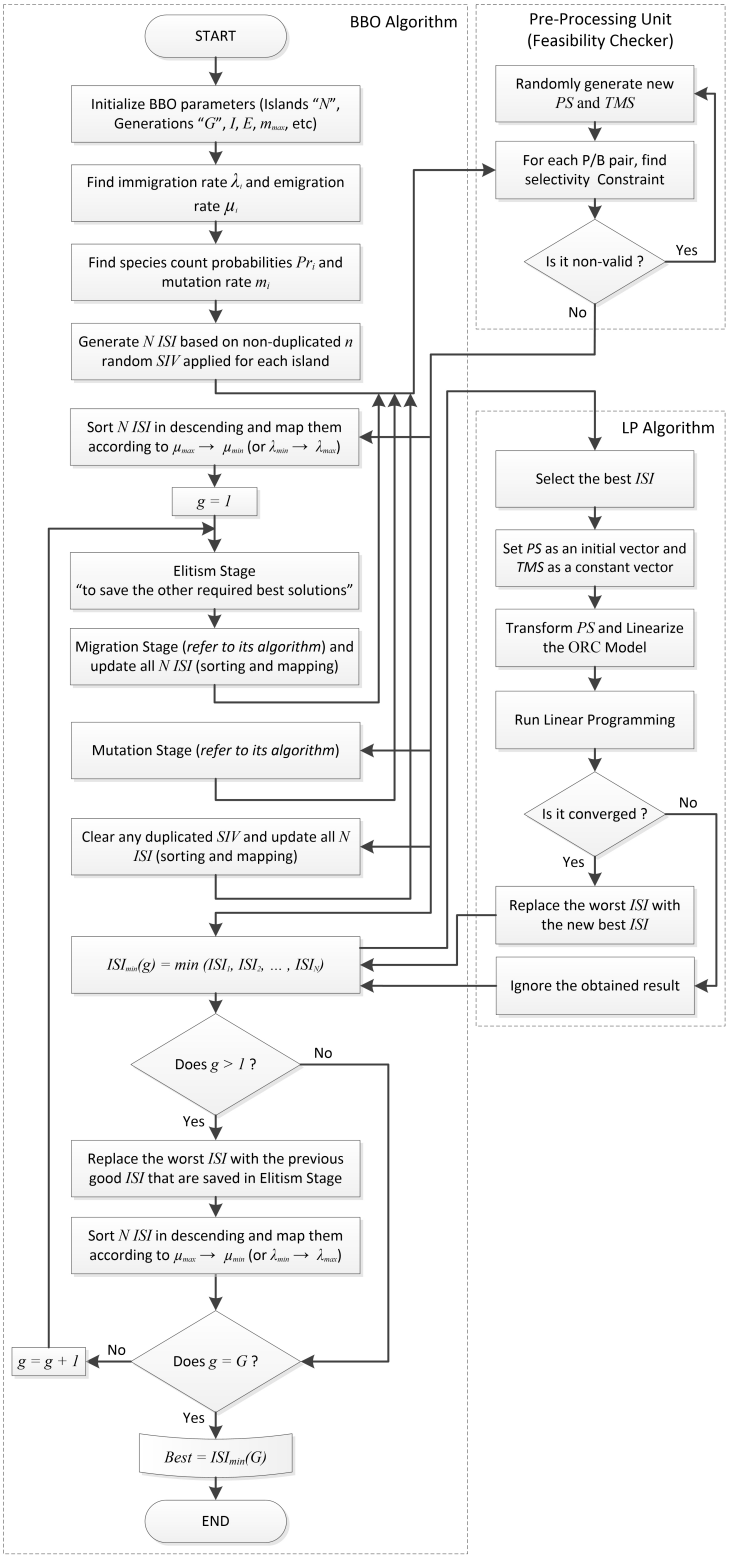


Figure 4.5: Flowchart of the PS-Based BBO-LP Algorithm Reinforced by the Feasibility Checker Sub-Algorithm

This topic is still fresh. Actually, based on an extensive literature review, no one yet talked about it. This is the first time ever. The flowchart shown in Figure 4.6 reveals the mechanism of this innovative hybrid optimization algorithm.

It has to be always remembered that the PS variables are mostly discrete, and for some relay technologies, both PS and TMS are discrete. Because the strategy here is to fine-tune these settings in a separate mode[22], so both LP sub-algorithms or only the LP sub-algorithm responsible to fine-tune PS could be provided as **integer linear programming** (**ILP**). Based on the nature of relay technologies, it is impossible to have LP to fine-tune PS and ILP to fine-tune TMS. That is, either both PS and TMS or only PS can be fine-tuned by ILP.

4.3.3 Classical Approach to Enhance the Performance by Using Multiple TCCCs

If each DOCR has its own TCCC, then different curves have to be used. This means that the relay parameters $\{\alpha, \beta, \gamma\}$ are not constant anymore. Thus, instead of dealing with just two decision variables in each DOCR (i.e., PS and TMS), they will become five, which in turn increases the ORC problem dimension n by 250%.

Based on the relay technology[23], TCCC could be restricted to some **European** and **North American standards**, which are tabulated in Table 4.2 and depicted in Figure 4.7 for the most popular standards [56, 147], or even user-defined curves by optimizing the parameters $\{\alpha, \beta, \gamma\}$.

To ensure having feasible solutions, there are four main corrections that have to be applied to optimization algorithms, which are:

- If someone wants to apply the preceding hybrid optimization algorithms shown in Figures 4.4-4.6, then these three parameters have to be set as constants in the LP sub-algorithm(s). This can be easily done by fixing the latest optimal values of $\{\alpha, \beta, \gamma\}$ that were detected by BBO or any other EA.

- In the BBO sub-algorithm, the problem dimension n can be automatically calculated based on the number of branches being protected by DOCRs. This can

[22] This is because of the linearization done in the ORC model.
[23] It can be electromechanical, solid-state, digital "hardware-based", or numerical.

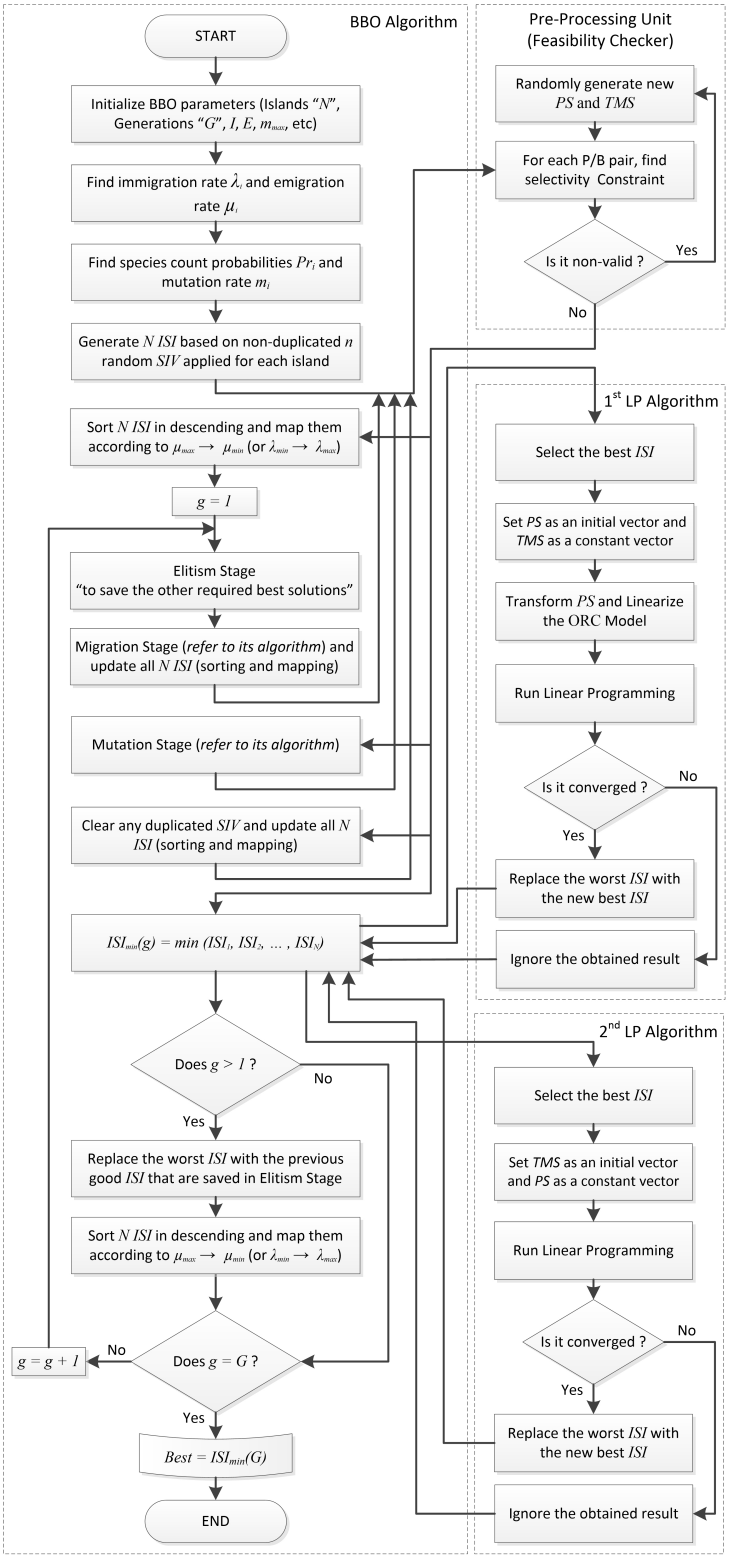


Figure 4.6: Flowchart of the TMS/PS-Based BBO-2LP Algorithm Reinforced by the Feasibility Checker Sub-Algorithm

Table 4.2: Standard European and North American Inverse TCCCs

Type of Curve	**Standard**	α	β	γ
Moderately Inverse	IEEE	0.02	0.05150	0.11400
Very Inverse	IEEE	2.00	19.6100	0.49100
Extremely Inverse	IEEE	2.00	28.2000	0.12170
Inverse	CO-8[a]	2.00	5.95000	0.18000
Short-Time Inverse	CO-2	0.02	0.02394	0.01694
Standard Inverse[b]	IEC	0.02	0.14000	0.00000
Very Inverse	IEC	1.00	13.5000	0.00000
Extremely Inverse	IEC	2.00	80.0000	0.00000
Long-Time Inverse	AREVA / UK	1.00	120.000	0.00000
Short-Time Inverse	AREVA	0.04	0.05000	0.00000

[a] **CO-x** are brands of the Westinghouse's electromechanical OCRs, which were acquired by ABB since 1989. The acronym "CO" stands for "**Circuit Opening**". The OCRs listed under this series are: **CO-2**, **CO-5**, **CO-6**, **CO-7**, **CO-8**, **CO-9** and **CO-11** [24, 56, 147].

[b] It is also called the **inverse definite minimum time** (**IDMT**).

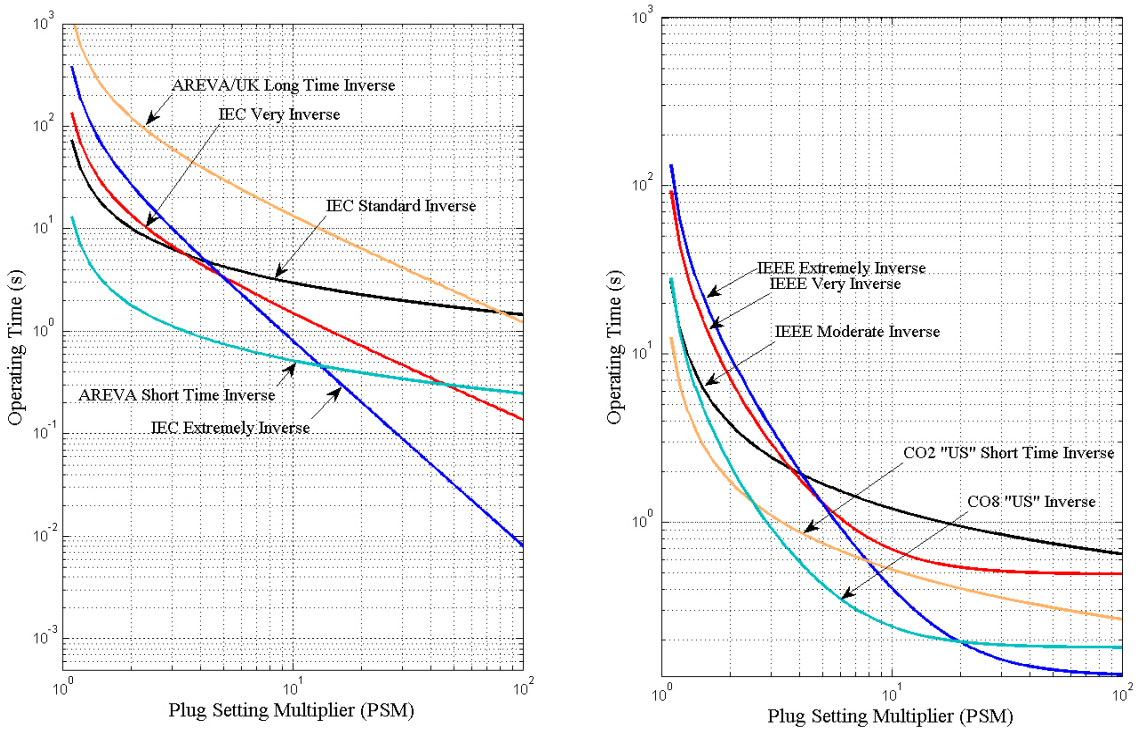
be done by modifying (4.2) to be as follows:

$$n = 5\varrho = 10ß \tag{4.34}$$

However, if standard TCCCs are used, then their parameters $\{\alpha, \beta, \gamma\}$ are linked together. Thus, instead of using three variables of scalar type, it is possible to use only one variable of vector type. For that case, the dimension n can be effectively reduced down to:

$$n = 3\varrho = 6ß \tag{4.35}$$

- The lower and upper bounds of each TMS_i depend on the standard TCCC adopted for the ith relay. It starts from 0.1 to 1.1 for the European standard TCCCs and from 0.5 to 11 for the North American standard TCCCs [46]. Although the acronyms TMS and TDS have the same meaning and are used interchangeably in the literature, both acronyms could be seen in the same reference. The main reason for that is to differentiate between the two standards where TDS is used for the North American standard TCCCs and TMS is used for the European standard TCCCs. Thus, to match the IEC/BS and the



(a) European Standards

(b) North American Standards

Figure 4.7: Time-Current Characteristic Curves of Inverse-Time OCRs (TMS=1.0)

ANSI/IEEE models, given in (4.4)-(4.5), in the same optimizer, the following adjustment is required [56, 147]:

$$T_{i,k} = \frac{TDS_i}{7} \times \left[\frac{\beta_i}{PSM_{i,k}^{\alpha_i} - 1} + \gamma_i \right] \tag{4.36}$$

where $TMS_i = \left(\frac{TDS_i}{7}\right)$, and $PSM_{i,k}$ is called the **plug setting multiplier**, which is here calculated at the kth fault location for the ith relay. By referring to (4.4)-(4.5), $PSM_{i,k}$ equals to:

$$PSM_{i,k} = \frac{I_{i,k}}{PS_i} \tag{4.37}$$

- Since the parameters $\{\alpha, \beta, \gamma\}$ are considered as new variables in the BBO sub-algorithm, so FC should be modified to deal with this change. In Figure 4.8, which is modified from Figure 4.3, it is obvious that all the five variables of each relay are subject to change until the i-jth P/B relay pair is satisfied.

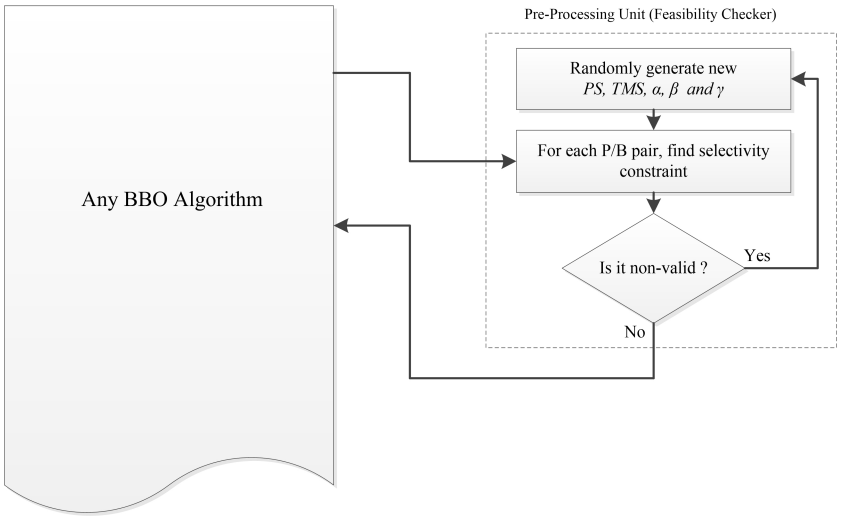


Figure 4.8: Mechanism of FC within BBO when Different TCCCs are Considered

Based on our recent investigation about the performance of this approach, it has been found that there are many unforgiven technical issues associated with it that make this approach impractical [14].

4.3.4 Effective Approach to Feasibly Enhance the Performance by Searching for the Optimal Unified TCCC

It has been seen in the last subsection how to implement multiple TCCCs in solving ORC problems. However, that approach complicates the whole ORC model. Some of these complications are summarized as follows:

- The problem dimension is higher than the normal one by 150% to 250%[24].
- As a consequence of that increase, the algorithm becomes slower.
- Based on the last point, this ORC solver is inappropriate for adaptive coordination schemes.
- Because of that increase in the dimension, the algorithm requires more generations G and/or population size N to explore the search space for the optimal or near optimal solutions.

[24]It is 250% if each one of the parameters $\{\alpha, \beta, \gamma\}$ occupies one decision element in the optimization algorithm, and that dimension can be reduced down to just 150% if the preceding three parameters are optimized together as a vector of length 3.

- To deal with these additional variables, the algorithm requires many modifications in its main structure.

- Moreover, using many TCCCs could satisfy all the design constraints, but they might also force the optimizer to settle on unselective settings. This phenomenon could happen based on the fact that satisfying optimal coordination at the near-end 3ϕ fault could not be correct for other fault locations[25]. That is, because different TCCCs are imposed in the model, so the smallest CTI does not always equal the value measured at the highest fault current. Thus, dealing with many TCCCs means that the selectivity constraint between each P/B relay pair must be satisfied by considering a gradient of fault locations to cover all the length of each branch from the near-end to the far-end bus. Therefore, continuing to use the multiple TCCCs approach means that the most exhausting objective function given in (4.6) should be used again to prevent having infeasible solutions.

Instead, dealing with only one TCCC unified to all ϱ DOCRs makes the mission much simpler than before. For one TCCC approach, most of the papers presented in the literature use the IEC/BS IDMT-TCCC. Unfortunately, the other European and North American standards get less attention, although they might provide better solutions. For modern numerical relays, the user can adjust his/her own TCCC, so this option can increase the chance to get better solutions by finding just one TCCC unified to all the relays and satisfies the optimality criterion.

4.3.4.1 Numerical Experiment on the 15-Bus Test System

This numerical experiment provides a comprehensive analysis on the performance of these TCCCs when both discrete and continuous PS are considered[26]. The 15-bus test system shown in Figure 4.9 is used in this study. The full information about this test system is given in [25]. The TMS-based BBO-LP algorithm depicted in Figure 4.4 is used in this experiment.

[25] This point is covered in the last section of Chapter 8.
[26] To simulate the performance of electromechanical and numerical relays.

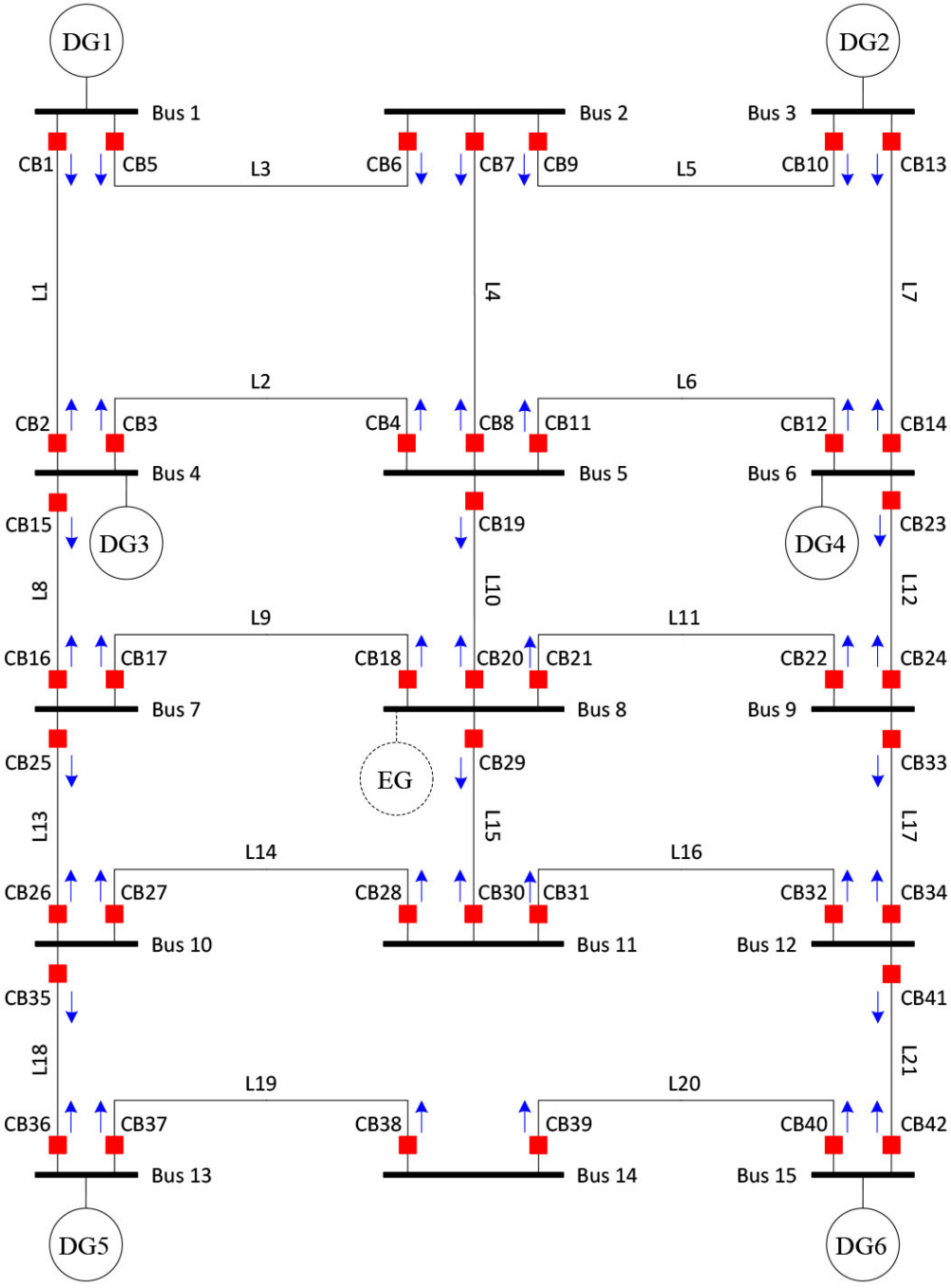


Figure 4.9: Single-Line Diagram of the 15-Bus Test System

Possible Structures of the ORC Solver

Here, two scenarios are considered. The first one focuses on searching for one unified standard TCCC, and the other scenario focuses on searching for one user-defined TCCC. The first scenario can be easily satisfied by selecting one standard TCCC at each simulation run and in the end displaying the best type. This technique is shown in Figure 4.10a. The second scenario can be satisfied by two different techniques:

1. Executing the BBO/BBO-LP algorithms many times with randomly generated $\{\alpha, \beta, \gamma\}$. Thus, the problem dimension n is kept without any change. But, this technique needs Tr number of simulation runs or trials to optimize the preceding parameters as shown in Figure 4.10b.

2. The other technique treats these parameters as three additional variables, so the problem dimension becomes:

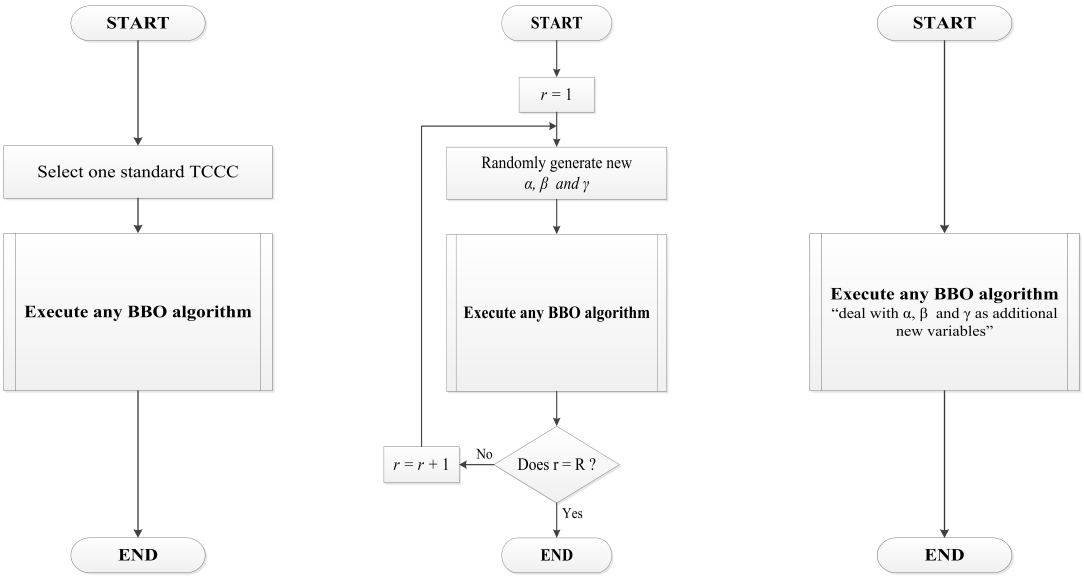
$$n = 3 + 2\varrho = 3 + 4ß \tag{4.38}$$

   Thus, the mission is given to the optimizer to find the best user-defined values of $\{\alpha, \beta, \gamma\}$. This technique is shown in Figure 4.10c. Although the problem dimension is slightly increased, the third technique is very smart and much faster than the second technique shown in Figure 4.10b.

Numerical Experiment

The test system shown in Figure 4.9 is an example of electric power networks with **distributed generation** (**DG**). This system consists of 15 buses and 21 branches, and hence it has 42 DOCRs and 82 P/B relay pairs with 84 variables[27]. The total constraints are 250, and addressed as: 82 inequality constraints for the P/B selectivity criteria, 42 inequality constraints for the minimum allowable operating times $T^{\min}$, 42 inequality constraints for the maximum allowable operating times $T^{\max}$, 42 side constraints for the time multiplier settings TMS and 42 side constraints for the plug settings PS.

[27] If $\{\alpha, \beta, \gamma\}$ are considered as constants.



(a) First Technique (b) Second Technique (c) Third Technique

Figure 4.10: Different Techniques to Change TCCC within BBO/BBO-LP Algorithms

The two scenarios are tested by considering two options. The first option uses discrete PS to simulate the operation of electromechanical relays, and the second option uses continuous PS as one of many advantages of modern numerical relays[28]. Both BBO and BBO-LP are used to minimize the sum of operating times of all ϱ DOCRs when they act as primary protective devices.

For having a fair speed performance comparison with seeker algorithm presented in [51], the three innovative optimization algorithms shown in Figure 4.10 are coded using MATLAB R2011a and simulated in a computing machine having the following specifications: Pentium IV, 3.06 GHz, and 512 MB RAM with 32 bit Windows XP SP3 operating system. The algorithm initialization parameters are tabulated in Table 4.3. For the first scenario, the plug settings are considered discrete in uniform steps of 0.5 A.

Results and Discussion

[28]In real-world problems, both PS and TMS are discrete settings. This will be covered later in Chapter 8. Thus, continuous and discrete settings are used to simulate the actual operation of relays manufactured by different technologies. This approximation has been seen in [110, 111].

Table 4.3: Initialization Parameters of BBO and BBO-LP for the 15-Bus Test System

Relay	Gen. #	PopSize	TMS		PS (A)			CTI	BBO Settings	
Technology	G	N	min	max	min	max	step-size	(s)	m_{max}	R_m
Electromechanical	385	120	> 0.1	1.1	0.5	2.5	0.5	0.2	0.1	50%
Numerical	30	20	> 0.1	1.1	0.5	2.5	0.5	0.2	0.1	50%

Table 4.4 shows the results obtained by the BBO and BBO-LP algorithms when PS are treated as discrete and continuous independent variables. From the first look on Table 4.4, it can be clearly observed that the violations happen when short-time inverse TCCCs are used; for both the European and North American standards. That is, as the rate of change of TCCC becomes low, the algorithm will require more effort to detect feasible solutions. Therefore, with these slow response TCCCs, satisfying the selectivity constraints of the P/B relay pairs becomes very hard. This infeasibility problem can be easily avoided if the hybrid BBO-LP algorithm is used. Besides, it is obvious that when PS of ϱ relays are discrete, the FC sub-algorithm requires more CPU time to find feasible solutions by these limited elements of the PS vector. This phenomenon happens with both BBO and BBO-LP versions. However, most of the results obtained by discrete PS show better performance than continuous PS; for both BBO and BBO-LP. The reason is that the continuous PS mode needs more generations to explore the search space for good solutions. A similar thing can be observed with user-defined TCCCs. To be on the safe side, the lower and upper bounds of the TCCC parameters $\{\alpha, \beta, \gamma\}$ are the lowest and highest values tabulated in Table 4.2; for each corresponding parameter (i.e., $\alpha \in [0.02, 2]$, $\beta \in [0.05, 120]$, and $\gamma \in [0, 0.491]$). Thus, by searching for optimal continuous values within these side constraints, the optimization algorithms require more generations to explore this non-restricted/open domain. However, if enough number of generations and population size are provided, then there is a higher chance to get better solutions.

By referring to the last two rows of Table 4.4, the third technique shown in Figure 4.10c is significantly much faster than the second technique shown in Figure 4.10b. Also, it performs better even without hybridizing it with the LP sub-algorithm. This is logical because the second technique uses one set of $\{\alpha, \beta, \gamma\}$ at each simulation run, while the third technique optimizes these three parameters as new three variables as seen in (4.38). In this contest, the IEC/BS extremely inverse TCCC always scores

the best position compared with the others if the LP sub-algorithm is not activated or if continuous plug settings are used for all ϱ DOCRs. The ANSI/IEEE extremely inverse TCCC scores the best position when the LP sub-algorithm is activated with considering discrete plug settings.

We think that this innovative approach is much better than the preceding weary approach described in the last subsection[29] and used in all the studies presented in the literature. To see how it is useful, Table 4.5 shows the results obtained by using multiple TCCCs [14]. In this case, the original dimension n significantly increases by 150% to 250%, and the best smallest result obtained by this approach is much bigger than that obtained by our effective approach.

From Table 4.5, there are two visible issues and one hidden issue. The first visible issue is the complexity of the optimization algorithms if they are built with these many extra variables. The second visible issue, which is related to the first one, is the processing speed. As a rule of thumb, the processing speed decreases as the problem dimension n increases (i.e., CPU time $\propto n$). The third issue, which cannot be seen from that table, is the feasibility of ORC problem solutions. If multiple TCCCs are used to optimize the coordination problem of these ϱ DOCRs, then the obtained solution is feasible under the given short-circuit currents calculated for the defined P/B relay pairs. As per the information available for this test system, the short-circuit analysis was carried out using near-end 3ϕ faults on all the lines. Thus, the optimal solutions obtained for this ORC problem using the multiple TCCCs approach might not be feasible for other fault locations, such as at the far-end or middle point [34]. Based on this vital fact, using our one optimal TCCC approach for all ϱ DOCRs can satisfy the feasibility, optimality, simplicity, and processing speed performance criteria.

[29] i.e., the one that uses multiple TCCCs.

Table 4.4: Performance Comparison Between BBO and BBO-LP Algorithms for the 15-Bus Test System

Type of TCCC	Discrete Plug Setting						Continuous Plug Setting					
	BBO			BBO-LP			BBO			BBO-LP		
	Fitness	Violations	CPU	Fitness	Violations	CPU	Fitness	Violations	CPU	Fitness	Violations	CPU
IEEE Moderately Inverse	21.40	**0**	649.97	12.60	**0**	15.10	21.86	**0**	321.67	12.82	**0**	10.93
IEEE Very Inverse	7.86	**0**	435.64	5.02	**0**	12.86	8.32	**0**	201.20	5.17	**0**	9.72
IEEE Extremely Inverse	3.33	**0**	333.92	**2.28**	**0**	12.09	3.27	**0**	147.98	2.31	**0**	9.27
Inverse "CO-8"	5.39	**0**	1031.71	4.47	**0**	23.60	5.68	**0**	623.02	4.95	**0**	18.80
Short-Time Inverse "CO-2"	18.20	8	1060.25	10.95	**0**	20.71	18.82	11	559.25	12.01	**0**	14.55
IEC Standard Inverse (IDMT)	33.84	**0**	488.83	13.76	**0**	13.58	32.81	**0**	231.65	13.75	**0**	10.33
IEC Very Inverse	7.54	**0**	320.26	5.24	**0**	11.04	7.00	**0**	134.64	5.00	**0**	8.64
IEC Extremely Inverse	**2.09**	**0**	**273.89**	2.45	**0**	**10.56**	**2.01**	**0**	119.35	**2.26**	**0**	8.66
AREVA/UK Long-Time Inverse	32.11	**0**	284.83	32.22	**0**	10.75	39.15	**0**	**114.92**	33.32	**0**	**8.16**
AREVA Short-Time Inverse	15.60	9	1026.76	10.17	**0**	20.47	16.49	6	538.95	11.55	**0**	14.36
User-Defined "Technique No.2"	3.00	**0**	3576.6	2.56	**0**	115.47	2.74	**0**	2049.76	2.53	**0**	93.97
User-Defined "Technique No.3"	2.44	**0**	440.78	2.89	**0**	12.86	2.32	**0**	208.93	3.35	**0**	9.77

Table 4.5: Performance Comparison Between the Classical IDMT-Based Approach and the Multiple TCCCs Approach

Comparison Criteria	**Only IDMT**			**Standard TCCCs**						**User-Defined TCCCs**	
	Seeker [51]	**BBO**	**BBO-LP**	**BBO**[1]	**BBO**[2]	**BBO**[3]	**BBO**[4]	**BBO**[5]	**BBO**[6]	**BBO**[7]	**BBO**[8]
Population Size	120										
TCCC Standards	Only IEC IDMT			[1]All	[2]N. American	[3]European	[4]IEEE/IEC	[5]IEEE	[6]IEC	User-Defined	User-Defined
PS Mode	Discrete									[7]Discrete	[8]Continuous
Generation No.	385										
Variables No.	84			210 if using (4.34) or 126 if using (4.35)						210	
Min. OBJ (s)	12.227	33.842	12.609	7.622	5.941	7.311	5.683	5.833	**5.260**	8.992	8.826
CPU Time (s)	**406.3**	488.8	568.8	3498.5	3546.1	3422.1	3265.3	3315.0	3177.3	1798.9	925.8
Is It Feasiblee?	No	Yes[a]									

[a]This feasibility is seen under the given short-circuit analysis. Thus, there is no guarantee of the feasibility under other unconsidered fault locations since the time-current characteristic curves are not unified.

Chapter 5

Hybridizing Meta-Heuristic Optimization Algorithms with Machine Learning Tools to Optimally Forecast Energy and Power Data of Nova Scotia

5.1 Overview

Electric energy is one of the most essential keys to indicate the economic growth of any country. It is one of the most common forms of energy that are continually used by many consumers; starting from individual users and ending with industries. No doubt that there is a highly significant relationship between electric energy and business cycles of most countries.

Forecasting electric energy gets high attention from many researchers. In electric power systems engineering, this field is known as **electric power forecasting**[1] or **demand forecasting**. It can be seen as an independent study or as an embedded tool in many other studies, such as power system operation and planning.

The amazing thing in electric demand forecasting compared with other forecasting studies is that the first forecasting study can be presented as accumulative forecasting of other forecasting studies. To clarify this point, suppose that a private energy company wants to forecast the daily power consumption. Then, many factors have direct and indirect effects on that analysis. For example, the weather, economy, social and political activities, sports events, holidays, etc. Thus, the forecasting models of these individual factors can be seen together inside the power forecasting model. That is, precise readings of these factors can lead to a precise estimation of power.

[1] Actually, **energy forecasting** is a more broad topic, where the term "**power**" is "**energy**" per unit of "**time**". In modern electric power systems, especially **smart grids**, many alternatives of energy can be seen, such as electric and heat energy.

5.1.1 The Importance of Energy Forecasting and Its Horizons

To follow the rapid economic growth, the electricity industries become more complicated to be operated, monitored, controlled, and protected. That is for the utility side, while for the consumers' side, the electricity production should meet the requirement of power any time they need it; no matter if the load is at its peak, regular, or valley point. The dependency should always be met, otherwise the system is considered unreliable. The other thing that has to be fulfilled is the total operating cost of that power. This includes the production, transmission, utilization, and maintenance costs, which raises the question of **economic operation** [370].

Fortunately, with modern electric power systems, all the preceding issues can be technically solved by what is called an **energy management system** (**EMS**[2]). Analyzing the major components of EMS, someone may find that there are different forecasting tools used within that system. These tools can be broadly divided into, but not limited to, the following areas [109, 370, 386]:

- Load Demand Forecasting
- Electricity Market Forecasting
- Fuel Market Forecasting
- Wind[3] Forecasting
- Sunlight[4] Forecasting
- Hydro[5] Forecasting

Also, each one of these forecasting sub-areas can be further categorized based on the forecasting horizons[6,7] [172, 211, 386]:

- **Very Short-Term Forecasts**: minutes ahead
- **Short-Term Forecasts**: from a few minutes to a few days ahead
- **Medium-Term Forecasts**: from a few days to a few months ahead
- **Long-Term Forecasts**: months, quarters, or even years

[2]Detailed information about the EMS environments and their functionalities, including both the hardware and software aspects, can be found in [370, 386].

[3]It is used with **wind farms** to predict the power output of **wind turbines**.

[4]It is utilized to predict the amount of power that can be generated from **solar power plants**.

[5]It is a direct measure used in **hydroelectric generation plants** to know how much power can be extracted from **hydro turbines**.

[6]In other words, the length of the dataset being used.

[7]These different terms of energy forecasting are not fixed where each jurisdiction has its own definition [172].

Table 5.1: The Nature of Forecasts, Lead Times, and Applications

Nature of Forecast	Lead Time[a]	Applications
Very Short-Term	A few seconds to several minutes	Generation, distribution schedules, and contingency analysis for system security
Short-Term	Half an hour to a few hours	Allocation of spinning reserve, operational planning and unit commitment, and maintenance scheduling
Medium-Term	A few days to a few weeks	Planning for seasonal peak-winter and peak-summer
Long-Term	A few months to a few years	Planning generation growth

[a]Please, refer to the last footnote for the same reason why each reference has different lead times.

These four categories can be further explained in Table 5.1 [211]. In spite of the period specified for each category, each one of these categories can also be affected by their updating frequency of that data[8]. Thus, the forecasts can be named based on their data resolution as well. For example, an hourly forecast, a daily forecast, a monthly forecast, a quarterly forecast, and an annual forecast. That is, a monthly forecast (i.e., updated every month) can be a daily forecast (i.e., updated every day), and it can also be an hourly forecast (i.e., updated every hour) [172]. In these three forecasts, the horizon covers months of records where the resolution depends on the updating manner of each individual forecast. Mathematically speaking, this can be explained as follows:

$$\text{RES}_{\text{hourly forecast}} > \text{RES}_{\text{daily forecast}} > \text{RES}_{\text{monthly forecast}} \tag{5.1}$$

where $\text{RES}_{\varpi \text{ forecast}}$ denotes the resolution on a ϖ basis. This basis could be seconds, minutes, hours, days, months, etc.

As can be seen, there are many forecasting horizons and resolutions. Table 5.2 lists the recommended horizon and updating window for each specific application in electric power systems [386].

[8]It can be defined as the step-size resolution taken during measuring the real data [172].

Table 5.2: Power Applications of Demand Forecasts

Power Application	Forecast Horizon	Forecast Intervals
Automatic Generation Control (**AGC**)	Next 15 min	5 sec
Economic Load Dispatch (ELD)	Next hour	30 sec
Power Flow (**PF**)	Next 2 days	5 min
Optimal Power Flow (OPF)		
Contingency Analysis (**CA**)	Next 2 days	10 min
Situational Awareness (**SA**)	Next hour	120 samples per second
Voltage Stability (**VS**)	Next hour	120 samples per second
Unit Commitment (**UC**)	Next 14 days	Hourly
Transaction Evaluation and Management (**TEM**)	Next 14 days	Hourly
Wind Forecasting	Next 5-60 min	30 sec
Hydro Forecasting	Next 14 days	Hourly
Fuel Scheduling	Next 1-6 months	Weekly

5.1.2 Some Fundamentals of Forecasting Models

As can be clearly seen in the preceding subsection, there are many definitions and terminologies used to describe the term "forecasting". For power and energy forecasting, the main question that might be raised here is: *What are the independent variables that have significant contributions in explaining the variability of power and energy?* To be able to answer this very crucial question, let's first define the following symbols:

- P_D: actual power demand
- $\tilde{P}_D$: forecasted power demand
- t: time
- $\breve{T}$: temperature
- RH: relative humidity
- ω_s: wind speed
- ω_d: wind direction
- u_r: sun rays
- u_d: dust
- u_c: clouds
- u_s: social and political events
- u_e: economic growth

As seen, many independent variables can be used as **predictors** to estimate the actual **response** P_D. Some of these predictors are significant for some specific applications, which are not necessarily significant for other applications. For example,

if the symbol f is used to denote the function or model used to forecast the actual power P_D, then for:

- **Predicting Load Demand:**

  Based on many references, such as [109, 172, 211, 254, 370, 386], the load demand can be forecasted by considering the time, temperature[9], relative humidity, social/political events, and economic growth. Thus, the forecasted power for this particular application is:

$$\tilde{P}_D = f(t, \breve{T}, \mathrm{RH}, u_s, u_e) \tag{5.2}$$

- **Harvesting Wind Power:**

  It is known that the power output of wind turbines mainly depends on wind speed and direction. These two variables are not constant[10]. Thus, the time itself should be provided as a variable to get a more accurate forecast as follows:

$$\tilde{P}_D = f(\omega_s, \omega_d, t) \tag{5.3}$$

  Also, the temperature and relative humidity could be used to explain the variability for a very short period, because at night the temperature and relative humidity are different than those at afternoon. Similar thing during the summer and winter seasons. Thus, (5.3) becomes:

$$\tilde{P}_D = f(\omega_s, \omega_d, t, \breve{T}, \mathrm{RH}) \tag{5.4}$$

- **Harvesting Solar Power:**

  A similar thing can be applied here where the power output harvested from solar panels depends on the sun rays, clouds (shading), dust (especially if there is a sand storm), and time (from the sunrise to the sunset). Thus, the forecasted power can be determined as:

$$\tilde{P}_D = f(u_r, u_c, u_d, t) \tag{5.5}$$

  If the temperature and relative humidity are also considered, then (5.5) becomes:

$$\tilde{P}_D = f(u_r, u_c, u_d, t, \breve{T}, \mathrm{RH}) \tag{5.6}$$

[9] For example, heating during the winter season and air conditioning during the summer season.

[10] They vary during the morning, noon, afternoon, and night; and during summer, winter, spring, and autumns.

The model might face a high correlation between some predictors, such as "temperature vs sun rays", "clouds vs temperature", "sun rays vs clouds", etc. Also, some predictors could be not significant for some horizons. Adding many predictors could lead to complex models that might consume more CPU time with very little improvement in the performance. Thus, the final selected model should compromise between different criteria, such as solution quality, computation time, and model complexity. For the first performance criterion, the easiest way to measure that is to take the absolute difference as follows:

$$\mathrm{Err}_{\mathrm{abs}} = \left| P_D - \tilde{P}_D \right| \tag{5.7}$$

The other effective metrics to evaluate the solution quality are: **coefficient of correlation** (**R**), **coefficient of determination** ($\mathbf{R^2}$), **mean squared error** (**MSE**), **root mean squared error** (**RMSE**), **mean absolute error** (**MAE**), **mean absolute percentage error** (**MAPE**), **error sum of squares** (**SSE**), etc. Of course, each one of these methods has its pros and cons [214].

As can be seen in the preceding equations, it is important to know which predictors should be considered in any forecasting model. This can save both the time and effort spent to model the problem. The smart and automatic approach to do that is to implement some **machine learning** (**ML**) tools to select the proper set of predictors. This process is called the **feature selection**.

5.2 Literature Review on Techniques Used to Forecast Electric Power

Based on a quick review in some references, it has been found that the literature is highly crowded by many topics in electric power forecasting. This can be clearly observed by searching for the keyword "power demand forecasting" in **Google Scholar** where the results are almost two million. Thus, it is a very hard task to trace all the methods and techniques used today to forecast electric power demand. In general, the main[11] techniques covered in electric power textbooks are summarized in [254, 386], which are listed in Table 5.3.

[11] Or, more precisely, the most popular techniques covered, nowadays, in many textbooks.

Table 5.3: List of Load Forecasting Models

Category	Electric Load Model	Uncertainty Model
Deterministic	Curve Fitting - Straight Line - Exponential Correlation Models - Econometric Models - Weather Models	None
Probabilistic	Curve Fitting Regression Models Correlation Models - Econometric Models - Weather Models	Uncertainty is treated as a random variable
Stochastic	Time Series - Auto-Regressive (**AR**) - Moving Average (**MA**) - Auto-Regressive Moving Average (**ARMA**) - Auto-Regressive Integrated Moving Average (**ARIMA**) - Others (**ARMAX**, **ARIMAX**, **SARMAX**, **SARIMAX**, **NARMA**)	Uncertainty is treated as a stochastic process
AI-Based	Artificial Neural Networks (ANNs) Knowledge-Based Expert System (**KBES**) Others	It depends on the technique involved

There are many other ways to categorize these forecasting models. For example, in [211], the authors categorize forecasting models into: extrapolation, correlation, and extrapolation/correlation. In [333], the author cites a book that divides forecasting models into: empirical and model-based, extrapolation and causal, and static and dynamic. In [173], the author divides them into:

- Traditional approaches, including Box–Jenkins autoregressive integrated moving average (ARIMA) model, autoregressive moving average with exogenous variables (ARMAX) model, seasonal autoregressive integrated moving average (**SARIMA**) model, exponential smoothing models - including Holt–Winters model (**HW**) and seasonal Holt–Winters' linear exponential smoothing (**SHW**) -, state space/Kalman filtering models, and linear regression models.
- AI-based approaches, including knowledge-based expert system (KBES) model,

artificial neural networks (ANNs) models, and fuzzy inference system models.

- Support vector regression (**SVR**) models and their hybrid/combined models.

It is obvious that there are many ways to categorize these forecasting models. Moreover, the preceding models are not a completed list of what can be seen in the literature. One of the interesting studies is introduced in [148], which gives an overview of the latest novel, modified, and hybrid energy demand forecasting methods published in the period between 2005-2015.

5.3 Approaches to Hybridize Global Optimizers with ANNs and SVRs

Before diving directly into the subject, it is important to first briefly introduce the two ML computing systems.

5.3.1 Brief Introduction to Artificial Neural Networks (ANNs)

Any Artificial neural network (ANN) can be defined as a group of interconnected virtual **neurons** (also called **units**, **nodes**, and **processing elements**) created by some algorithms to describe the actual behavior of the biological neurons of animal brains; or, in other words, an electronic network designed to simulate the process of the biological neurons. This biological behavior was first modeled in some mathematical expressions by Warren S. McCulloch and Walter Pitts where their paper, written in 1943 and reported in [252], becomes a seminal work. It opens a new era of computing intelligence. Actually, that work is considered as the basis of many modern neural networks presented in the literature [160].

This highly advanced computing system can be illustrated by thinking of it as a group of interconnected nodes that act as virtual neurons in computing machines. These nodes work together, through a learning process, to solve many very complicated mathematical problems within a very short time compared with classical methods that consume lots of time with expensive information[12]. The input variables are processed through these nodes after being normalized and weighted with some biases. Thus, the overall knowledge can be represented by these virtual neurons where

[12]The speed here is measured after completing the learning phase; i.e., with trained neural networks.

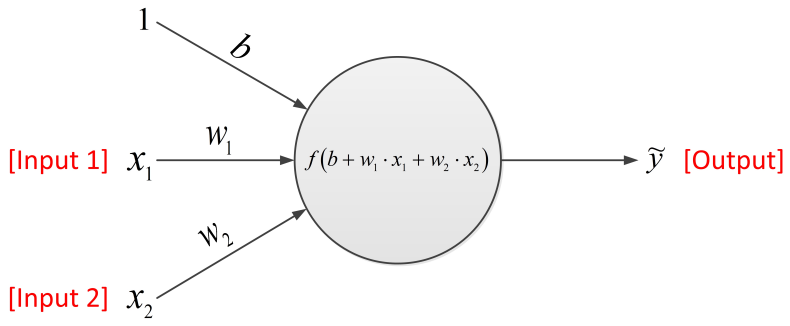


Figure 5.1: Mathematical Representation of a Single Neuron

the priority or contribution of each neuron is controlled through these weights and biases. That is, the biological neural networks can be electronically modeled to solve many real-world problems directly without expressing any mathematical model, and this is why ANNs are called **black-boxes**. Nowadays, there are many applications of ANNs, such as regression analysis (or **function approximation**), **clustering and categorization**, **pattern classification**, **forecasting/prediction**, **control and robotics**, optimization, etc.

5.3.1.1 Single Neuron

To understand the mechanism of ANNs, let's first build a very primitive network by using just one neuron. Knowing how this simple network works will allow us to quickly understand the basic idea behind ANNs.

The process of this neuron can be illustrated in Figure 5.1. The purpose of this node is to collect the inputs coming from an external source (i.e., the actual inputs) or other nodes, and then computes its output. By referring to Figure 5.1, there are two inputs (x_1 and x_2). To provide an output $\tilde{y}$, which is an estimate of the actual output y, the virtual neuron multiplies these two inputs respectively by **weights** (w_1 and w_2) and then adds the result to a **bias** b as follows:

$$\tilde{y} = f\left(b + w_1 x_1 + w_2 x_2\right) \tag{5.8}$$

Thus, for n inputs, the output y can be mathematically expressed as:

$$\tilde{y} = f\left(b + \sum_{i=1}^{n} w_i x_i\right) = f\left(\sum_{i=0}^{n} w_i x_i\right) \tag{5.9}$$

where $w_i = b$ and $x_i = 1$ when $i = 0$.

To reduce the error between the actual response y and the approximated response $\tilde{y}$ by this single neuron, a proper function f should be applied with optimal weights (w_1 and w_2) and bias (b or w_0). Good normalized values for these three parameters can be obtained by **training** this simple network via an optimization algorithm. The function f is called the **activation function** or the **transfer function**. It could be linear by using an identity function, or nonlinear by using any standard or user-defined function[13].

5.3.1.2 Feed-Forward Neural Network

The feed-forward structure is the simplest and first fully functioning type of ANNs. It can be designed by connecting multiple nodes (virtual neurons) arranged in vertical and horizontal directions. Each vertically aligned group of nodes is called a **layer**. The first layer (i.e., placed on the far left) is called the **input layer**, and the last layer (i.e., located on the far right) is called the **output layer**. The layer located in between is called the **hidden layer**. The input and output layers always exist, while the hidden layer depends on the total number of layers imposed on the network. If there are Ł layers, then the number of hidden layers $\hbar$ can be calculated as follows:

$$\hbar = Ł - 2 \tag{5.10}$$

where $Ł \geqslant 2$.

If $Ł = 2$, then the neural network does not have a hidden layer. This special type is called a **single-layer perceptron** (**SLP**). Otherwise, the neural network has at least one hidden layer. Thus, adding one or multiple hidden layers between the input and output layers will lead to what is called a **multilayer perceptron** (**MLP**). Furthermore, if $\hbar = 1$ (i.e., $Ł = 3$), then there is only one hidden layer. This special type of MLP is called a **non-deep** or **shallow neural network**. Thus, a **deep**

[13]Some of the standard activation functions available in the literature are: Bent Identity, SoftPlus, Binary Step, Logistic (SoftStep or Sigmoid), Sigmoid Linear Unit (SiLU), Soft Clipping, SoftExponential, Softmax, Maxout, Sinusoid, Sinc, ArSinH, TanH, ArcTan, Gaussian, Inverse Square Root Unit (ISRU), Inverse Square Root Linear Unit (ISRLU), SoftSign or ElliotSig, Square Nonlinearity (SQNL), Rectified Linear Unit (ReLU), Bipolar Rectified Linear Unit (BReLU), Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), Randomized leaky Rectified Linear Unit (RReLU), Parametric Rectified Linear Unit (PReLU), S-shaped Rectified Linear Activation Unit (SReLU), Leaky Rectified Linear Unit (LReLU), and Adaptive Piecewise Linear (APL)

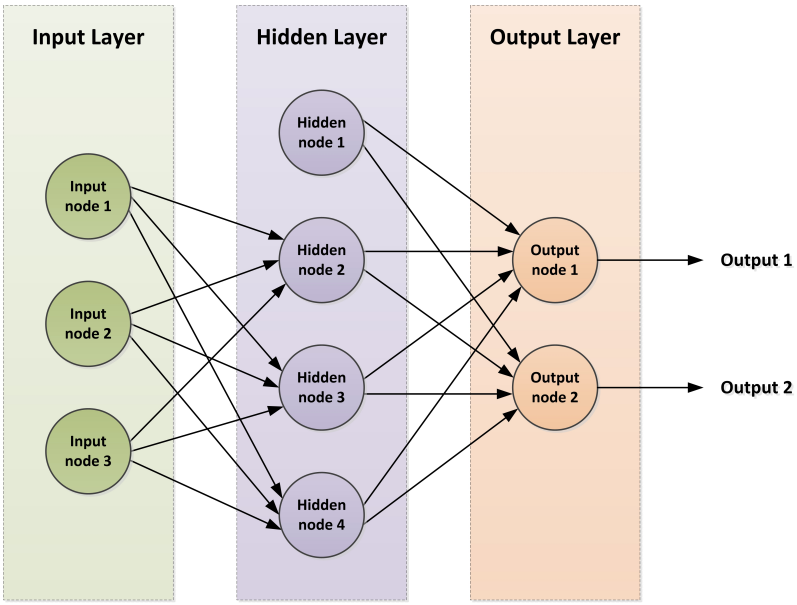


Figure 5.2: Simple Structure of a Shallow Neural Network

neural network is MLP constructed by multiple hidden layers. A simple shallow neural network is shown in Figure 5.2.

By merging the information gained from Figure 5.1 in Figure 5.2, a more informative illustration is given in Figure 5.3. The neuron highlighted in red can be mathematically expressed by any activation function assigned to the weighted sum of (5.9). Again, finding the best weights assigned to all the neural network nodes is done by training that network via an optimization algorithm[14]. A good literature review with detailed information about modern ANNs can be found in [160].

5.3.1.3 Hybridizing Meta-Heuristic Optimization Algorithms with ANNs to Optimize Their Hyperparameters

It has been seen that to approximate an output y there are many configurations with many open questions regarding the optimality of these configurations. Some of these headache questions are: *What is the best neural network type to use? What is the best number of hidden layers? What is the best number of neurons or nodes assigned to each hidden layer? What is the best activation function assigned to each hidden*

[14]Some popular algorithms will be seen later when a numerical experiment is conducted.

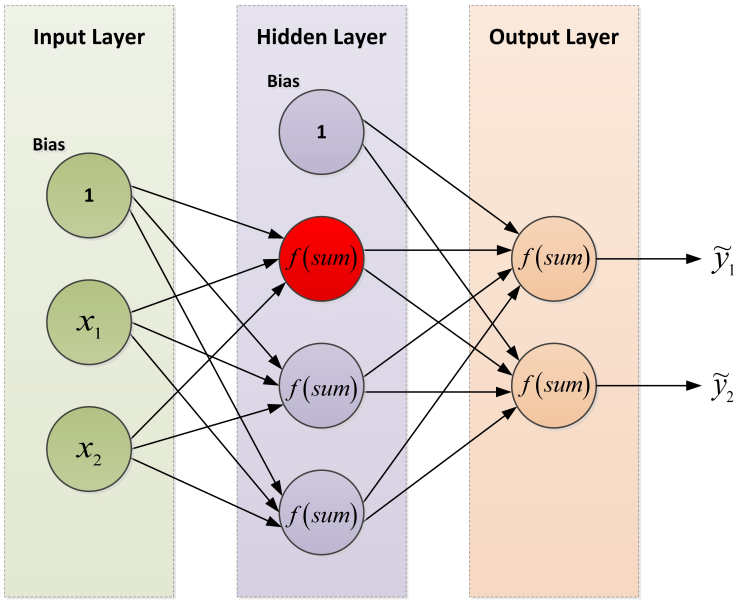


Figure 5.3: Shallow Neural Network with Detailed Information

layer? What is the best activation function assigned to the output layer? What is the best optimization algorithm that should be used to train the network? What are the optimal settings of the training algorithm? What is the best feature set? Does any one of these features need to be transformed? How many epochs does the optimization algorithm need to complete its training process? Which configuration has the best compromise between the solution quality and the processing speed? etc.

For example, Figure 5.4 shows five possible feed-forward neural-network topologies to approximate a response based on six predictors. All the topologies shown are of MLP type. One of them is a shallow neural network designed with ten hidden nodes and all the six predictors. The others have more than one hidden layer with an unequal number of hidden nodes and different feature subsets. The dilemma is concentrated in not knowing which topology is the best. The word "best" itself needs a clear definition! *Does it mean the simplicity, accuracy, speed, or a compromise between some or all of them?*

In some recent studies, EAs have been successfully hybridized with ANNs. The modern structures of these hybrid ANNs/EAs can be classified into two main approaches as shown in Figure 5.5. The checkmark means that this approach is implemented in this chapter.

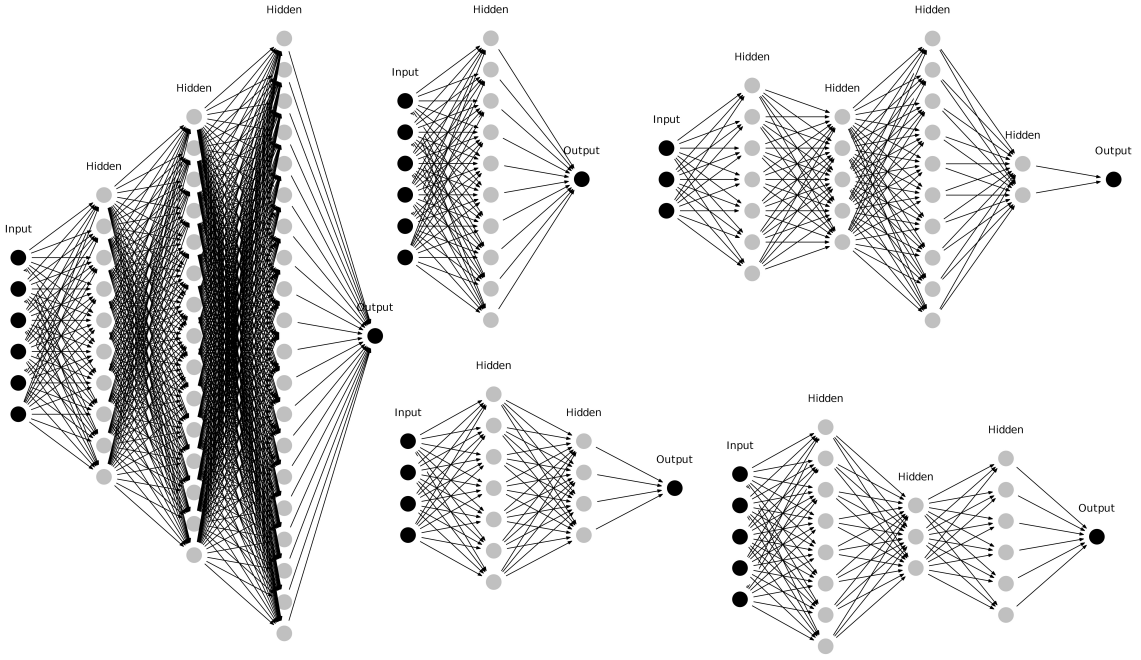
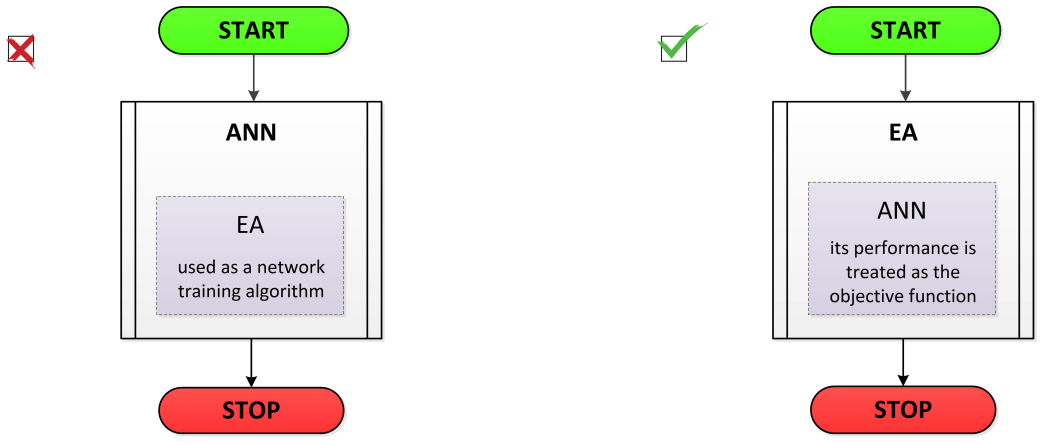


Figure 5.4: Neural Networks with Different Topologies



(a) Strategy No.1

(b) Strategy No.2

Figure 5.5: Main Strategies to Hybridize EAs with ANNs

- **EA as a part of ANN:**

  EAs are embedded as network training algorithms. In this approach, the classical **back-propagation** (**BP**) algorithms are replaced with EAs. The goal of this hybridization is to avoid trapping into local optimum solutions and to accelerate the learning process. However, some researchers claim that this approach is very slow compared with gradient-based algorithms, and thus they are preferred for extremely *high*-dimensional problems. This approach is depicted in Figure 5.5a, which is not implemented in this dissertation.

- **ANN as a part of EA:**

  The idea of this approach is to find the best configuration (or hyperparameters) of the given network. That is, the goal is to determine which input variables should be selected (i.e., feature selection), how many layers should be used, how many neurons should be assigned to each of these layers, the type of activation functions used in each layer (including both the output layer and the hidden layers), the type of training algorithm, etc. The process is like running EAs in their normal operation mode where the objective function that needs to be optimized is the performance of the neural network. This performance can be measured by many evaluation metrics available in the literature, such as MSE, RMSE, MAE, MAPE, R^2, R, etc. This approach is depicted in Figure 5.5b.

Of course, there are some more advanced possible approaches. For example:

- Hybridizing both BP algorithms and EAs into the same training block to have what are called "global-local" optimizers.
- Hybridizing both Figure 5.5a and Figure 5.5b by using one external EA to optimize the configuration of ANNs and one internal EA to train ANNs. That is, finding both the optimal parameters (i.e., the weights and biases) and the optimal hyperparameters (i.e., the best number of neurons and hidden layers, the best activation function types, the best training algorithm and its internal settings, the best network topology, etc).
- The last two approaches can also be combined to have one external EA and one internal BP/EA.

5.3.2 Brief Introduction to Support Vector Machines (SVMs)

Based on the literature review, Vladimir N. Vapnik is a renowned name. In 1963, Vapnik introduced the generalized portrait algorithm [372], which can be considered as the basis of modern SVMs. Some researchers link the background history of SVMs back to 1936 when Ronald A. Fisher, in [132], suggested the first algorithm - called "**discriminant analysis**" - for **pattern recognition**. It has to be said that SVMs are built based on the statistical learning theory. Thus, in contrast with ANNs, SVMs can provide firm analysis with a solid theoretical foundation. They have the advantage of theoretical completeness, global optimization, good adaptability, and generalization ability [78,281]. To trace the full history of SVMs, please refer to [247].

5.3.2.1 Difference Between SVM and SVR

In regression field, there are two types of variables called **continuous** and **categorical**. The problems come with the continuous type are solved by classical linear and nonlinear regression algorithms. For the second type, different logistic regression algorithms can be applied. The same thing can be seen in SVM where the classical SVM algorithm is used to solve classification and clustering problems. For continuous variables, a special algorithm called a **support vector regression** (**SVR**), which is an extension of the classical SVM algorithm, is used to predict continuous variables [321]. The report given in [351] provides a self-contained introduction to SVR with some detailed theoretical background and mathematical explanation, which is out of the scope of this chapter.

5.3.2.2 Hybridizing the Basic Random Search Algorithm with SVRs to Optimize Their Hyperparameters

By referring to the preceding forecasting examples, all the variables measured from power and weather stations are continuous. Thus, to be able to use SVM as a forecaster, the special SVR algorithm should be applied here. Without going into too much detail, the SVR algorithm given in [305] has the following hyperparameters:

- **kernel:** `string`, (default F_{SVR} ='`rbf`')
  The other possible kernels are: '`linear`', '`poly`', and '`sigmoid`'

- **degree:** `int`, (default $d = 3$)
  It is exclusively used with the polynomial kernel '`poly`'
- **gamma:** `float`, (default $\gamma =$'`auto`')
  Kernel coefficient for '`rbf`', '`poly`', and '`sigmoid`'
- **coef0:** `float`, (default coef0 $= 0$)
  Independent term for '`poly`' and '`sigmoid`' kernels
- **C:** `float`, (default $r_C = 1.0$)
  Penalty parameter C of the error term
- **epsilon:** float, (default $\varepsilon = 0.1$)
  Epsilon in the epsilon-SVR model
- **shrinking:** boolean, (default shrk =`True`)
  Whether to use the shrinking heuristic

The other hyperparameter that affects the overall performance of SVR is the subset ($U \subseteq X$) of the features selected to predict the actual response. That is, the performance of SVR is enhanced by optimizing these hyperparameters. The most basic **random search algorithm** (**RSA**) is used as a global optimizer to find the best SVR hyperparameters. This primitive algorithm can work as a population-based or as a single-search algorithm. It does not require any knowledge from biology or any other discipline; as faced with EAs. Instead, it can be considered as a systematic **trial and error** process. The reason for selecting this basic optimizer is to achieve two important criteria; the simplicity and the processing speed. It can reach an acceptable solution quality within acceptable processing time. Of course, going with any evolutionary optimization algorithm[15] will increase the chance to get better solutions, and it could be faster in some forecasting problems[16]. Based on our observations, during analyzing our hybrid BBO-ANN algorithm, it has been found that the population-based EAs are time-consuming techniques. Furthermore, the exploration level is the most important factor in selecting the best global optimizer because the exploitation level could be compensated by the internal optimizers of ML tools.

[15]Including some of our innovative optimization algorithms presented in Chapter 2.

[16]The reason is simple, EAs can learn from their previously obtained solutions. Thus, EAs could require relatively few iterations to converge.

There is no doubt that the random search method is a very fast algorithm compared to both classical and modern optimization algorithms because it is a derivative-free technique and it can be initiated with one candidate solution. Furthermore, because the solution generated at each iteration does not depend on the preceding iteration, as with classical and modern optimization algorithms, so it has a very high exploration level. However, a large number of iterations is required to get good solutions because the mechanism of RSA depends on a purely stochastic process. Instead, the following two possible approaches can be tried:

1. Executing the basic RSA in multiple stages. For each new stage, the total number of iterations is reduced by half and the side constraints are adjusted around the best solution obtained by the last stage.

2. Hybridizing the basic RSA with any EA. In this approach, there are just two stages. For the first stage, RSA is executed for a specific number of iterations. Then, in the second stage, the other smart EA algorithm is executed to search around the pre-optimal solution found by RSA. To reduce the total CPU time, it is good to design such hybrid optimization algorithms with SA or TS since they are single-search optimizers. GA, PSO, and others are population-based optimizers, so they are time-consuming algorithms.

The second advanced approach could be covered in future work. The pseudocode given in Algorithm 10 describes the programmatic procedure of the first approach; i.e. using RSA with multiple stages. The non-zero positive integer value of χ_j given in line 4 of Algorithm 10 can be calculated as follows:

$$\chi_j = \left\lfloor 2^{-j} s_{\varsigma}^{-1} G_{\mathrm{RSA}} \right\rceil \tag{5.11}$$

and s_{ς} is a geometric series sum of ς terms, which can be calculated as follows:

$$\begin{aligned} s_{\varsigma} &= \sum_{j=1}^{\varsigma} r^j \quad , \quad \text{where} \quad r = \frac{1}{2} \\ &= 1 - 2^{-\varsigma} \end{aligned} \tag{5.12}$$

where the full derivations of χ_j and s_{ς} are given in Appendix C.

Algorithm 10 Random Search Algorithm to Optimize SVR Hyperparameters

Require: ς = number of stages needed to run RSA, G_{RSA} = number of iterations assigned to RSA, G_{SVR} = maximum number of iterations that the SVR algorithm can use, E_{SVR} = acceptable tolerance to stop the internal optimizer of the SVR algorithm, default SVR hyperparameters (F_{SVR} =‘`rbf`’, $d = 3$, γ =‘`auto`’, coef0 = 0, $r_C = 1.0$, $\varepsilon = 0.1$, shrk =`True`, and $U = X$)

1: **for** each stage j, $j \in [1, \varsigma]$ **do**
2: Split the dataset into train and test datasets (80% ratio is used)
3: Find MSE_0 using the default hyperparameter settings, and save both the settings and MSE_0 in the 1st row of the jth solution matrix $\text{MAT}(j, 1, :)$
4: **for** each iteration i, $i \in [1, \chi_j]$ **do**
5: Generate random hyperparameters $\Upsilon = \{F_{\text{SVR}}, d, \gamma, \text{coef0}, r_C, \varepsilon, \text{shrk}, U\}$[a]
6: Find MSE_i of SVR using Υ
7: **if** $\text{MSE}_i < \text{MSE}_{i-1}$ **then**
8: Save MSE_i and Υ in $\text{MAT}(j, i+1, :)$
9: **else**
10: Discard the new solution and keep the old one
11: **end if**
12: **end for**
13: Minimize the domain of $\{F_{\text{SVR}}, d, \gamma, r_C, \varepsilon, \text{shrk}, U\}$ to be close to the best configuration obtained in the last iteration of χ_j
14: **end for**

[a]For the first stage, the side constraints of these hyperparameters are taken as follows:

- $F_{\text{SVR}} = \{$‘`rbf`’, ‘`linear`’, ‘`poly`’, ‘`sigmoid`’$\}$
- $d \in [1, 9]$
- $\gamma \in [0.001, 100]$
- coef0 $\in [-1000, 1000]$
- $r_C \in [0.001, 100000]$
- $\varepsilon \in [0.001, 10]$
- shrk $= \{$`True`, `False`$\}$
- $U \subseteq X$

5.4 Numerical Experiments on Optimally Configured ANNs to Forecast Some Nova Scotian Heat Consumption Profiles

In this section, only the second approach shown in Figure 5.5b is implemented to find the best configuration of some very simple multilayer feed-forward ANNs. To do that, a new hybrid approach between ANN and BBO[17] is introduced. That is, the BBO algorithm is selected as the EA part shown in Figure 5.5b. For the ANN part, of course, recycling the load variable into the input vector could improve the solution quality; which might be considered in future work. However, this study keeps everything as simple as possible just to prove the effectiveness of employing EAs to optimize the hyperparameters of ANNs. The same concept can be applied to any other structure, including **convolutional neural networks** (**CNNs**), **time delay neural networks** (**TDNNs**), **long short-term memory** (**LSTM**), **Hopfield**, etc.

This experiment represents a medium-term forecast for some energy consumptions in Nova Scotia, Canada. The actual energy and **metrology measurements** were hourly recorded for the whole days of January 2012 as shown in Figure 5.6. The length of the dataset is 744 for 31 days. However, some few missing hours are filtered out from the records as can be clearly noticed in the periodic shape of the top-left plot. The source of these metrology measurements is the "**Halifax Dockyard Weather Station**", which can be accessed via: Government of Canada > Environment and Natural Resources > Weather, Climate, and Hazard webpage [381].

The actual energy measurements (i.e., the responses or dependent variables) are taken from Nova Scotia Power Inc.[18] [285]. Three types of heat energy are forecasted in this experiment. The First two types are representative profiles of a typical Nova Scotian customer. The first one is for residential non-electric heat and the second one is for residential electric heat, and both were measured in watt-hour. The third type represents the total municipal heat, which was measured in kilowatt-hour. These three measurements are plotted in Figure 5.7.

The program is coded in MATLAB 2017b and simulated using the following computing machine: ALIENWARE X51 Desktop, 64-bit Windows 10 OS, Intel Core

[17] The full description of BBO is given in Chapter 2.

[18] It is privately owned by Emera and regulated by the provincial government via the Nova Scotia Utility and Review Board (NSUARB).

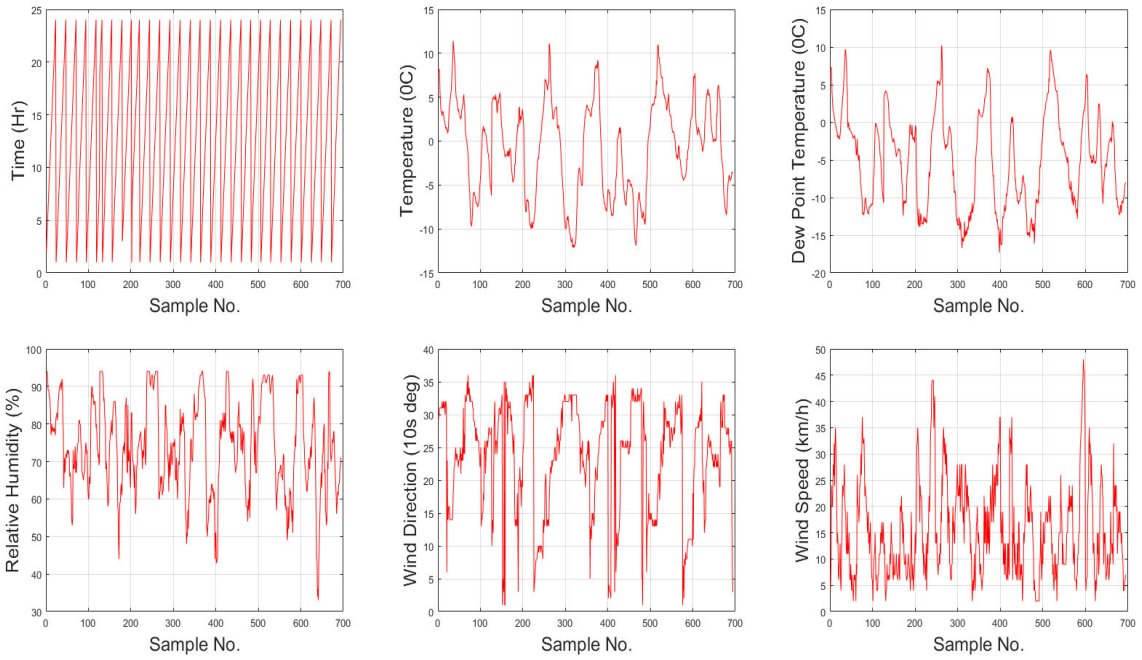


Figure 5.6: Metrology Data of Nova Scotia (Hourly Recorded for January 2012). From Top, Left to Right: Sampling Time (t_h), Temperature $\breve{T}$, and Dew Point Temperature ($\breve{T}_{\text{dp}}$). From Bottom, Left to Right: Relative Humidity (RH), Wind Direction (ω_d), and Wind Speed (ω_s)

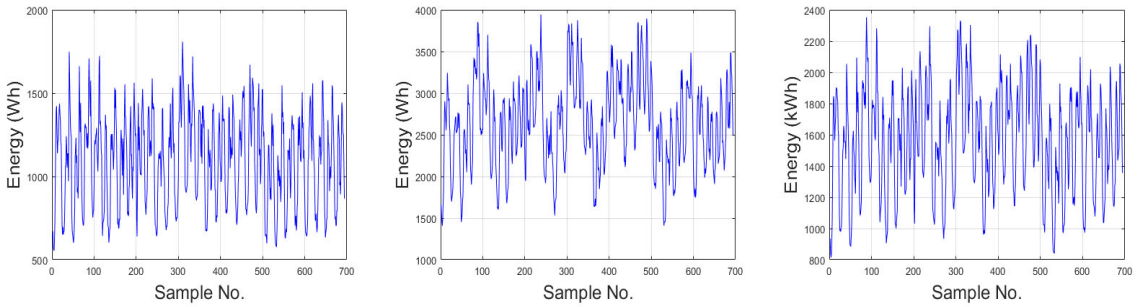


Figure 5.7: Heat Energy Data of Some Nova Scotian Customers (Hourly Recorded for January 2012). From Left to Right: Residential Non-Electric Heat, Residential Electric Heat, and Municipal Heat

i7-6700 CPU @ 3.4 GHz, and 8 GB RAM. The initialization parameters of BBO are:

- Population size: 50
- Number of generations: 1000
- User-defined probability mutation: 0.1
- Mutation range: 50%
- Elitism: 1
- Clear duplication stage: disabled
- Migration rates: linear "standard"

Also, the initialization parameters of ANNs are:

- Maximum number of epochs to train: 1000
- Performance goal: 0
- Maximum validation failures: 6
- Maximum number of layers: 4
- Maximum number of neurons per layer: 30
- Ratio of vectors for training = 70%
- Ratio of vectors for validation = 15%
- Ratio of vectors for testing = 15%

During optimizing the neural network hyperparameters, the BBO algorithm searches for the best combination of activation functions used in the hidden layers and the output layer. The algorithm searches within the following fifteen activation functions:

- `compet`: Competitive transfer function
- `elliotsig`: Elliot sigmoid transfer function
- `hardlim`: Positive hard limit transfer function
- `hardlims`: Symmetric hard limit transfer function
- `logsig`: Logarithmic sigmoid transfer function

- `netinv`: Inverse transfer function
- `poslin`: Positive linear transfer function
- `purelin`: Linear transfer function
- `radbas`: Radial basis transfer function
- `radbasn`: Radial basis normalized transfer function
- `satlin`: Positive saturating linear transfer function
- `satlins`: Symmetric saturating linear transfer function
- `softmax`: Soft max transfer function
- `tansig`: Symmetric sigmoid transfer function
- `tribas`: Triangular basis transfer function

A similar thing is done for the training process where the following twelve gradient-based training algorithms are used:

- `trainlm`: Levenberg-Marquardt
- `trainbfg`: BFGS Quasi-Newton
- `trainrp`: Resilient Backpropagation
- `trainscg`: Scaled Conjugate Gradient
- `traincgf`: Fletcher-Powell Conjugate Gradient
- `traincgp`: Polak-Ribiére Conjugate Gradient
- `trainoss`: One Step Secant
- `traingdx`: Variable Learning Rate Gradient Descent
- `traingdm`: Gradient Descent with Momentum
- `traingd`: Gradient Descent
- `traincgb`: Conjugate Gradient with Powell/Beale Restarts
- `traingda`: Gradient Descent with Adaptive Learning Rate

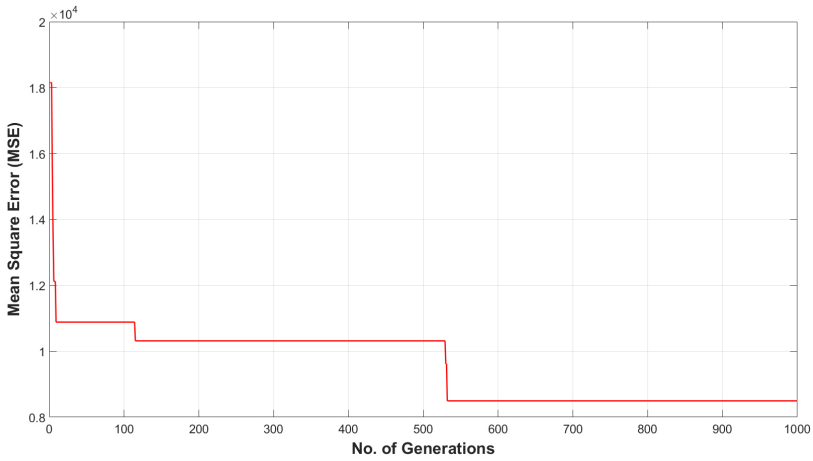
Table 5.4: Optimal Hyperparameters of the Three ANNs (Objective Function: MSE)

Performance Criteria	Residential Non-Electric Heat	Residential Electric Heat	Municipal Load
MSE	8494.2137	32609.2119	1452719.0585
MAE	68.3635	147.5984	919.5616
R^2	0.8374	0.8785	0.9000
R	0.9151	0.9373	0.9487
No. of Layers	2	3	4
Neurons for Layer 1	6	10	11
Neurons for Layer 2	18	12	23
Neurons for Layer 3	—	1	29
Neurons for Layer 4	—	—	2
Function for Layer 1	`purelin`	`softmax`	`softmax`
Function for Layer 2	`logsig`	`radbasn`	`tribas`
Function for Layer 3	—	`purelin`	`radbas`
Function for Layer 4	—	—	`softmax`
Function for Output Layer	`purelin`	`purelin`	`satlins`
Training Algorithm	`trainlm`	`trainlm`	`trainlm`
Activated Input Variables	$\{t, \check{T}_{\mathrm{dp}}\}$	All	$\{t, \check{T}, \check{T}_{\mathrm{dp}}, \omega_s\}$
Epochs Used	21 out of 1000	37 out of 1000	42 out of 1000

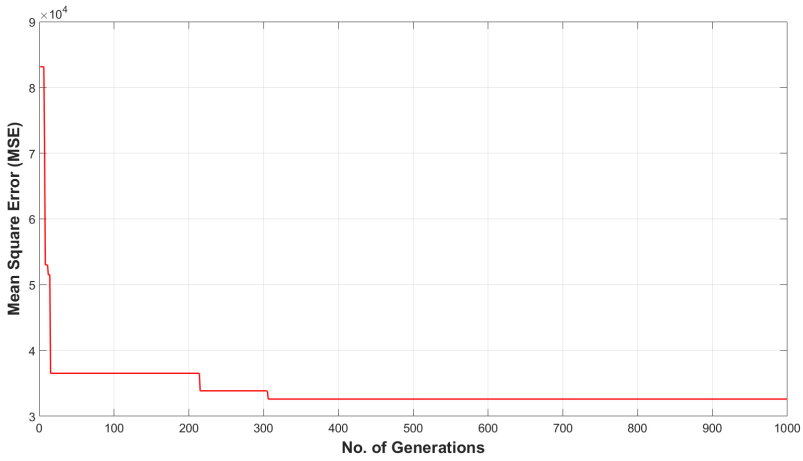
Figure 5.8 shows the fitness obtained for each heat profile. These three fitness plots are the MSE performance of the neural networks measured during optimizing their configurations. From these plots, it is obvious that the performance of ANNs can be effectively enhanced by optimizing their hyperparameters. Doing that manually (i.e., through what it is called the "trial and error" technique) is a daunting task and does not guarantee to reach the optimal value. Instead, meta-heuristic optimization algorithms can be employed to do that task easily and smoothly. It can be seen that the optimal configuration of each ANN is quickly obtained before completing all the generations. These optimal configurations, obtained by BBO, are tabulated in Table 5.4 where MAE, R^2, and R are also listed. From the preceding table, it is obvious that the residential non-electric heat does not need to know the temperature to forecast the actual heat energy. The graphical comparisons between the actual and forecasted heat energy profiles are shown in Figure 5.9 for all the three ANNs.

Although MSE, MAE, R, and R^2 all are shown in Table 5.4, only MSE is taken as the objective function of BBO. Thus, the main purpose of BBO is to minimize MSE during optimizing the hyperparameters of ANNs. Now, let's select R^2 as the objective function[19] of BBO. Table 5.5 shows the new optimal configurations obtained

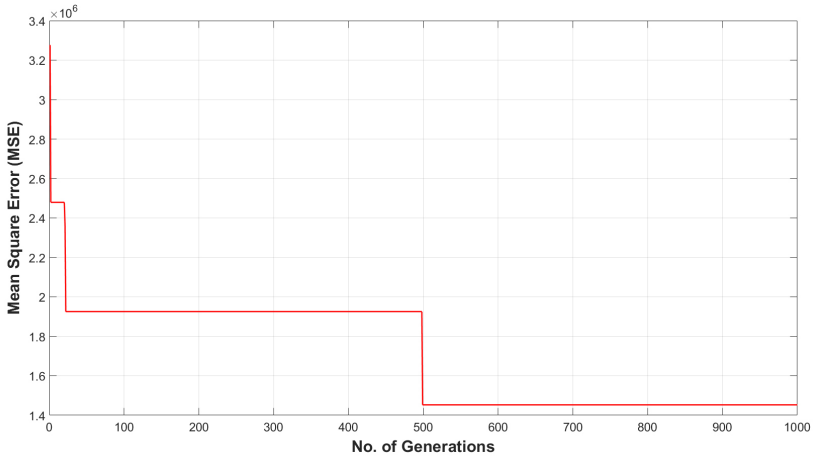
[19]Or even thinking about it as a multi-objective problem (MOP) where both MSE and R^2 are



(a) Fitness of the Residential Non-Electric Heat Profile



(b) Fitness of the Residential Electric Heat Profile



(c) Fitness of the Municipal Heat Profile

Figure 5.8: Fitness Curves of the Three Heat Profiles (Objective Function: MSE)

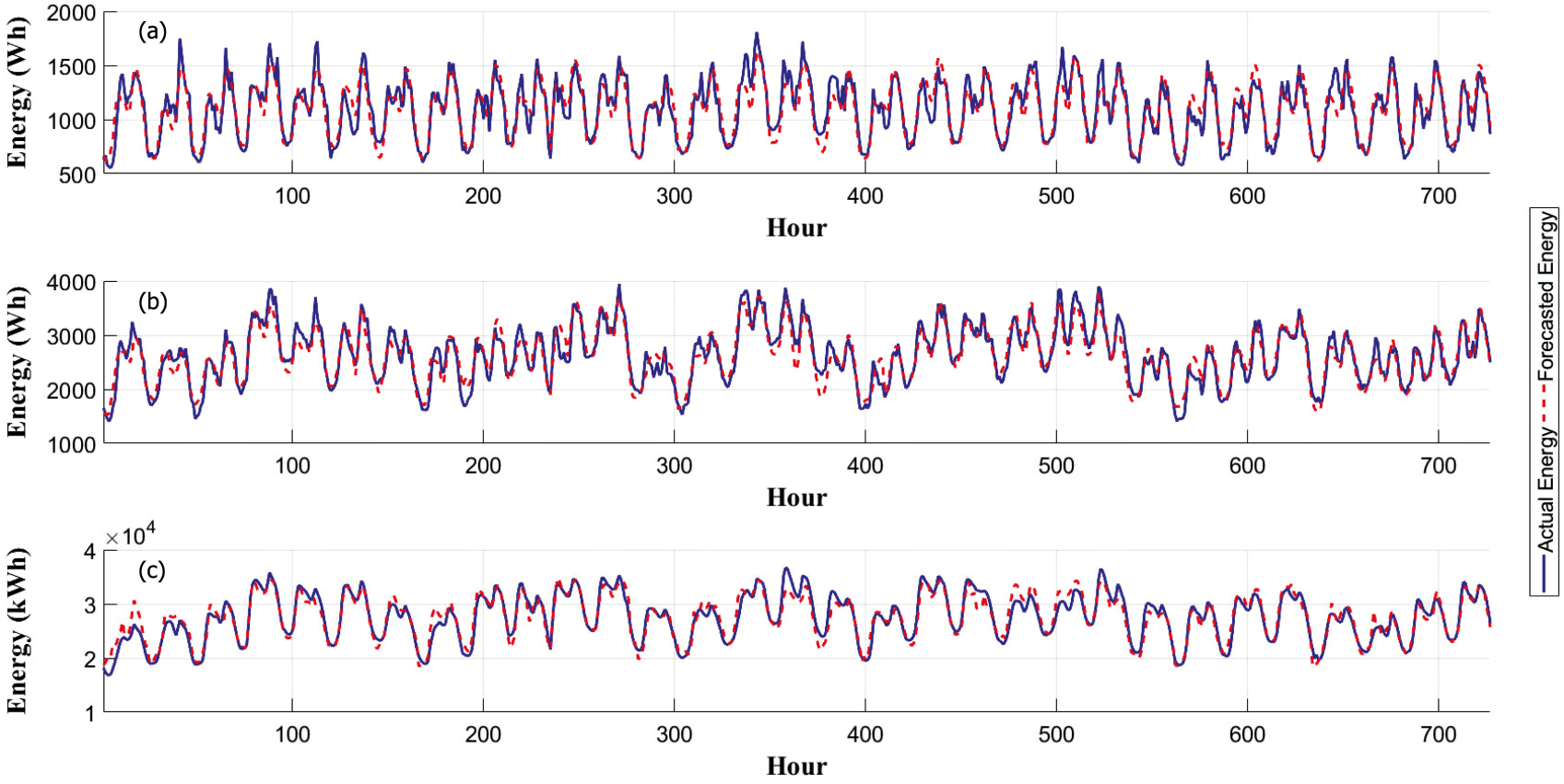


Figure 5.9: Graphical Comparisons Between the Actual and Forecasted Heat Energy Profiles (Objective function: MSE). (a) Residential Non-Electric Heat, (b) Residential Electric Heat, and (c) Municipal Heat

Table 5.5: Optimal Hyperparameters of the Three ANNs (Objective Function: 1-R^2)

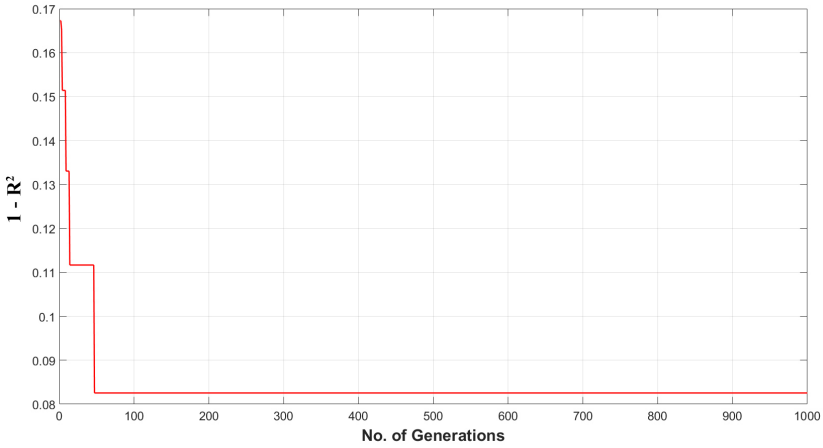
Performance Criteria	Residential Non-Electric Heat	Residential Electric Heat	Municipal Load
MSE	16936.5566	70556.7453	2432829.2816
MAE	101.8129	188.9996	1133.7686
R^2	0.9178	0.9040	0.9245
R	0.9580	0.9508	0.9615
No. of Layers	2	2	1
Neurons for Layer 1	30	22	16
Neurons for Layer 2	15	15	—
Neurons for Layer 3	—	—	—
Neurons for Layer 4	—	—	—
Function for Layer 1	`radbas`	`radbas`	`tansig`
Function for Layer 2	`logsig`	`satlin`	—
Function for Layer 3	—	—	—
Function for Layer 4	—	—	—
Function for Output Layer	`satlins`	`satlins`	`satlins`
Training Algorithm	`trainlm`	`trainlm`	`trainlm`
Activated Input Variables	All	All	All
Epochs Used	20 out of 1000	29 out of 1000	32 out of 1000

when (1-R^2) is selected as the objective function. The reason for using (1-R^2) instead of R^2 is that the new objective function maximizes R^2, while the previous objective function minimizes MSE. Thus, to avoid re-coding the whole BBO algorithm, it is easier to convert the objective function itself from being in the maximization mode to a minimization mode[20]. Based on this, for the second modified objective function (i.e., after applying the duality), the fitness curves obtained for these three heat profiles are shown in Figure 5.10.

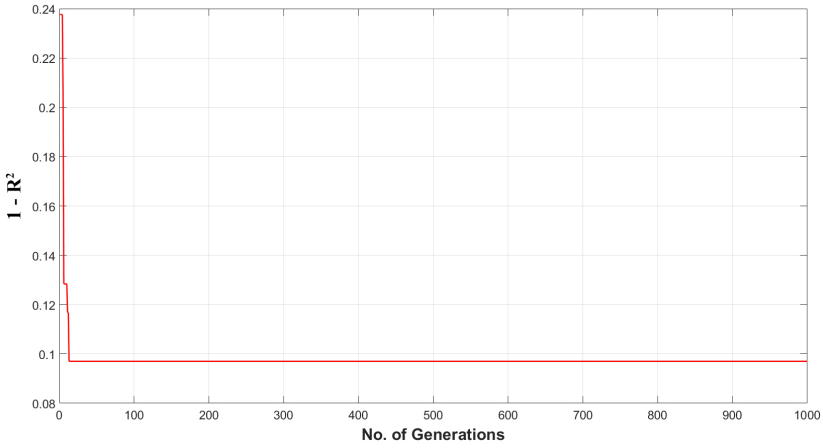
From Table 5.5, it is obvious that all ANNs require the complete set of input variables or features to precisely forecast the three heat energy profiles. Also, for both approaches given in Tables 5.4-5.5, the best gradient-based algorithm used for the training phase is `trainlm`. This can be observed in the optimal configurations obtained for all the six ANNs (i.e., the three ANNs optimized based on "MSE", and the three ANNs optimized based on "1-R^2"). The graphical comparisons between the actual and forecasted heat energy profiles are shown in Figure 5.11 for all the three ANNs when the objective function is 1-R^2.

considered during minimizing the neural network configurations.

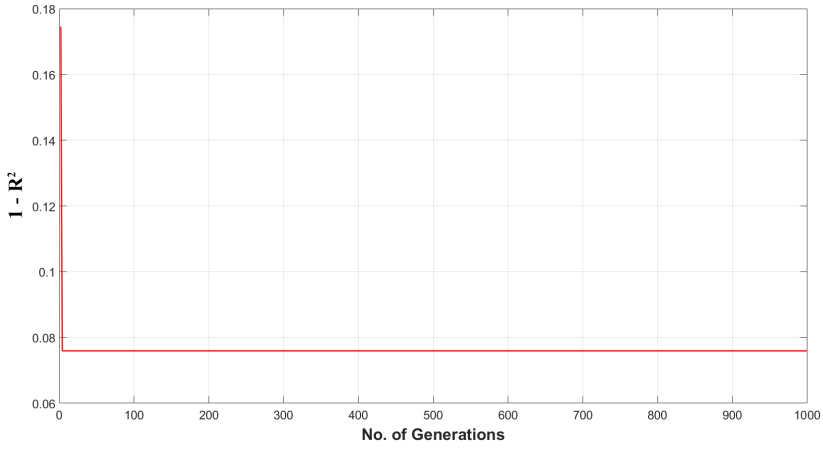
[20]The other duality approach can be done by changing the sign of R^2. This approach was described in Chapter 2.



(a) Fitness of the Residential Non-Electric Heat Profile



(b) Fitness of the Residential Electric Heat Profile



(c) Fitness of the Municipal Heat Profile

Figure 5.10: Fitness Curves of the Three Heat Profiles (Objective Function: 1-R^2)

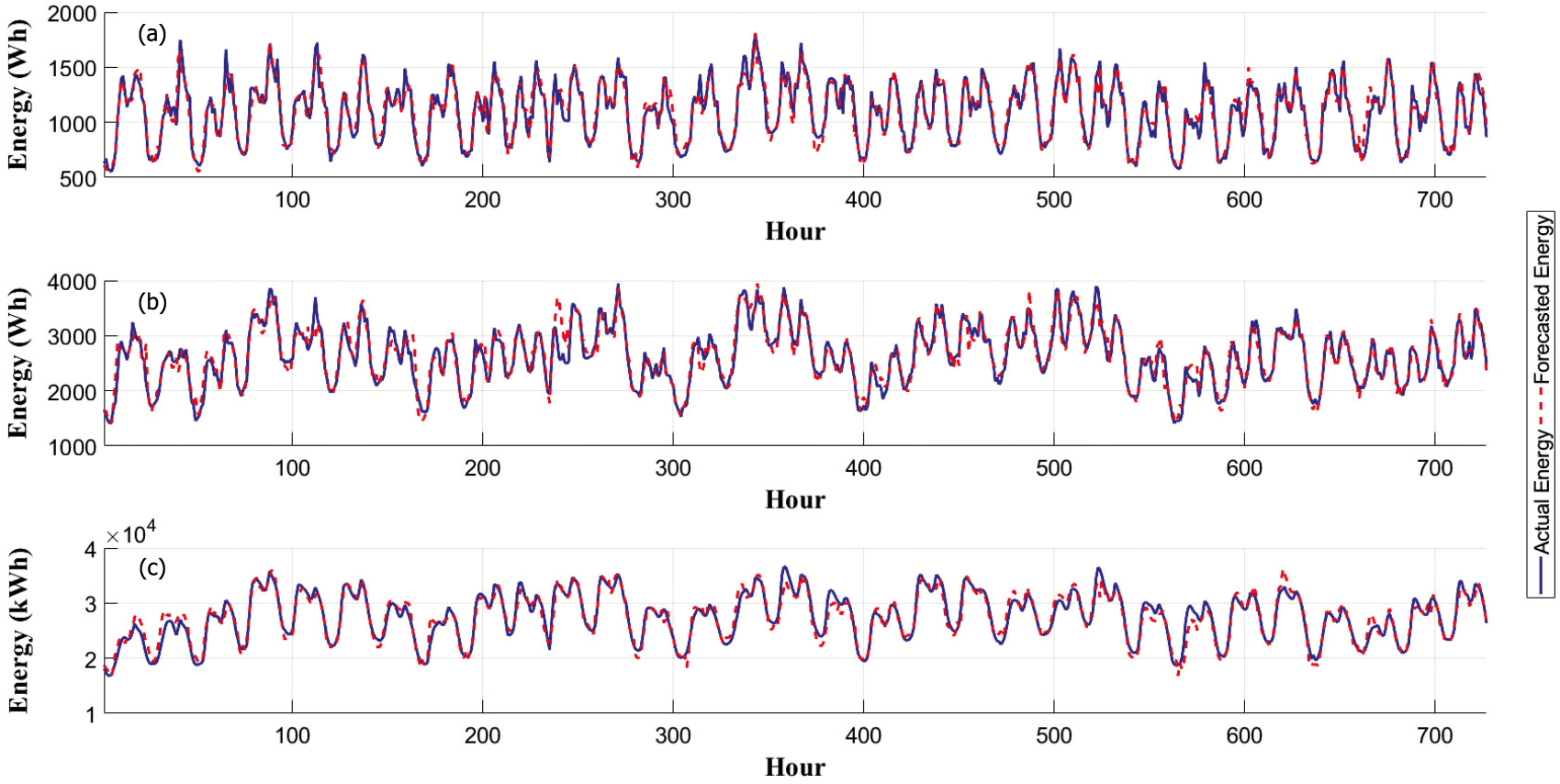


Figure 5.11: Graphical Comparisons Between the Actual and Forecasted Heat Energy Profiles (Objective function: 1-R^2). (a) Residential Non-Electric Heat, (b) Residential Electric Heat, and (c) Municipal Heat

Before ending this section, it is interested to see the graphical representation of the optimal feed-forward neural network topologies listed in Tables 5.4-5.5 for MSE and (1-R^2), respectively. The comparisons of these optimally configured networks are illustrated, by two different diagrams, in Figures 5.12-5.14 for the residential non-electric heat, residential electric heat, and municipal heat profiles, respectively.

5.5 Numerical Experiments on Optimally Configured SVRs to Forecast the Peak-Load of Nova Scotia During Winter 2018-2019

As said before, EAs and other hybrid optimization algorithms can efficiently optimize the SVR hyperparameters. However, the basic random search algorithm (RSA) described in Algorithm 10 is selected. The reason is to examine its performance and also because RSA is very simple to code. In this experiment, six stages are used for RSA with a total number of generations of $G_{\text{RSA}} = 96000$. Thus, by applying (5.11), the total number of generations or iterations assigned to each stage are:

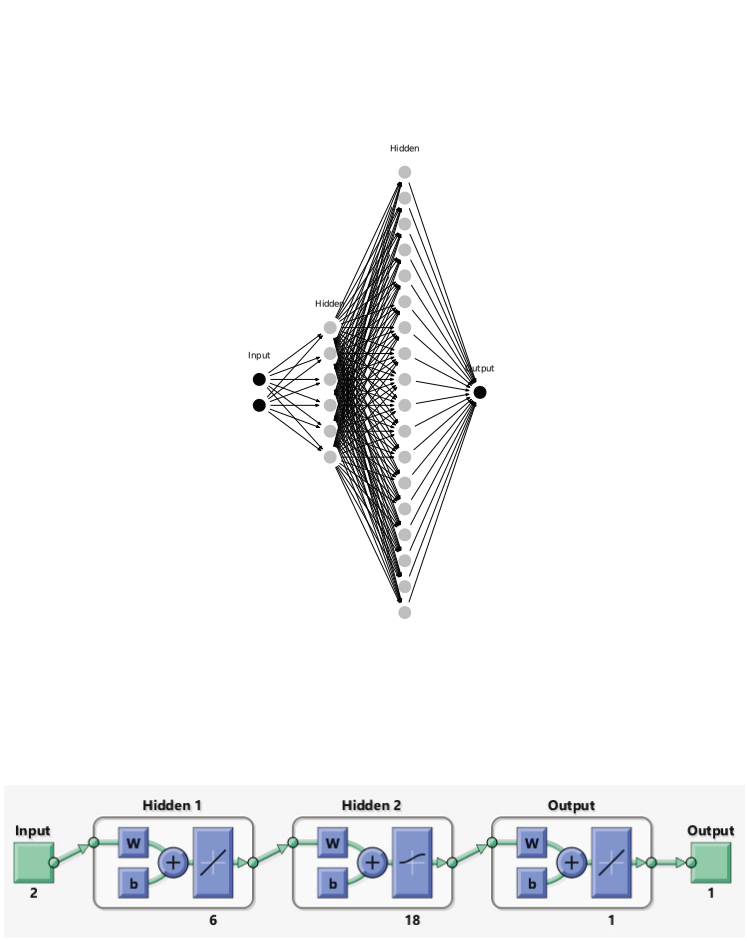
$$\chi = \{48762, 24381, 12190, 6095, 3048, 1524\}$$

where $G_{\text{RSA}} = \sum_{j=1}^{\varsigma} \chi_j$; refer to Appendix C.

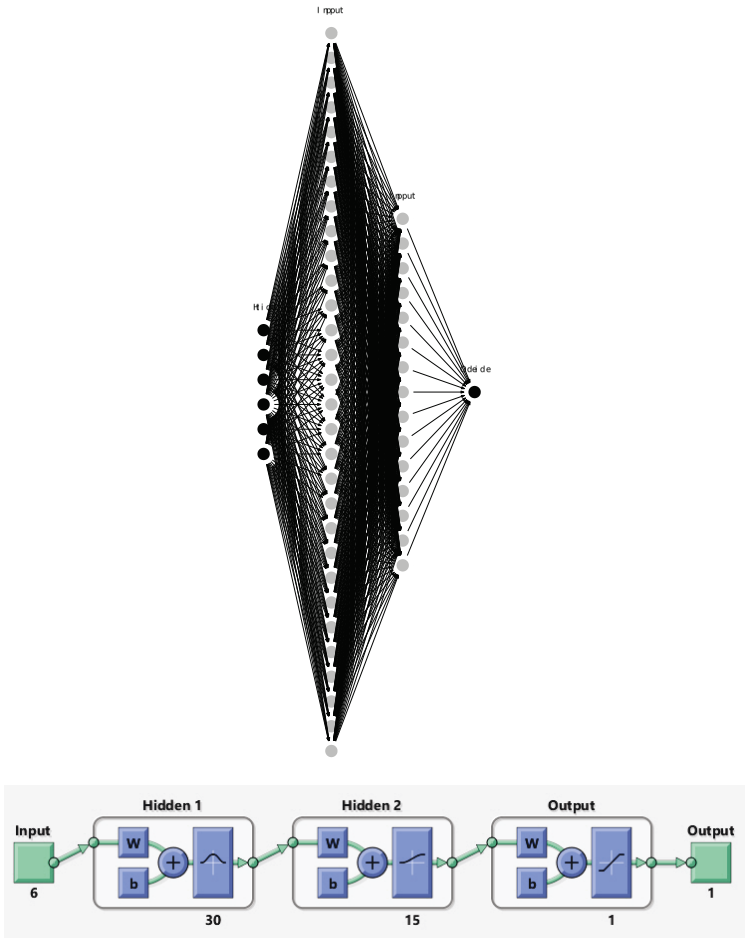
This experiment represents a long-term forecast for the hourly power demand consumed across Nova Scotia during the peak-load of 2018-2019. Since Canada is a cold country, so the peak-load is supposed to be during the winter season[21]. To prove that, a big real dataset consists of 102854 observations is used, which is provided by Emera's Nova Scotia Power [286]. The length of this hourly sampled power demand is twelve years and five months, which covers the period from the 1st of January 2007 to the 31st of May 2019. Figure 5.15 shows the plot of these observations. To indicate the peak-load season in the plot, each year is delimited by a vertical line. It can be clearly observed from the pattern shown in the plot that the power demand is almost stable over the past twelve years and five months[22]. Also, the annual cycles show that the higher demand levels occur during the winter season and vice versa during the summer season.

The task here is to forecast the winter of 2018-2019, which covers the period from the 1st of December 2018 to the 28th of February 2019. This portion of the

[21] In hot countries, like middle east countries, the peak-loads happen during the summer season.
[22] This can also be used as a measure to indicate the population growth of Nova Scotia, Canada.

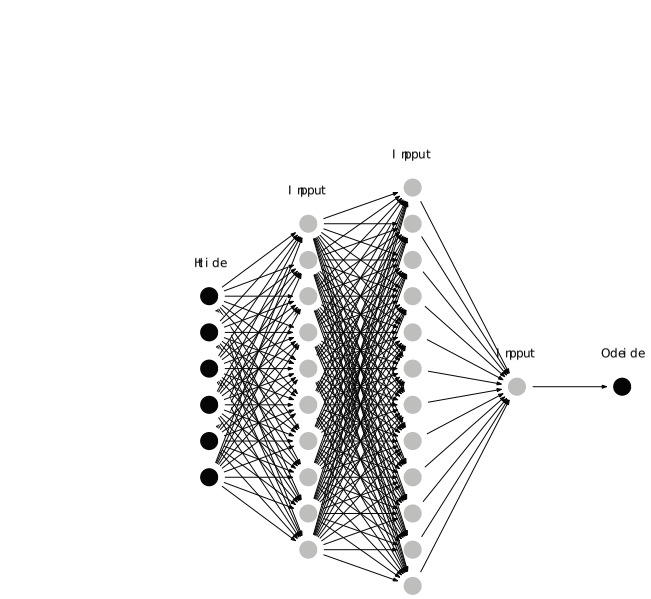


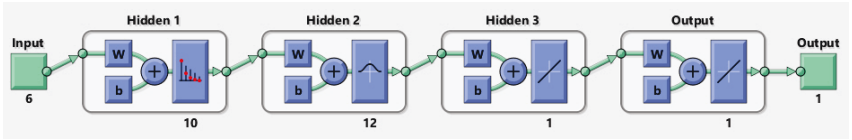
(a) BBO Obj: MSE



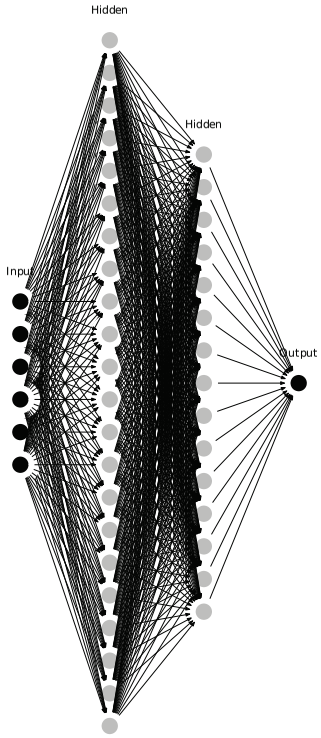
(b) BBO Obj: 1-R^2

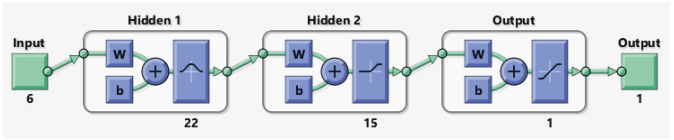
Figure 5.12: Optimal Topologies Obtained for the Residential Non-Electric Heat Profile





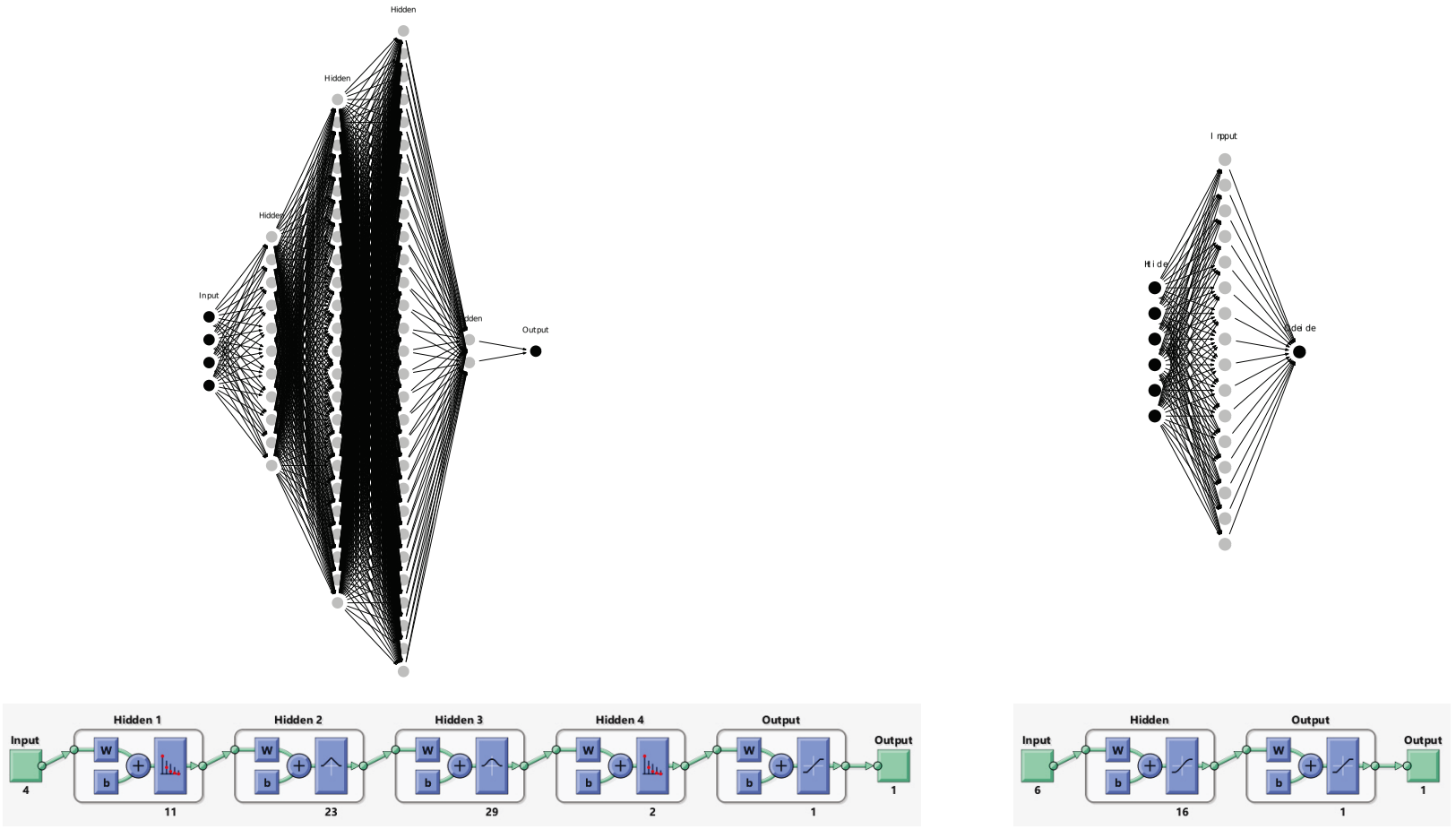
(a) BBO Obj: MSE





(b) BBO Obj: 1-R^2

Figure 5.13: Optimal Topologies Obtained for the Residential Electric Heat Profile



(a) BBO Obj: MSE

(b) BBO Obj: 1-R^2

Figure 5.14: Optimal Topologies Obtained for the Municipal Heat Profile

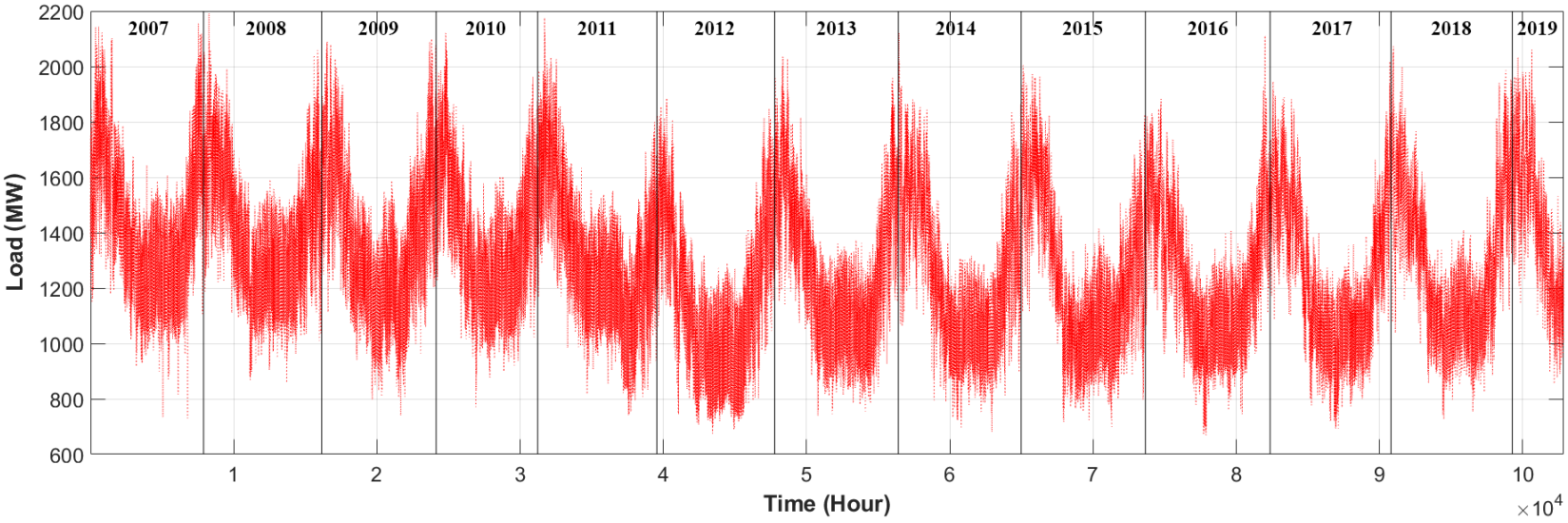


Figure 5.15: Actual Power Demand Data of the Entire Load of Nova Scotia (Hourly Recorded from the 1st of January 2007 to the 31st of May 2019)

dataset consists of 1557 observations where some hours are filtrated out because of some unavailable metrology measurements. These measurements are taken from the "**Halifax Windsor Park Weather Station**[23]", which can be accessed via: Government of Canada > Environment and Natural Resources > Weather, Climate, and Hazard webpage [381]. The measurements of time (in hours t_h, in days t_d, and in months t_m), metrology (temperature $\breve{T}$, dew point temperature $\breve{T}_{\text{dp}}$, relative humidity RH, wind direction ω_d, wind speed ω_s, station pressure $\mathcal{P}_s$, and wind chill $\breve{T}_{\omega}$), and active power P all are plotted in Figure 5.16.

For this experiment, the scikit-learn library, given in [304], is coded in Python 3.5 using PyCharm 2019.1.3 IDE with the same computing machine used in the last section. The results, obtained by executing Algorithm 10 with $\varsigma = 6$ and $G_{\text{RSA}} = 96000$, are tabulated in Table 5.6. Although MSE of the train dataset could go below 1 if the test dataset is neglected, this approach should not be taken to avoid having an **overfitted** response. Thus, the focus here is to reduce MSE of the test dataset. This can be observed by tracing MSE from one stage to another. Although RSA is considered as a very simple optimizer, the results show that MSE can be reduced from 21539.6596 down to just 629.5284, and the score can be increased from 0.002502 up to 0.9717, which is impressive. The MSE performance of the train and test datasets are depicted in Figure 5.17 for the default SVR hyperparameters and in Figure 5.18 for the final tuned hyperparameters.

Again, the SVR performance could be further enhanced by employing some modified EAs as discussed before in Subsection 5.3.2. This could be considered in future work. However, the main goal of this experiment is to prove that even basic optimization algorithms, like RSA, can find very good results if they are coded correctly.

[23]In the previous forecasting problem solved by ANNs, the metrology measurements were taken from the Halifax Dockyard Weather Station. The Halifax Windsor Park Weather Station is used in this problem because its archive contains more metrology data than the previous one.

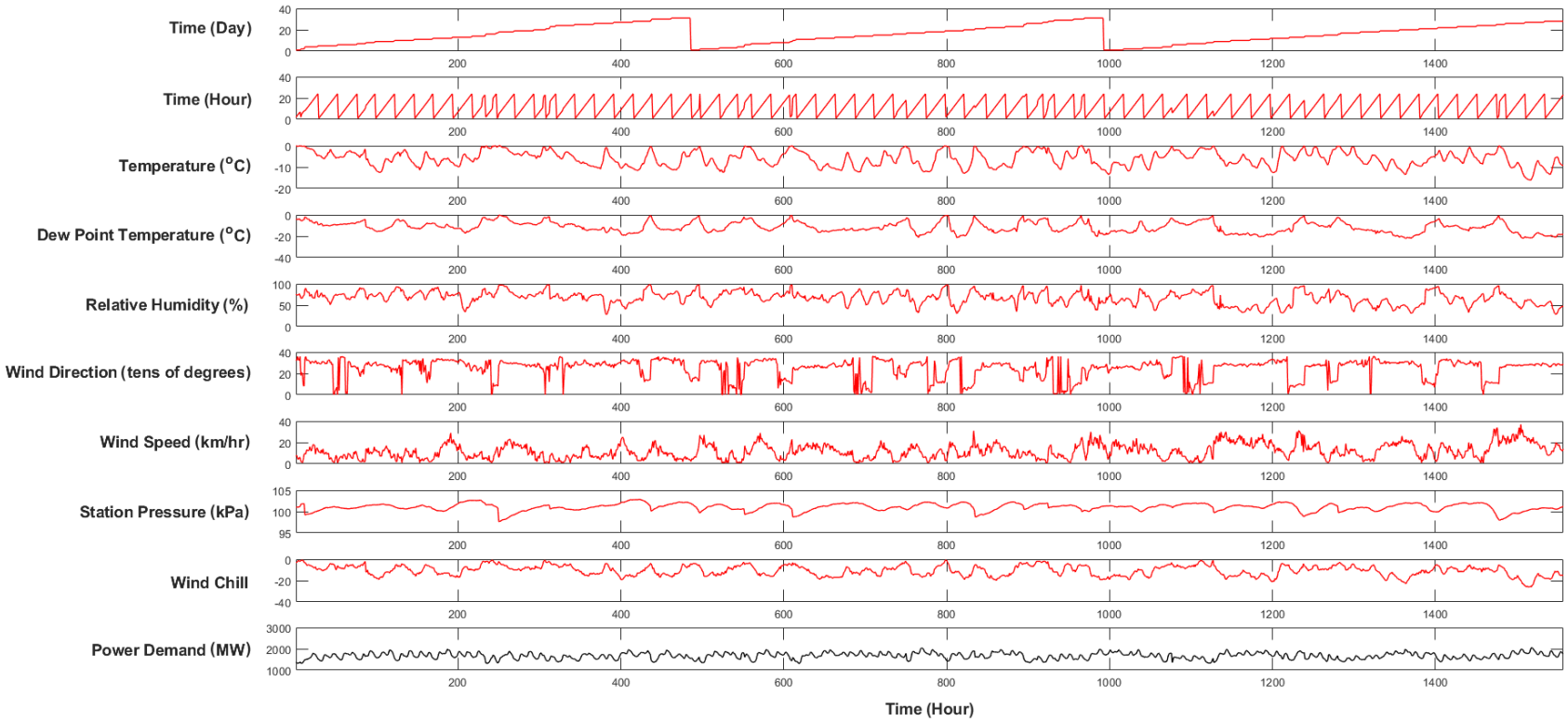


Figure 5.16: Time, Metrology, and Power Data of Nova Scotia Measured During the Last Winter (Hourly Recorded from the 1st of December 2018 to the 28th of February 2019)

Table 5.6: Optimal Hyperparameters of SVR Obtained by RSA with Six Stages (Objective Function: MSE)

Optimal Results	**Default SVR Settings**[a]	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**	**Stage 5**	**Stage 6**
G_{RSA}	—	48762	24381	12190	6095	3048	1524
F_{SVR}	—	all kernels	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'
$d^{\min}$	—	1	—	—	—	—	—
$d^{\max}$	—	9	—	—	—	—	—
$\gamma^{\min}$	—	0.001	0.001	0.01	0.01	0.05	0.1
$\gamma^{\max}$	—	100	1	0.5	0.3	0.2	0.16
$\text{coef0}^{\min}$	—	-1000	—	—	—	—	—
$\text{coef0}^{\max}$	—	1000	—	—	—	—	—
$r_C^{\min}$	—	0.001	10000	50000	50000	60000	71000
$r_C^{\max}$	—	100000	100000	90000	80000	80000	79000
$\varepsilon^{\min}$	—	0.001	1	1	2	2	2.4
$\varepsilon^{\max}$	—	10	5	5	4	3	2.8
shrk	—	{True, False}	{True, False}	{True, False}	{True, False}	{True, False}	{True, False}
U	—	all predictors	all predictors	all predictors	all predictors	all predictors	all predictors
F_{SVR}^*	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'
d^*	—	—	—	—	—	—	—
γ^*	'auto'	0.05416	0.05827	0.1221	0.1105	0.1414	0.1206
coef0^*	—	—	—	—	—	—	—
r_C^*	1	74644.1619	71666.9559	56100.7105	62292.9281	71658.4812	77615.6699
ε^*	0.1	3.0369	1.1078	3.4887	2.5313	2.9526	2.6868
shrk^*	True	False	False	True	True	True	True
U^*	all predictors	$\{t_d, t_h, \breve{T}, \mathcal{P}_s, t_m\}$	$\{\breve{T}, \breve{T}_{\text{dp}}, t_d, \mathcal{P}_s, t_h, t_m\}$	$\{\mathcal{P}_s, t_d, t_h, \breve{T}_{\text{dp}}, t_m\}$	$\{t_m, \breve{T}, t_h, t_d, \mathcal{P}_s\}$	$\{\breve{T}, t_m, t_d, t_h, \mathcal{P}_s\}$	$\{\breve{T}, t_h, t_m, \mathcal{P}_s, t_d\}$
Train MSE	21758.0957	232.4796	42.2837	19.814	31.6823	14.6085	23.3358
Test MSE	21539.6596	1425.0731	1184.9843	1106.0034	902.289	855.5508	629.5284
Train Score	0.006785	0.9894	0.9981	0.9991	0.9986	0.9993	0.9989
Test Score	0.002502	0.9319	0.946	0.9458	0.9567	0.9579	0.9717

[a]The solution depends on the train and test datasets, which are varied at each run.

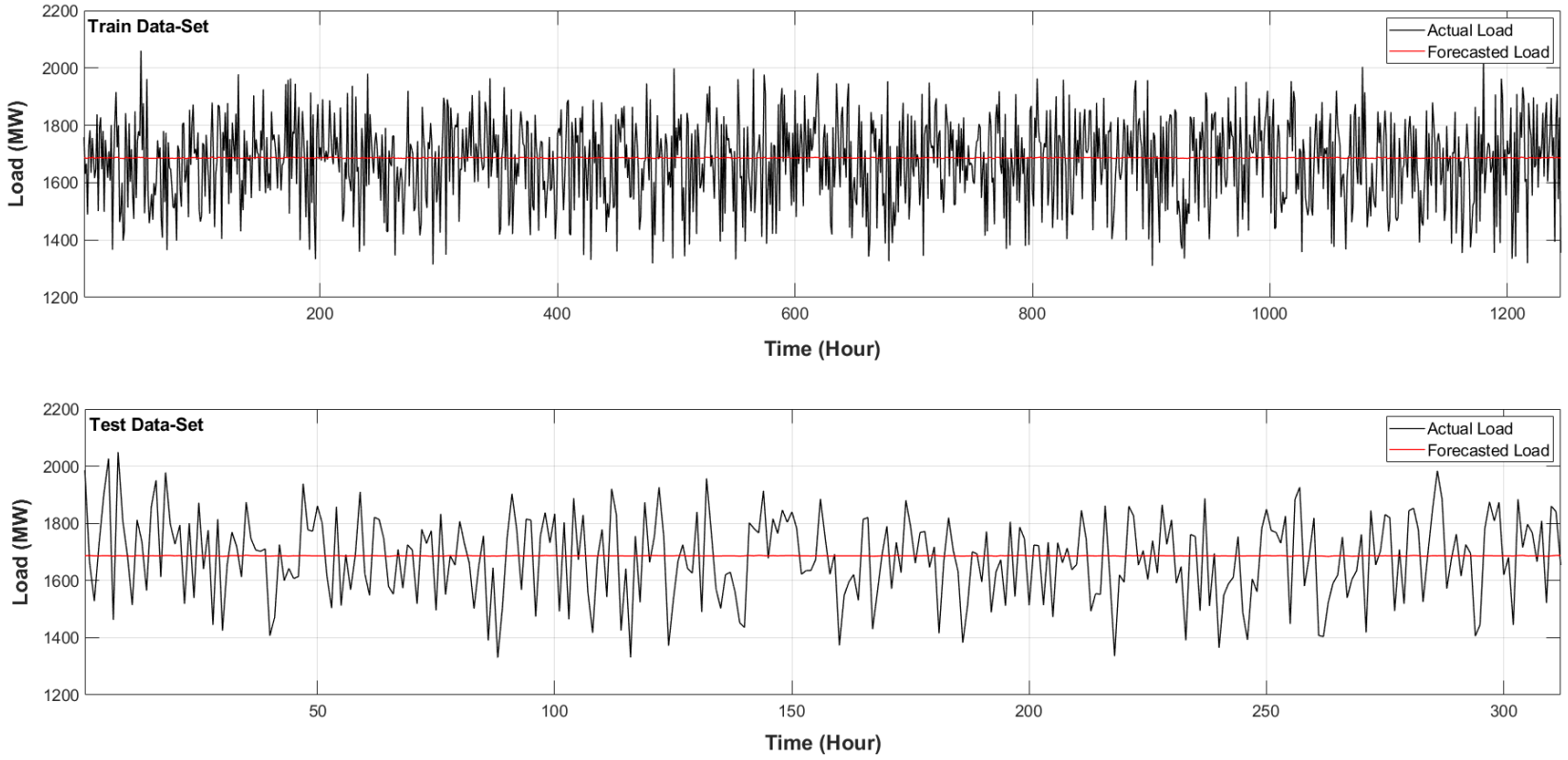


Figure 5.17: Graphical Comparisons Between the Actual and Forecasted Active Load (Objective Function: MSE — Default SVR Hyperparameters)

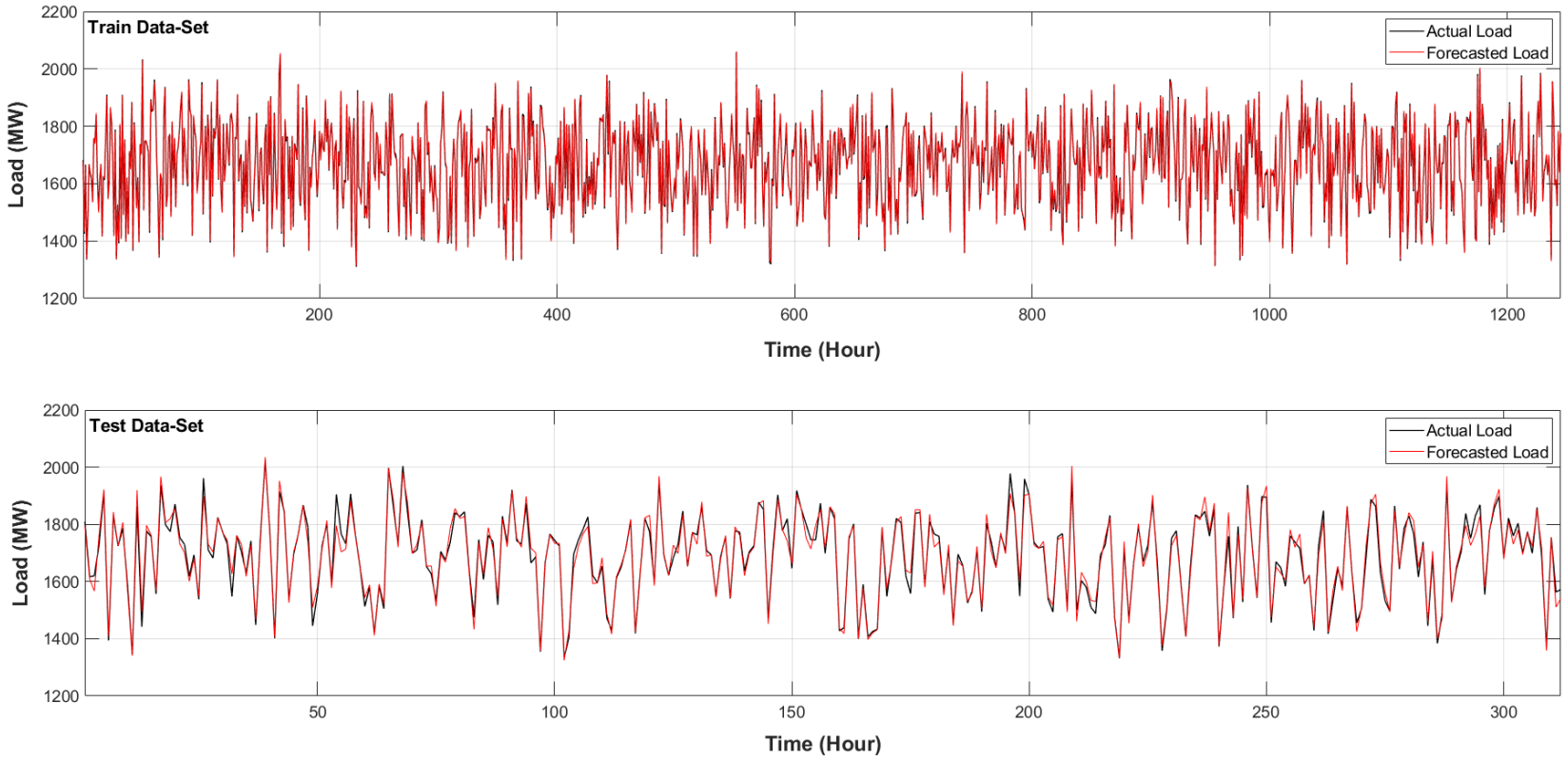


Figure 5.18: Graphical Comparisons Between the Actual and Forecasted Active Load (Objective Function: MSE — Optimal SVR Hyperparameters)

Chapter 6

Realizing Transmission Line Models

If someone opens any popular electric power engineering textbook, he/she will realize that the most important stage is PF analysis [125, 149, 155, 326]. The reason is that all the other stages[1] mainly depend on data received from the PF analysis. Therefore, if that data is incorrect or inaccurate, then the whole process will be affected! By looking at Figure 1.1, it is obvious that the PF analysis itself depends on the quality and accuracy of the model used to represent real transmission lines. Thus, even going with highly precise PF solvers, weak representation of transmission lines leads to significant errors in all power system analysis.

In the **circuit theory**, some approximations are applied to simplify Maxwell's equations of electromagnetism[2]. To represent real transmission lines as mathematical equations, there are two possible representations:

1. The **lumped parameter** model[3].
2. The **distributed parameter** model.

The aim of this chapter is to enhance the models used in representing transmission lines. First, it presents the conventional lumped parameter models as an introduction to the subject. Then, a new more realized lumped parameter model is presented, and its performance is evaluated against all the known lumped parameter models. After that, the conventional steady-state distributed parameter model is revisited to integrate many phenomena, such as the variations in the surrounding weather conditions and the fluctuations in the system frequency. Finally, some leading topics are introduced as a result of this work.

[1] Such as economic load dispatch, unit commitment, optimal power flow, fault analysis, contingency analysis, stability and control, **state estimation** (**SE**), relay coordination, etc.

[2] In the period between 1884-1885, Heaviside successfully re-wrote Maxwell's 20 fundamental equations to obtain a new set of four compact equations, which were standardized by the late 1890s [174].

[3] It is also called the **lumped component** model and the **lumped element** model.

6.1 Conventional Lumped Parameter Models

For any two-terminal circuit, the lumped parameter model proposes that the current passes through all the passive components (i.e., resistance, inductance, and capacitance) instantaneously without any transient action. This assumption contradicts the fact that the current change propagates with a limited speed based on the nature of the circuit. Based on this, the lumped parameter models are simplified circuits, which are exclusively used to represent short and medium transmission lines.

6.1.1 Short-Length Transmission Line Model

This is the first and simplest transmission line model. Any power transmission line with a length of 80 km (∼50 miles) or less can be simplified as a short line model where the effects of the per-phase capacitance C and conductance G are neglected. The simplified circuit of this model is shown in Figure 6.1. The derivation of this model is given in Appendix D. From that derivation, the following two equations are obtained:

$$V_S = V_R + ZI_R \tag{6.1}$$

$$I_S = I_R \tag{6.2}$$

In electric power systems engineering, the **ABCD** or the **two-port network** representation, shown in Figure 6.2, is preferred. It is very useful and practical for many simulations. Thus, (6.1) and (6.2) can be formulated as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.3}$$

where, by analogy, $A = 1$, $B = Z$, $C = 0$, and $D = A$.

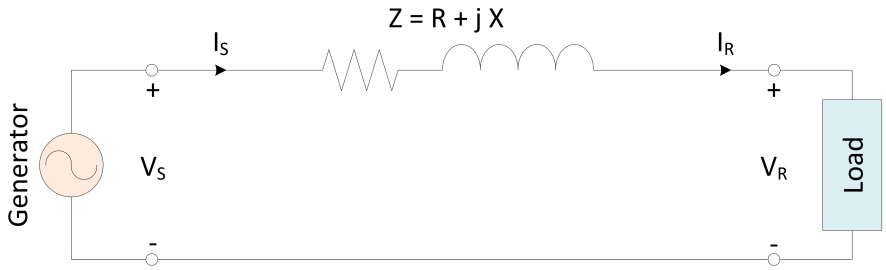


Figure 6.1: Short-Length Transmission Line Model

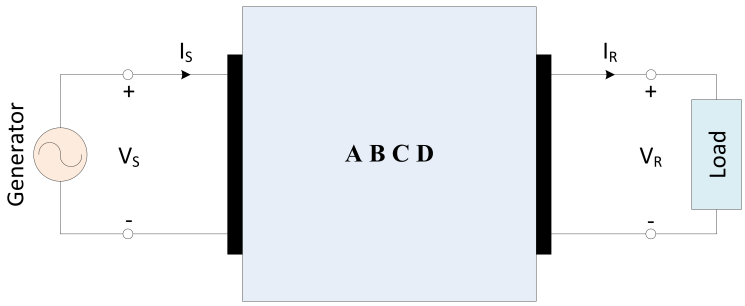


Figure 6.2: Two-Port Network Representation of a Transmission Line

6.1.2 Medium-Length Transmission Line Models

In fact, four popular models are currently used when the total length exceeds 80 km (∼50 miles) but not longer than 250 km (∼155 miles) [326]. These medium line models have less relaxation where the line charging current due to the shunt capacitance cannot be ignored anymore. Thus, for any of these four models, the total shunt admittance of the line can be calculated as follows:

$$Y = (\not{G} + j\omega C)\ \mathfrak{L} \tag{6.4}$$

where G is the **per-unit-length** shunt conductance and it is neglected under normal condition[4].

6.1.2.1 Γ-Model

This model is one of the simplest medium-length transmission line models. It is also called **L-Shape** [123], which is shown in Figure 6.3. As can be clearly seen, the lumped shunt admittance is placed at the sending terminal of the transmission line. The full derivation of this model is given in Appendix E. The two-port network of this model is mathematically expressed as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ Y & 1+ZY \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.5}$$

[4] i.e., the **corona** phenomenon and the leakage current over the insulators are ignored [326].

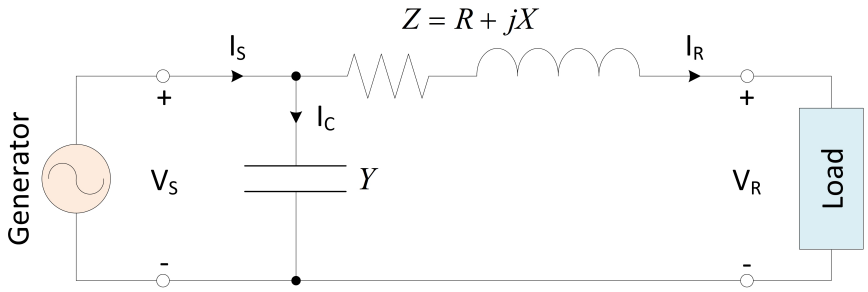


Figure 6.3: Medium-Length Transmission Line Model (Nominal Γ-Circuit)

6.1.2.2 Ɪ-Model

Also, this **Opposite-Gamma** representation is one of the simplest medium-length transmission line models. It is similar to the **Gamma** or L-Shape model except that the lumped shunt admittance is placed at the receiving terminal of the transmission line. The single-line diagram of this model is shown in Figure 6.4, and the full derivation is given in Appendix F. The two-port network of this model is:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+ZY & Z \\ Y & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.6}$$

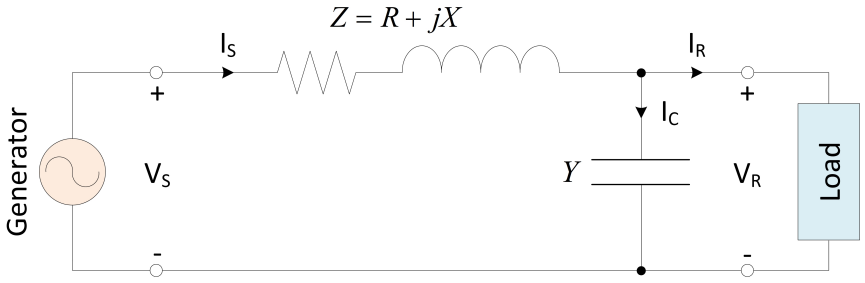


Figure 6.4: Medium-Length Transmission Line Model (Nominal Ɪ-Circuit)

6.1.2.3 T-Model

This is the second popular medium line model. To build this model, the lumped shunt admittance is placed in the middle of the transmission line and the lumped series impedance is divided into two equal parts. The first part is placed between the first end (or the sending terminal) and the shunt admittance, and the second

part is placed between the shunt admittance and the second end (or the receiving terminal) of the transmission line as shown in Figure 6.5. The full derivation is given in Appendix G. The two-port network of this model is:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & Z\left(1+\frac{ZY}{4}\right) \\ Y & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.7}$$

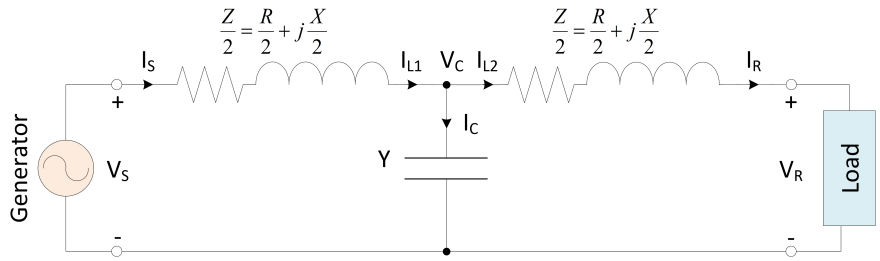


Figure 6.5: Medium-Length Transmission Line Model (Nominal T-Circuit)

6.1.2.4 Π-Model

This is the most popular medium line model. In this model, the lumped series impedance is placed in the middle of the transmission line and the lumped shunt admittance is divided into two equal parts and placed at the two ends (or the sending and receiving terminals) of the transmission line. The single-line diagram of this model is shown in Figure 6.6, and the full derivation is given in Appendix H. The two-port network of this model is:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & Z \\ Y\left(1+\frac{ZY}{4}\right) & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.8}$$

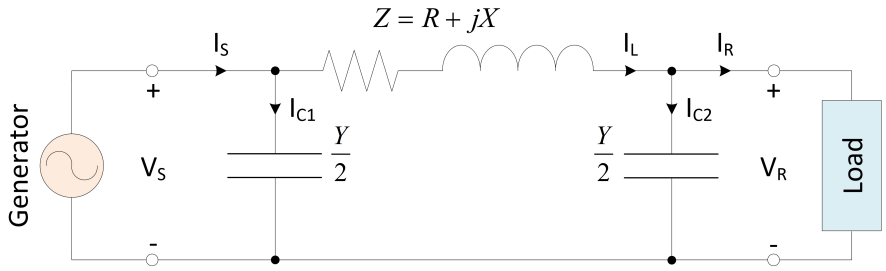


Figure 6.6: Medium-Length Transmission Line Model (Nominal Π-Circuit)

6.1.3 Long-Length Transmission Line Model

The distributed parameter model is used when the length is longer than 250 km (∼155 miles) [326]. In this model, all the per-unit-length passive components are considered to have very accurate calculations. That is, the **transmission line theory**[5] can be considered as a correction or modification to the classical circuit theory, which is required to account the effects of per-unit-length capacitance and conductance [49,64].

Let's consider Figure 6.7, which represents a single-line diagram of a per-unit-length section of a long transmission line with distributed parameters. It can be noticed that the incremented voltage $v(x+\Delta x,t)$ and current $i(x+\Delta x,t)$ are relocated to be on the left side, while $v(x,t)$ and $i(x,t)$ are on the right side! Actually, these arrangements just change the signs of some equations to explain the direction of source voltage and current and answer whether they are received on that terminal or sent. This adjustment is useful when deriving the two-port network of this model, which is given in Appendix J. Based on this derivation, the model can be mathematically expressed as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \cosh(\gamma\mathfrak{L}) & Z_c \sinh(\gamma\mathfrak{L}) \\ \frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.9}$$

where Z_c is called the *characteristic impedance*, which is given in (J.65).

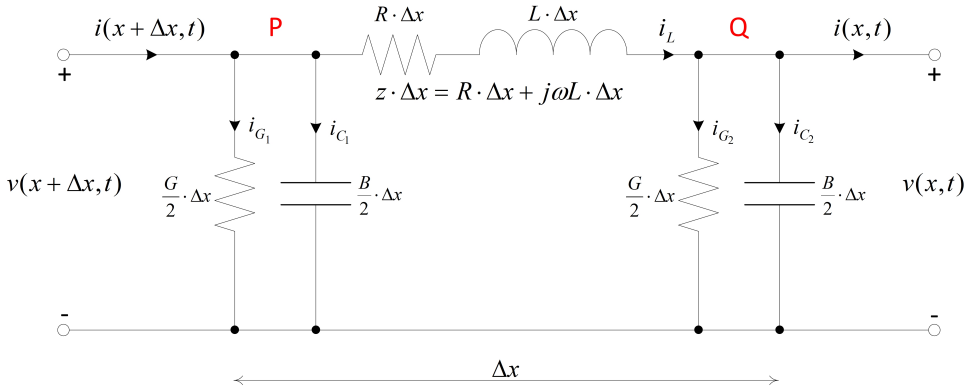


Figure 6.7: Transmission Line Section of Length Δx (Long-Line Model)

[5]The transmission line theory (which is well-known as the **Telegrapher's equations**, and sometimes called the **telegraph equations**) were developed in the horse and buggy days of the telegraph in the 1880s by Oliver Heaviside [268]. These two equations are respectively used to analyze the behavior of voltage v and current i across and along a transmission line as functions of time t and distance x.

6.2 M-Model: A New Precise Medium-Length Transmission Line Model

In the preceding section, the known transmission line models have been introduced. This section presents a new transmission line model[6]. The nominal single-line diagram of this model is shown in Figure 6.8. We call it the M-model because the lumped parameters are arranged in an M-shape. To validate its superiority, some numerical experiments are given to test its performance against the short-length model and all the types of the medium-length models (i.e., Γ-, Ƭ-, T-, and Π-models).

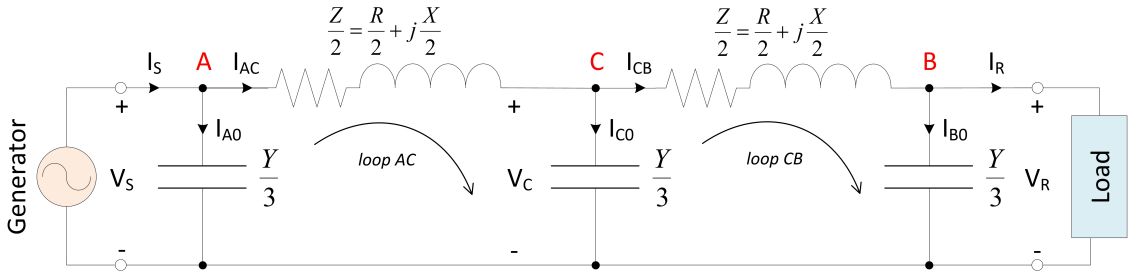


Figure 6.8: Medium-Length Transmission Line Model (Nominal M-Circuit)

6.2.1 Mathematical Formulation

The two important equations required for this model are [38]:

$$V_S = \left[1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right)\right] V_R + Z\left(1 + \frac{ZY}{12}\right) I_R \tag{6.10}$$

$$I_S = \left[Y + \frac{ZY^2}{9}\left(2 + \frac{ZY}{12}\right)\right] V_R + \left[1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right)\right] I_R \tag{6.11}$$

From these two equations, the ABCD network can be easily obtained as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right) & Z\left(1 + \frac{ZY}{12}\right) \\ Y + \frac{ZY^2}{9}\left(2 + \frac{ZY}{12}\right) & 1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{6.12}$$

The detailed derivation of this model is given in Appendix I.

6.2.2 Numerical Experiments

To evaluate the performance of our model, three different experiments are given below:

[6]Based on our knowledge with extensive search, there is no such model presented in the literature.

6.2.2.1 Testing Transmission Line Models for Different Scenarios

For conducting this experiment, a special **graphical user interface** (**GUI**) is designed using MATLAB R2017b. Figure 6.9 shows the program used in this experiment. The Γ- and Ꞁ-models will be entered into the competition later for more comprehensive performance comparison. As can be seen from GUI, the sending-end variables are given and then, from the models, the receiving-end variables are calculated. The performance of each calculated variable can be measured by considering the long line model as a reference.

For this experiment, four different scenarios are covered. Table 6.1 lists the line parameters and the sending-end variables used in each scenario. By taking the absolute difference between the lumped parameter models and the distributed parameter model, the performance of these lumped parameter models can be evaluated. Table 6.2 shows the performance of four different lumped parameter models. As can be seen, it is obvious that the M-model is the closest one to the distributed parameter model. This is correct for all or most measurements recorded in each scenario. Even for a few records, where the T-model wins, the M-model can also provide highly competitive results.

Table 6.1: Transmission Line Parameters and Variables Used in Each Scenario

Variables/	**Scenario No.**			
Parameters	**1**	**2**	**3**	**4**
V_S (kV)	345	345	220	500
I_S (A)	400	570	1000	200
R (Ω/km)	0.0173	0.036	0.15	0.045
L (mH/km)	1.0105	0.9549	1.3263	0.97
C (μF/km)	0.0117	0.125	0.0538	0.0115
G (p℧/km)	82.407	0	0	105.83
f (Hz)	60	50	60	50
$\mathfrak{L}$ (km)	478	130	40	350
pf_S	0.89	0.95	0.8	0.86
Angle Mode	lagging	lagging	leading	lagging

6.2.2.2 Testing Transmission Line Models by Varying I_S

For this part of the experiment, the line settings[7] taken from [71] are considered:

$$\mathfrak{L} = 165 \text{ km}, \quad R = 96.10 \text{ m}\Omega/\text{km}, \quad L = 1.1051 \text{ mH/km}, \quad V_S = 275\text{kV}$$
$$f = 50 \text{ Hz}, \quad C = 0.010215 \ \mu\text{F/km}, \quad G = 44.5 \text{ n}\mho/\text{km}, \quad \text{pf}_S = 0.9 \text{ (lags)}$$

The total length is calculated at the center between the minimum and maximum allowable lengths of medium transmission lines (i.e., $\mathfrak{L} = 80 + \frac{250-80}{2} = 165$ km), so it is not far away from the ranges of short and long transmission lines. To generate 2D plots, the sending-end current I_S is varied from 200 A to 1 kA. For each receiving-end variable, the solutions obtained by the lumped parameter models are subtracted from that of the distributed parameter model and then the absolute values are taken. Figure 6.10 shows the absolute error plots of these lumped parameter models. Except for V_R and REG where the T-model is the winner[8], it is clear that the M-model performs better than all the other models. For P_R and η, the competition is almost equal between the M- and T-models. Also, it can be seen that the Γ- and Ƭ-models could provide good results. However, they last just for small periods before the M-model defeats them again.

6.2.2.3 Testing Transmission Line Models by Varying Total Length

The same preceding line settings are considered in this experiment. However, here, the sending-end current I_S is set equal to 300 A with $f = 60$ Hz. The total length $\mathfrak{L}$ is varied from 10 to 1000 km. The absolute errors of the receiving-end variables are plotted in Figure 6.11 for all the lumped parameter models. Again, the M-model proves itself as a highly precise approximator of the distributed parameter model. Except for a few spots, the M-model wins in all the competitions.

6.2.3 Further Discussion

Beyond the wonderful performance of the M-model, many other points need to be covered. Some of these points are addressed below:

[7] Except the total length, power factor, and sending-end current.
[8] The variable REG will be discussed later when realizing the transmission line parameters.

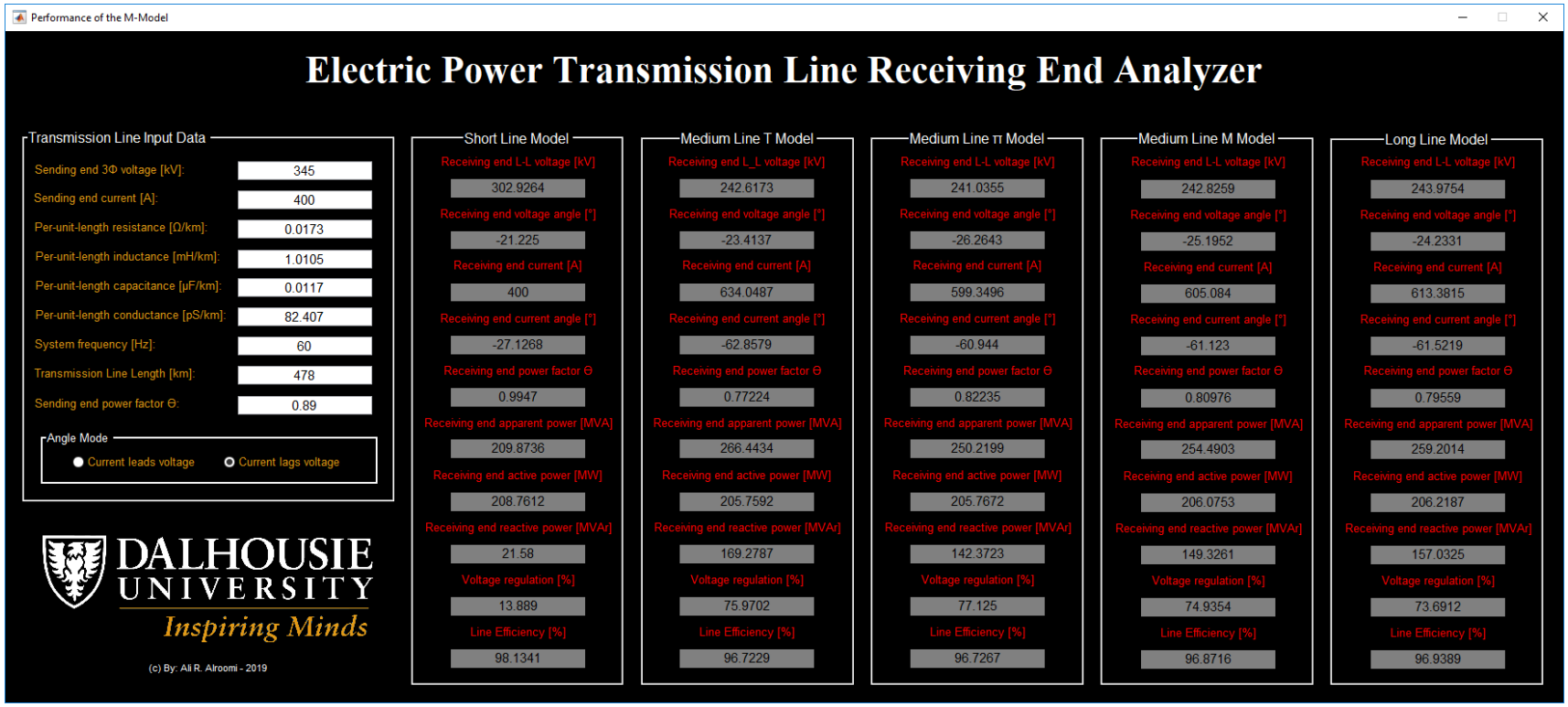


Figure 6.9: Transmission Line Receiving-End Analyzer

Table 6.2: Absolute Errors Between Some Lumped Parameter Models and the Distributed Parameter Model

Receiving-End	**Performance of Scenario No.1**			
Variables	**Short**	**T-Model**	**Π-Model**	**M-Model**
V_R (kV)	58.951	1.3581	2.9399	**1.14950**
θ_{V_R} (°)	3.008	**0.8195**	2.0311	0.96205
I_R (A)	213.380	20.6670	14.0320	**8.29750**
θ_{I_R} (°)	34.395	1.3360	0.5779	**0.39890**
pf_R	0.199	0.0233	0.0268	**0.01417**
$\lvert S_R \rvert$ (MVA)	49.328	7.2420	8.9815	**4.71110**
P_R (MW)	2.543	0.4595	0.4515	**0.14337**
Q_R (MVAr)	135.450	12.2460	14.6600	**7.70650**
REG (%)	59.802	2.2789	3.4337	**1.24420**
η (%)	1.195	0.2160	0.2122	**0.06739**
Receiving-End	**Performance of Scenario No.2**			
Variables	**Short**	**T-Model**	**Π-Model**	**M-Model**
V_R (kV)	33.457	0.2762	1.1294	**0.46440**
θ_{V_R} (°)	0.295	0.1296	0.2082	**0.10545**
I_R (A)	670.560	28.4010	15.6880	**9.98960**
θ_{I_R} (°)	48.314	0.7901	0.3858	**0.25697**
pf_R	0.488	0.0141	0.0090	**0.00551**
$\lvert S_R \rvert$ (MVA)	311.920	14.0030	10.4670	**6.12870**
P_R (MW)	6.706	2.3172	0.5357	**0.48106**
Q_R (MVAr)	488.190	17.2610	12.3410	**7.31230**
REG (%)	24.360	0.3506	0.7229	**0.27752**
η (%)	2.072	0.7161	0.1656	**0.14867**
Receiving-End	**Performance of Scenario No.3**			
Variables	**Short**	**T-Model**	**Π-Model**	**M-Model**
V_R (kV)	1.8617	0.012710	0.018336	**0.009564**
θ_{V_R} (°)	0.0911	0.012587	0.024283	**0.012177**
I_R (A)	66.2780	0.228710	0.098951	**0.069297**
θ_{I_R} (°)	4.9050	0.011438	0.006251	**0.004046**
pf_R	0.0591	0.000270	0.000344	**0.000183**
$\lvert S_R \rvert$ (MVA)	29.9810	0.112880	0.069604	**0.043444**
P_R (MW)	1.1416	**0.015686**	0.076499	0.035693
Q_R (MVAr)	45.4210	0.193580	0.198380	**0.109570**
REG (%)	1.5188	0.006145	0.006530	**0.003586**
η (%)	0.3745	**0.005146**	0.025095	0.011709
Receiving-End	**Performance of Scenario No.4**			
Variables	**Short**	**T-Model**	**Π-Model**	**M-Model**
V_R (kV)	32.983	**0.07269**	0.96389	0.41627
θ_{V_R} (°)	0.444	0.05119	0.06614	**0.03489**
I_R (A)	278.940	7.13750	3.83260	**2.47160**
θ_{I_R} (°)	39.456	0.46485	0.23479	**0.15569**
pf_R	0.500	0.00831	0.00483	**0.00306**
$\lvert S_R \rvert$ (MVA)	203.260	5.43210	3.74290	**2.24570**
P_R (MW)	3.641	0.99558	0.30707	**0.24800**
Q_R (MVAr)	263.980	6.30090	4.19820	**2.54480**
REG (%)	15.810	**0.11489**	0.35744	0.14453
η (%)	2.445	0.66837	0.20615	**0.16649**

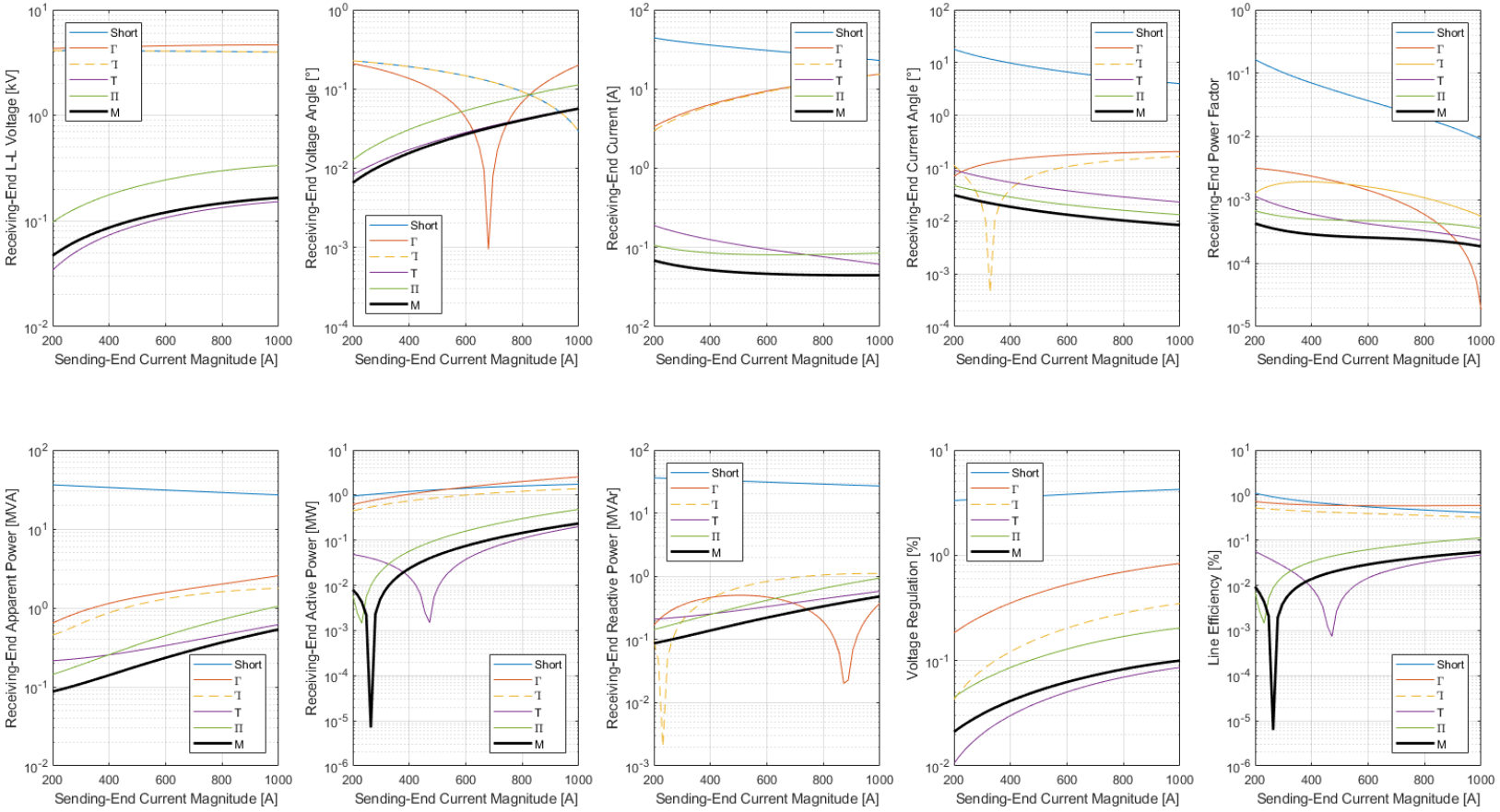


Figure 6.10: Absolute Errors Between the Lumped Parameter Models and the Distributed Parameter Model by Varying the Sending-End Current

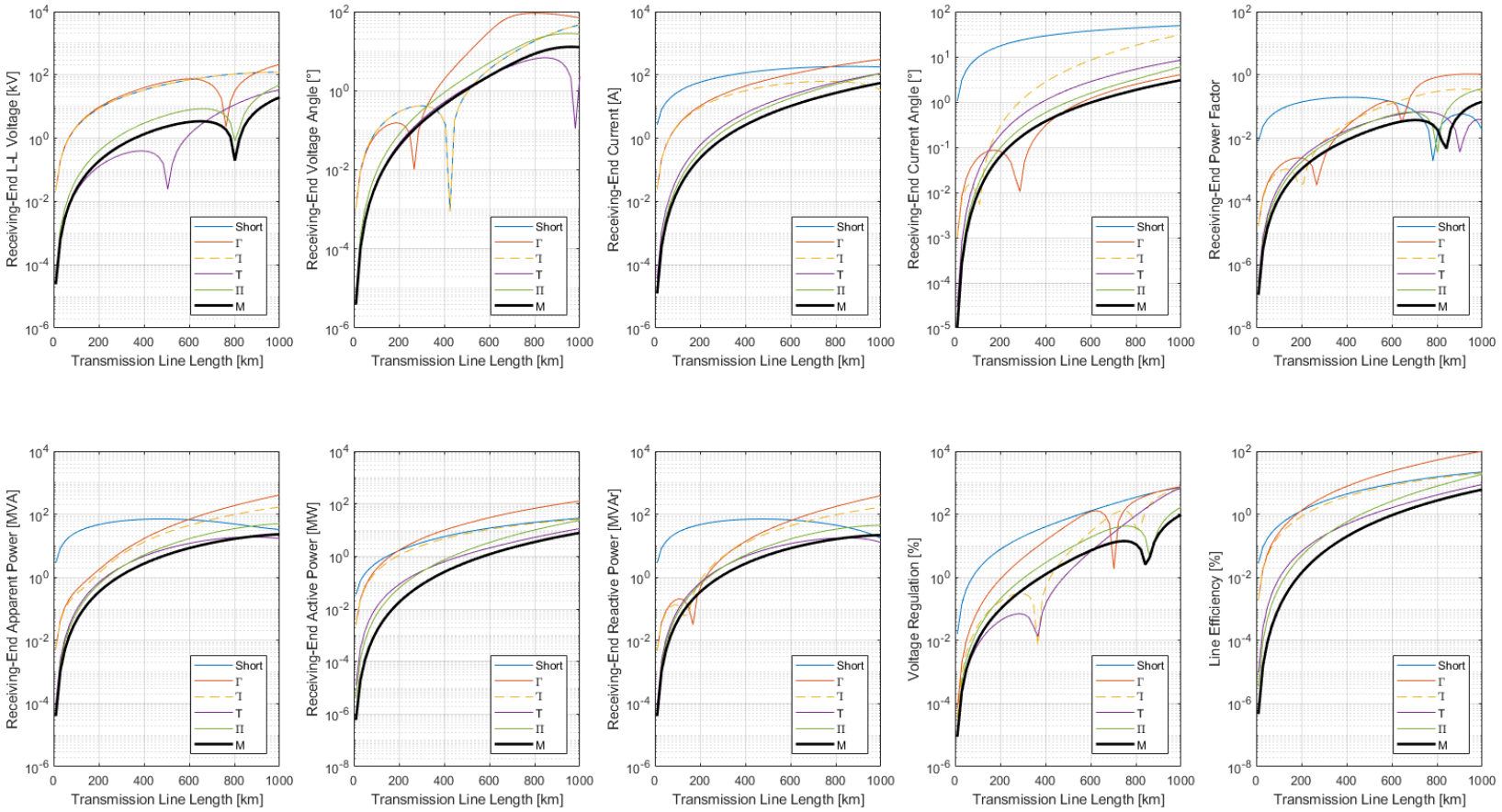


Figure 6.11: Absolute Errors Between the Lumped Parameter Models and the Distributed Parameter Model by Varying the Total Length

6.2.3.1 Equivalent Medium Line Models for the Distributed Parameters

Once the ABCD parameters of the long line model are obtained using (6.9), the accurate equivalent circuits of the T-, Π-, and M-models can be constructed. Since the M-model is new, so there is a need to construct the circuit shown in Figure 6.8 based on the values of the preceding ABCD parameters. But first, let's see the equivalent T- and Π-models.

Equivalent T-Model

The goal here is to convert the ABCD parameters of the two-port network given in (6.9) to the equivalent impedance and admittance shown in Figure 6.12. The full steps of this conversion procedure are given in Appendix K. The two important equations for the parameters shown in Figure 6.12 are:

$$\hat{Y} = Y\frac{\sinh(\gamma\mathfrak{L})}{\gamma\mathfrak{L}} \tag{6.13}$$

$$\frac{\hat{Z}}{2} = \frac{Z}{2}\cdot\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\frac{\gamma\mathfrak{L}}{2}} \tag{6.14}$$

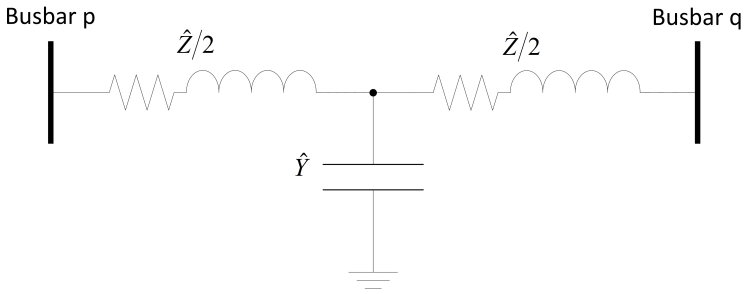


Figure 6.12: Equivalent T-Model for the Long Line Model

Equivalent Π-Model

Similar thing, the equations required to convert the ABCD parameters of (6.9) to the equivalent impedance and admittance shown in Figure 6.13 are:

$$\hat{Z} = Z\frac{\sinh(\gamma\mathfrak{L})}{\gamma\mathfrak{L}} \tag{6.15}$$

$$\frac{\hat{Y}}{2} = \frac{Y}{2}\cdot\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\frac{\gamma\mathfrak{L}}{2}} \tag{6.16}$$

where the full derivation is given in Appendix L.

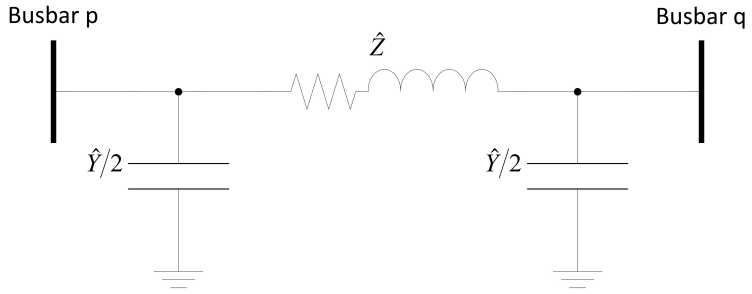


Figure 6.13: Equivalent Π-Model for the Long Line Model

Equivalent M-Model

Finding the preceding two equivalent models is an easy task. On the opposite side, it is very hard to find the equivalent M-model because the ABCD parameters of (6.12) have higher-order terms compared with the linear terms of (6.7) and (6.8). The detailed derivation of this accurate equivalent model is given in Appendix M. The two equations required to convert the ABCD parameters of (6.9) to the equivalent impedance and admittance shown in Figure 6.14 are:

$$\frac{\hat{Z}}{2} = \frac{-Z}{4} \cdot \frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\frac{\gamma\mathfrak{L}}{2}} \left[1 - \sqrt{5 + 4\cosh\left(\gamma\mathfrak{L}\right)}\right] \tag{6.17}$$

$$\frac{\hat{Y}}{3} = Y \frac{\coth\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\frac{\gamma\mathfrak{L}}{2}} \left[\frac{2 - \sqrt{5 + 4\cosh\left(\gamma\mathfrak{L}\right)} + \cosh\left(\gamma\mathfrak{L}\right)}{3 - \sqrt{5 + 4\cosh\left(\gamma\mathfrak{L}\right)} + 2\cosh\left(\gamma\mathfrak{L}\right)}\right] \tag{6.18}$$

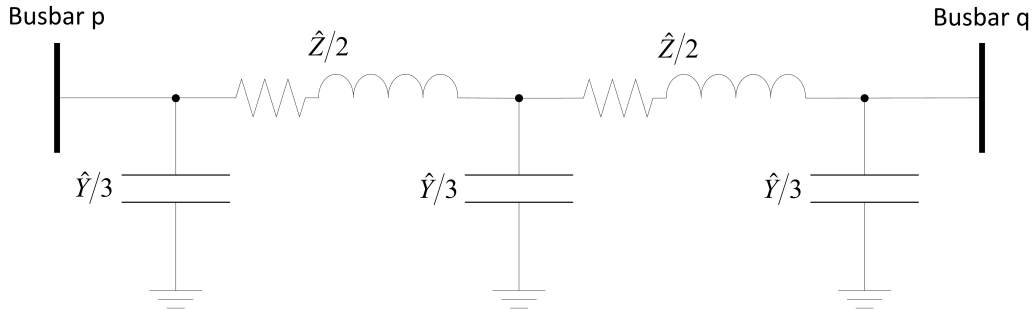


Figure 6.14: Equivalent M-Model for the Long Line Model

6.2.3.2 Optimal Lumped Parameters

In Figure 6.8, it has been seen that the two impedances and the three admittances are symmetrical. The same thing can be seen with the T- and Π-models. For many real reasons, these medium line models could have unsymmetrical parameters. Some of these reasons are[9]:

- The surrounding weather conditions are dynamic, and thus all the per-unit-length impedances are not constant.
- Because of the shunt admittance, so the sending-end and the receiving-end currents are not equal. This means that each per-unit-length impedance has a specific **self-heating** error.
- In the inclined-spans scenario[10], the **sag** is non-uniform. This could happen due to different tower heights or/and non-flat ground.

Thus, it could be possible to enhance the performance of the T-, Π- and M-models by searching for the optimal values of these lumped parameters.

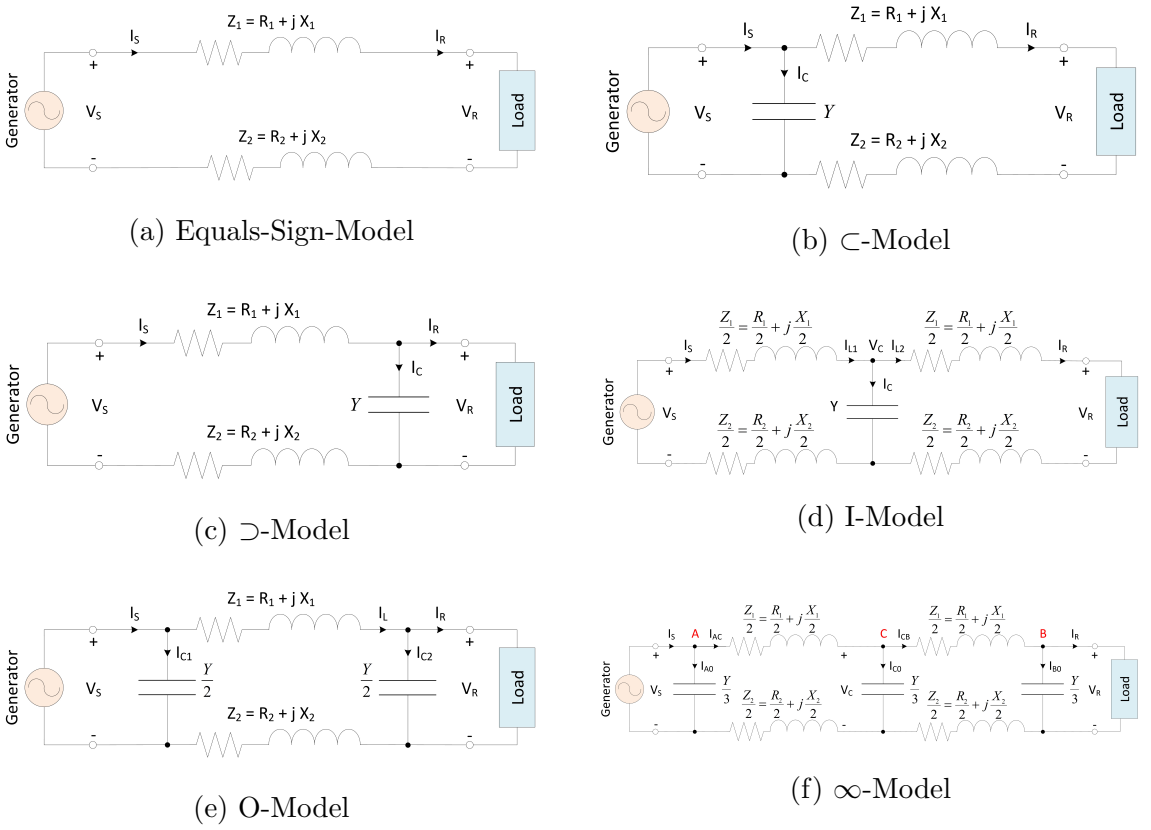
6.2.3.3 Piecewise Medium Line Model

Although the M-model provides the best accuracy in most readings of Table 6.2 and Figures 6.10-6.11, it has been seen from the preceding experiments that some other models could also win on some occasions. For example, in Figure 6.10, the T-model shows the best estimation of the receiving-end voltage V_R and the voltage regulation REG. Also, the Γ- and Ƭ-models can win in some small spots of some readings. Thus, a superior piecewise lumped parameter model could be designed to tackle the weaknesses of all the medium-length transmission line models. That is, based on the reading type and line data, the best model can be picked up.

6.2.3.4 Equivalent Circuits for Transmission Lines (Including Two-Line Circuits)

Some references consider the two-line representation of transmission lines. Thus, the distributed series parameters are divided into two parts where each part belongs to one

[9] These points will be discussed later in this chapter.
[10] It will be seen later in this chapter.



(a) Equals-Sign-Model

(b) ⊂-Model

(c) ⊃-Model

(d) I-Model

(e) O-Model

(f) ∞-Model

Figure 6.15: Transmission Line Models with the Two-Line Representation

line [289]. With this configuration, the T-model is called an **I-model** in [282, 377]. Also, the T-model is modified with this configuration in [154]. The shape of this configuration is similar to the superset symbol. Thus, it can be called a **⊃-model**. Similarly, the gamma shape of the Γ-model can be converted to be like the letter C or the subset symbol, and thus it can be called a **⊂-model**. Furthermore, the configured Π-model can be called an **O-model**, and the M-model can be called an **∞-model**. Even the short circuit model shown in Figure 6.1 can be configured with two impedances to represent the two-line circuits [289]. Figuratively, this modified short-length transmission line can be called an **Equals-Sign-model**. All these models can be shown in Figure 6.15.

If the sum of the series impedances of both lines equals the impedance of the corresponding single-line representation, then the ABCD parameters of these new models will be exactly similar to their original models. This point can be briefly clarified in Table 6.3.

Table 6.3: Steady-State Parameters of the Equals-Sign-, $\subset$-, $\supset$-, I-, O- and ∞-Models

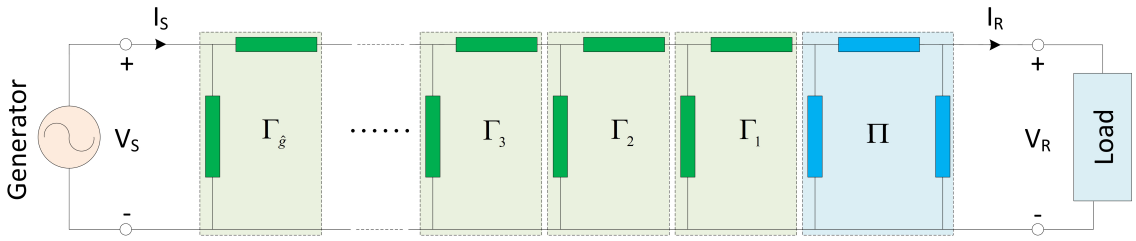
Modified Model	Original Model	Condition	ABCD Parameters
Equals-Sign-Model	Short Line Model	$Z_1 + Z_2 = Z$	Eq. (6.3)
$\subset$-Model	Γ-Model	$Z_1 + Z_2 = Z$	Eq. (6.5)
$\supset$-Model	$\reflectbox{\Gamma}$-Model	$Z_1 + Z_2 = Z$	Eq. (6.6)
I-Model	T-Model	$Z_1 + Z_2 = Z$	Eq. (6.7)
O-Model	Π-Model	$Z_1 + Z_2 = Z$	Eq. (6.8)
∞-Model	M-Model	$Z_1 + Z_2 = Z$	Eq. (6.12)

If the conditions given in Table 6.3 are not satisfied, then the ABCD parameters of these models will be different than those of the original models. This could be faced with single wire earth return (SWER) transmission line systems [330].

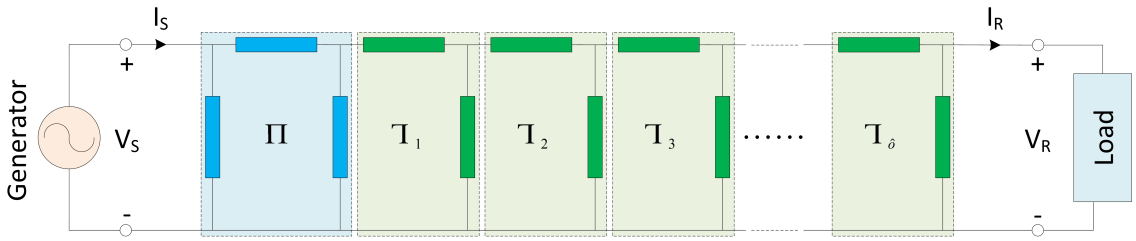
6.2.3.5 Extending the M-Model to a Comb-Model

By referring to Figures 6.3-6.6, it can be seen that the M-model is like a hybridization between the T-model and the Π-model. Also, the M-model can be seen as a combination of the Γ-model and the Π-model or a combination of the Π-model and the $\reflectbox{\Gamma}$-model. Thus, by intuition, the performance could be further enhanced by adding multiple Γ to the left-side of Π or multiple $\reflectbox{\Gamma}$ to the right-side of Π. This can be called a **Comb-model**, which is illustrated in Figure 6.16. The number of Γ or $\reflectbox{\Gamma}$ sectors is limited. Thus, the Comb-model is not a lumped parameter model nor a distributed parameter model. It is partially lumped and partially distributed.

The first approach shown in Figure 6.16a is preferred. The reason is that the voltage and current of each new sector can be respectively calculated by (I.7) and (I.9) until reaching the $\hat{g}$ sector. Although this new model has not been analyzed yet, it is expected that the performance could be enhanced by increasing $\hat{g}$ until reaching a saturated level where further increase in $\hat{g}$ is not worth anymore. This gives two facts: 1. the Comb-model is a hybrid between the lumped and distributed parameter models, and 2. the optimal $\hat{g}$ should be selected based on the solution quality and the circuit complexity.



(a) Version No.1



(b) Version No.2

Figure 6.16: Comb-Model Consists of One Π Sector and Multiple Γ or Τ Sectors

It is important to say that going with multiple shunt admittances is not recommended. This is why the Π-model is preferred over the T-model in many power system applications because the latter one creates one virtual bus and thus a **bus-reduction** technique should be applied. However, some other studies, like system stability and line performance, could take advantage of the Comb-model. This model could be studied in future work.

6.2.3.6 Other Benefits of the M-Model

Besides its great accuracy, the M-model has also some distinct applications. For example, the variation in line parameters[11] can be directly accounted. Thus, it can be used to realize the effects of sag on transmission lines. This will be seen later with some detailed information and innovative techniques.

6.3 Effects of Surrounding Weather Conditions, Cable Design, Load

[11]This variation happens due to many factors. For instance, temperature, wind direction, wind speed, humidity, snow, frequency, load, etc.

Condition and System Frequency on Distributed Parameters

In physics and different engineering disciplines, it is known that the conductor resistance proportionally increases with its temperature. Many applications based on this phenomenon can be seen in electric power systems, such as **thermal relays**, **sag cables**, **thermal ratings** of power cables, **superconducting materials**, **fault current limiters** (**FCLs**), etc. However, this phenomenon is almost completely neglected in power system studies, such as power flow (PF) analysis, economic load dispatch (ELD), optimal power flow (OPF), etc [138]. With significant temperature changes, the resistance of electric power components (including overhead lines, underground cables, and transformers) could be distinctly and markedly affected, and thus there is an argument about the optimality and feasibility of the classical steady-state solutions.

Many researchers have tried to solve this important issue by proposing some modifications and extensions to the existing mathematical models. However, these studies are either conducted for some specific power system studies or they complicate the original mathematical problem formulations. Some of these studies are [77, 332, 393] where their main drawbacks are summarized in [138]. Even in [138], the authors tried to solve this issue from just PF perspective, and the final model requires more CPU time than that of the conventional Newton-Raphson PF analysis where the **Jacobian matrix** is expanded with five extra sub-matrices. One of the most practical studies is presented in [82] where the authors mainly focused on how to model transmission lines themselves. However, only a few cases are covered with some serious simplifications. Also, that study does not show how to integrate the proposed model to solve other electric power system studies, which in turn make it uncompleted study.

This section tries to solve all these problems by correcting the existing transmission line models, which will be employed later to solve some electric power system problems.

6.3.1 Resistance

For each transmission line conductor, the calculation of the per-unit-length DC resistance can be used as a basis to determine the AC resistance. For DC, the cylindrical

conductor resistance is calculated as follows:

$$R_{\mathrm{DC}} = \frac{\rho \Delta x}{A} = \frac{\rho \Delta x}{\pi r^2} \tag{6.19}$$

where ρ is the conductor resistivity at a given temperature, Δx is the per-unit-length, A is the conductor cross-sectional area, and r is the conductor radius.

6.3.1.1 Frequency Effect

The North American nominal frequency is 60 Hz, and 50 Hz for the European standard. Although modern automation systems can keep the network frequency at its nominal value (i.e., 50 or 60 Hz), there is still a possibility to face some small disturbances. These variations can affect the actual values of the distributed parameters. Such effects are addressed below.

For AC, the conductor effective resistance is higher than that of the DC resistance due to the **skin effect** phenomenon. In DC circuits, the current is uniformly distributed over its cross-sectional area, while the current density in AC circuits tends to concentrate at the surface of the conductor. Thus, a **skin correction factor** k is applied to determine the effective resistance as follows [156]:

$$R(f) = kR_{\mathrm{DC}} \tag{6.20}$$

Different approaches could be used to find the mathematical expression of k. One of the popular approximations to find $R(f)$ is [230]:

$$R(f) \approx \frac{\Delta x \rho}{\pi(2r - \zeta)\zeta} \tag{6.21}$$

where ζ is called the **skin-depth**, which can be calculated as follows:

$$\zeta = \sqrt{\frac{2\rho}{\omega\mu}} = \sqrt{\frac{2\rho}{\omega\mu_r\mu_0}} = \sqrt{\frac{\rho}{\pi f \mu_r \mu_0}} \tag{6.22}$$

where μ_r is the relative magnetic **permeability** of the conductor and μ_0 is the permeability of free space.

Thus, (6.21) becomes:

$$R(f) \approx \frac{\Delta x \rho \sqrt{\frac{\pi f \mu_r \mu_0}{\rho}}}{\pi\left(2r - \sqrt{\frac{\rho}{\pi f \mu_r \mu_0}}\right)} \tag{6.23}$$

$$\approx \frac{\Delta x \sqrt{\frac{\rho f \mu_r \mu_0}{\pi}}}{2r - \sqrt{\frac{\rho}{\pi f \mu_r \mu_0}}} \tag{6.24}$$

Also, there are some other skin effect formulas used to calculate the complex impedance, such as the one given in [289].

6.3.1.2 Temperature Effect

The resistivity ρ of any conductive material depends on the temperature of that conductor. Different formulas can be used to calculate the resistivity ρ as a function of its temperature $\breve{T}$. From [138, 180], the following formula is used:

$$\rho(\breve{T}) = \rho_0 \left(\frac{\breve{T} + \breve{T}_F}{\breve{T}_0 + \breve{T}_F} \right) \tag{6.25}$$

where $\breve{T}$ and $\breve{T}_0$ are respectively the present and reference temperatures of the conductor. ρ and ρ_0 are the conductor resistivity at $\breve{T}$ and $\breve{T}_0$, respectively. $\breve{T}_F$ is a temperature constant, which depends on the metal type. The more precise formula can be found in [102]:

$$\frac{\Delta\rho}{\rho_0} = \alpha_1 \left(\breve{T} - \breve{T}_0 \right) + \alpha_2 \left(\breve{T} - \breve{T}_0 \right)^2 + \cdots + \alpha_n \left(\breve{T} - \breve{T}_0 \right)^n \tag{6.26}$$

where $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ are the **series resistance temperature coefficients**.

Because the higher-order terms have very small contributions, so they are dropped from the equation in most instrumentation and power system applications. Thus, (6.26) can be simplified to be in a linear form as follows:

$$\rho(\breve{T}) = \rho_0 \left[1 + \alpha \left(\breve{T} - \breve{T}_0 \right) \right] \tag{6.27}$$

where α is α_1 in (6.26), and $\breve{T}_F$ of (6.25) is an estimate of this coefficient.

Substituting (6.27) in (6.24) for ρ yields:

$$R(f, \breve{T}) = \frac{\Delta x \sqrt{\frac{f\mu_r\mu_0\rho_0\left[1+\alpha\left(\breve{T}-\breve{T}_0\right)\right]}{\pi}}}{2r - \sqrt{\frac{\rho_0\left[1+\alpha\left(\breve{T}-\breve{T}_0\right)\right]}{\pi f\mu_r\mu_0}}} \tag{6.28}$$

6.3.1.3 Relative Humidity, Temperature, and Pressure Effects

In the preceding equations, the permeability of free space μ_0 is taken as a constant, which is equal to $\mu_0 = 4\pi \times 10^{-7}$ H/m. Real transmission lines are exposed to dynamic changes in temperature, pressure, humidity, and other weather conditions. Thus, the

permeability constant μ_0 should be replaced with the parameter μ_{air} where the latter one is prone to change. This realized parameter can be calculated as follows:

$$\mu_{\text{air}} = \frac{n_{\text{air}}^2}{c_0^2\ \varepsilon_{\text{air}}} \quad [\text{H/m}] \tag{6.29}$$

where c_0, n_{air} and ε_{air} are respectively the speed of light in vacuum, the **refractive index** of air, and the **permittivity** of air.

There are two approaches to find the refractive index of air. The first one is to approximate it with a constant value, which is around $n_{\text{air}} = 1.000293$ [204]. The more realized value can be determined by taking n_{air} as a function of temperature and pressure as follows [120]:

$$\begin{aligned} n_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}} \quad = \quad 1 + \Bigg[& \frac{0.00138823\mathcal{P}_a\,(1.5291 + 0.00360\nu)\left(8342.13 + \frac{2406030}{130-\nu^2} + \frac{15997}{38.9-\nu^2}\right)}{1 + 0.003671\breve{T}_a} \\ & - \mathfrak{f}\left(5.722 - 0.0457\nu^2\right) \Bigg] \times 10^{-8} \end{aligned} \tag{6.30}$$

where $\mathcal{P}_a$ and $\breve{T}_a$ are respectively the atmospheric pressure (in torr) and the ambient temperature (in °C). Also, ν and $\mathfrak{f}$ are the vacuum **wave-number** (in μm^{-1}) and the pressure of water vapour (in torr), respectively.

The permittivity of air ε_{air} can be expressed as a function of temperature, pressure, and relative humidity as follows [176]:

$$\varepsilon_{\text{air}} = \varepsilon_0 \left[1 + \frac{1.5826}{\breve{T}_{\text{abs}}}\left(\mathcal{P}_{\text{ma}} + \frac{0.36\mathcal{P}_{\text{sw}}}{\breve{T}_{\text{abs}}}\text{RH}\right) \times 10^{-6}\right] \quad [\text{F/m}] \tag{6.31}$$

where $\mathcal{P}_{\text{ma}}$ and $\mathcal{P}_{\text{sw}}$ are respectively the pressure of moist air (in Pa) and the pressure of saturated water vapour (in Pa). The symbols $\breve{T}_{\text{abs}}$ and RH are the absolute temperature (in K) and the relative humidity (in %), respectively. Finally, ε_0 is the permittivity of free space, which can be approximated to $\varepsilon_0 \approx 8.854187817620389 \times 10^{-12}$ F/m [253]. The whole derivation of n_{air} is given in Appendix N.

Summing up all these equations leads to the following realized resistance:

$$R\left(f, \mathfrak{f}, \breve{T}, \breve{T}_a, \breve{T}_{\text{abs}}, \text{RH}, \mathcal{P}_a, \mathcal{P}_{\text{ma}}, \mathcal{P}_{\text{sw}}\right) = \frac{\Delta x \sqrt{\frac{f\mu_r\mu_{\text{air}}\rho_0\left[1+\alpha\left(\breve{T}-\breve{T}_0\right)\right]}{\pi}}}{2r - \sqrt{\frac{\rho_0\left[1+\alpha\left(\breve{T}-\breve{T}_0\right)\right]}{\pi f\mu_r\mu_{\text{air}}}}} \tag{6.32}$$

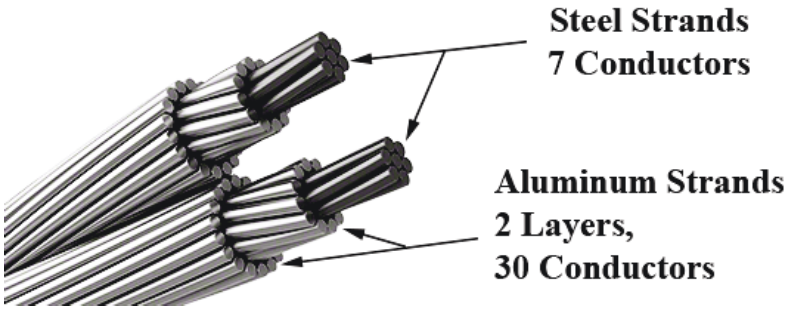


Figure 6.17: Aluminium-Conductor-Steel-Reinforced (ACSR) Cable

6.3.1.4 Snow, Ice, Wind, and Sandstorm Effects

Their main effects can be seen during calculating the loading of cables with sag and tension. This happens because the extra weight gained from snow, ice, wind or sandstorm forces the cable to bend, and thus all the distributed parameters can be affected. The detailed information about these effects will be discussed later when adopting the M-model to realize the **sagging phenomenon**. Also, heavy falling snow or sandstorm of charged dust particles could affect the magnetic permeability of air μ_{air} and the dielectric properties between transmission lines and earth [222]. There is a lack of information about these cases.

6.3.1.5 Spiraling and Bundle Conductor Effects

Commercial transmission line cables come with multiple conductors arranged in multiple layers. They are available in the market with different designs, such as AAC, AAAC, ACAR, and ACSR[12]. The last type is the most popular one, which is shown in Figure 6.17. As can be clearly seen, ACSR has multiple layers and each one has multiple conductors. The stranded conductors of the outer layers are made of aluminum, while the stranded conductors of the core or the inner layers are made of steel to provide a high ratio between strength and weight. The reason for using opposite spiral shapes of conductors between layers is to hold them together.

As can be concluded, two important things need to be taken into account: 1. the actual conductor length is longer than the apparent length because of the spiral

[12]AAC: all-aluminum-conductor, AAAC: all-aluminum-alloy-conductor, ACAR: aluminum-conductor-alloy-reinforced, and ACSR: aluminium-conductor-steel-reinforced.

shape, and 2. the transmission line cables are made of multiple conductors. Thus, the total resistance should account for these two facts.

The per-unit-length resistance of each wound conductor, belonging to any layer, can be calculated as follows [156]:

$$R_{\text{cond}} = \frac{\rho}{A}\sqrt{1+\left(\pi\frac{1}{\mathfrak{p}_{\text{cond}}}\right)^2} \quad [\Omega/\text{m}] \tag{6.33}$$

where: R_{cond} = wound conductor resistance (in Ω)

$\sqrt{1+\left(\pi\frac{1}{\mathfrak{p}_{\text{cond}}}\right)^2}$ = wound conductor length (in m)

$\mathfrak{p}_{\text{cond}} = \frac{l_{\text{turn}}}{2r_{\text{layer}}}$ = wound conductor relative pitch

l_{turn} = length of one turn of the spiral (in m)

$2r_{\text{layer}}$ = diameter of the layer (in m)

By assuming each ith layer has n_i conductors with the same diameter, then the total resistance of the ith parallel combination can be calculated as follows:

$$R_{\text{layer}_i} = \frac{1}{\sum_{i=1}^{n_i}\frac{1}{R_{\text{cond}_i}}} \quad [\Omega/\text{m}] \tag{6.34}$$

Because there are m parallel layers, so the total per-unit-length resistance can be computed as follows:

$$R = \frac{1}{\sum_{j=1}^{m}\frac{1}{R_{\text{layer}_j}}} \quad [\Omega/\text{m}] \tag{6.35}$$

6.3.1.6 Other Aspects

The word "branch" could mean a cable or a transformer. For cable, it could be an underground cable or an overhead line. Because the structure and materials of transformers are different than those of powerlines, so the transformer resistance calculation is also different. Further, the overhead line resistance calculation is different than that of underground cables. Some of the main reasons are:

- Overhead lines are prone to sag.
- Overhead lines have different heat transfer mediums.
- Overhead lines have different designs and materials.

A good review of these aspects can be found in [138].

6.3.2 Inductive Reactance

The line inductance is the most effective part of transmission lines. In the short-length transmission line model, the capacitance and conductance parameters are neglected. Even the resistance parameter is neglected in some power system analysis, such as small disturbance power system stability studies [123, 213, 291]. Thus, special care should be given to this parameter to ensure having a precise approximation of actual transmission lines. To see how, first, let's consider the following nominal inductive reactance X_{L_0}:

$$X_{L_0} = \omega_0 L_0 = 2\pi f_0 L_0 \tag{6.36}$$

where L_0 is the nominal inductance of the conductor, and ω_0 and f_0 are respectively the nominal angular and regular frequencies.

This inductive reactance is added to the resistance as an imaginary component to have a complex impedance Z. This impedance is divided into two parts as follows [123, 156, 326]:

$$Z = Z_{\text{int}} + Z_{\text{ext}} \tag{6.37}$$

where Z_{int} and Z_{ext} are the internal and external impedances, respectively.

These two parts are subject to different dynamic changes in the system and the surrounding weather conditions.

6.3.2.1 Frequency Effect

Similar to the resistance parameter, practically, maintaining the system frequency at its nominal value f_0 with zero error is impossible. The variation in frequency ($\Delta f = f - f_0$) has a direct impact on both parts of Z given in (6.37). For Z_{ext}, the following equation can be used [156, 318]:

$$Z_{\text{ext}} = jX_{\text{ext}} = j\omega L_{\text{ext}} = j2\pi f L_{\text{ext}} \tag{6.38}$$

where L_{ext} is the external inductance and X_{ext} is the external inductive reactance.

It is obvious that for any Δf variation, Z_{ext} changes by $j2\pi L_{\text{ext}}\Delta f$. For Z_{int}, the skin effect must be realized. Different approaches can be applied to achieve that. The one reported in [318] is adopted here, which is expressed as follows:

$$Z_{\text{int}}(f) = R_{\text{DC}} \frac{\upsilon_r}{2} \cdot \frac{J_0(\upsilon_r)}{J_1(\upsilon_r)} \tag{6.39}$$

where $J_0(\upsilon_r)$ and $J_1(\upsilon_r)$ are respectively the zero-order and the first-order **Bessel functions** of the first kind, and υ_r is expressed as follows:

$$\upsilon_r = j^{3/2}\sqrt{2}\frac{r}{\zeta} \tag{6.40}$$

where r is the radius of the conductor, and ζ is the skin-depth expressed earlier in (6.22).

6.3.2.2 Weather Effect

First of all, it has to be said that the inductance could be affected when there is a sag caused by wind, snow, ice, sandstorm, etc. This part will be discussed later in its section. The other variable that causes transmission lines to sag is the temperature. Thus, both X_{ext} and X_{int} are indirectly affected by $\breve{T}$ due to the sagging phenomenon, which can be mathematically expressed as follows [12, 82]:

$$X_{\text{ext}}(f, \breve{T}) = 2\pi f L_{\text{ext}_0}\left[1 + \beta_{\text{ext}}\left(\breve{T} - \breve{T}_0\right)\right] \tag{6.41}$$

$$X_{\text{int}}(f, \breve{T}) = 2\pi f L_{\text{int}_0}\left[1 + \beta_{\text{int}}\left(\breve{T} - \breve{T}_0\right)\right] \tag{6.42}$$

where L_{ext_0} and L_{int_0} are respectively the nominal external and internal inductance of the conductor at $\breve{T}_0$, and β_{ext} and β_{int} are respectively the external and internal **series inductance temperature coefficients** of the conductor.

More than that, by referring to any electric power systems textbook, the permeability of free space μ_0 is involved in determining the distributed inductance. Thus, both L_{ext_0} and L_{int_0} tend to change due to the variation in μ_{air}. Also, the skin-depth ζ given in (6.22) is a function of ρ, f and μ_{air}. Furthermore, ρ is a function of $\breve{T}$, and μ_{air} is a function of n_{air} and ε_{air}. Thus, the skin-depth ζ can be re-expressed as follows:

$$\zeta\left(f, \mathfrak{f}, \breve{T}, \breve{T}_a, \breve{T}_{\text{abs}}, \text{RH}, \mathcal{P}_a, \mathcal{P}_{\text{ma}}, \mathcal{P}_{\text{sw}}\right) = \sqrt{\frac{\rho_0\left[1 + \alpha\left(\breve{T} - \breve{T}_0\right)\right]}{\pi f \mu_r \mu_{\text{air}}}} \tag{6.43}$$

The other consideration is about the flux-linkage in terms of self- and mutual-inductance. These values could be temporarily affected by strong wind, snow, ice or sandstorm. They can also be permanently affected by design aspects, such as symmetrical/unsymmetrical spacing between lines and whether they are well-transposed or not. More information about how to calculate the inductance per their geometry and cable design can be found in [69, 123, 155, 156, 326], which is out of scope.

6.3.3 Capacitive Reactance

The capacitive reactance is considered for the medium- and long-line models. The per-unit-length nominal capacitive reactance can be calculated as follows:

$$X_{C_0} = \frac{1}{\omega C_0} = \frac{1}{2\pi f_0 C_0} \tag{6.44}$$

where C_0 is the nominal shunt capacitance.

This distributed parameter is subject to many dynamic changes, which are covered in the following lines.

6.3.3.1 Frequency Effect

Based on (6.44), the capacitive reactance depends on the system frequency. Because f is not constant, so the preceding equation should be modified to account the effect of this variation as follows:

$$X_C(f) = \frac{1}{2\pi f C_0} = \frac{1}{2\pi f_0 C_0 + 2\pi \Delta f C_0} \tag{6.45}$$

6.3.3.2 Weather Effect

Similar to the series inductive reactance X_L, the shunt capacitive reactance X_C is a function of many atmospheric variables. The variable that is repeatedly introduced in the literature is the temperature $\breve{T}$ of the conductor. The capacitance is indirectly affected by $\breve{T}$ due to the sagging phenomenon, which can be mathematically expressed as follows [12, 82, 219]:

$$X_C(f, \breve{T}) = \frac{1}{2\pi f C_0 \left[1 + \kappa \left(\breve{T} - \breve{T}_0\right)\right]} \tag{6.46}$$

where C_0 is calculated at f_0 and T_0, and κ is the **shunt capacitance temperature coefficient**.

Also, there are some other atmospheric variables that could affect X_C. For example, the conductor could gain extra weight due to wind, snow, ice or sandstorm, so it will tend to sag. Moreover, C_0 given in (6.46) can be further realized by involving the dynamic change in the dielectric of each per-unit-length capacitance. First, let's consider the following scenarios [123, 155, 156, 326]:

- **Capacitance of a Single-Phase Line with Two Wires:**

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{D}{r}\right]} \quad [\mathrm{F/m}] \tag{6.47}$$

where D is the distance between the two wires with radius r.

- **Capacitance of a Three-Phase Line:**

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{\mathrm{GMD}}{r}\right]} \quad [\mathrm{F/m}] \tag{6.48}$$

where GMD can be calculated as follows:

$$\mathrm{GMD} = \sqrt[3]{D_{\mathrm{AB}} D_{\mathrm{BC}} D_{\mathrm{CA}}} \quad [\mathrm{m}] \tag{6.49}$$

where D_{AB} is the distance between nphase A and phase B, and similar thing for D_{BC} and D_{CA}.

- **Capacitance of Stranded Bundle Conductors:**

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{\mathrm{GMD}}{r_e}\right]} \quad [\mathrm{F/m}] \tag{6.50}$$

where r_e is the equivalent radius of n conductors per bundle, which can be calculated as follows:

$$r_e = \begin{cases} \sqrt[n]{d^{n-1} - r}, & \text{if } 1 \leqslant n \leq 3 \\ 1.09\sqrt[4]{d^3 r}, & \text{if } n = 4 \end{cases} \tag{6.51}$$

where d and n are respectively the distance between bundle conductors (in meters) and the number of conductors per bundle.

- **Capacitance Due to Earth:**

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{D}{r}\right] - \ln\left[\frac{\sqrt[3]{D_{\mathrm{AB'}} D_{\mathrm{BC'}} D_{\mathrm{CA'}}}}{\sqrt[3]{D_{\mathrm{AA'}} D_{\mathrm{BB'}} D_{\mathrm{CC'}}}}\right]} \quad [\mathrm{F/m}] \tag{6.52}$$

where $D_{\mathrm{AB'}}$ is the distance between the overhead conductor A and the image conductor B$'$. Similar thing for $D_{\mathrm{BC'}}$ and $D_{\mathrm{CA'}}$. Also, $D_{\mathrm{AA'}}$ is the distance between the overhead conductor A and its image A$'$ located underground. Similar thing for $D_{\mathrm{BB'}}$ and $D_{\mathrm{CC'}}$.

The distance from the overhead conductor A to the earth's surface and that from the earth's surface to the image conductor A$'$ are exactly the same (i.e., $D_{\mathrm{AA}'} = 2D_{\mathrm{A}}$). Thus, (6.52) can also be expressed as follows:

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{D}{r}\right] - \ln\left[\frac{\sqrt[3]{D_{\mathrm{AB}'}D_{\mathrm{BC}'}D_{\mathrm{CA}'}}}{2\sqrt[3]{D_{\mathrm{A}}D_{\mathrm{B}}D_{\mathrm{C}}}}\right]} \quad [\mathrm{F/m}] \tag{6.53}$$

- **Capacitance of Three-Phase Double-Circuit Lines:**

$$C_{\mathrm{AN}} = \frac{2\pi\varepsilon_o}{\ln\left[\frac{\mathrm{GMD}}{\mathrm{GMR_C}}\right]} \quad [\mathrm{F/m}] \tag{6.54}$$

where $\mathrm{GMR_C}$ can be calculated as follows:

$$\mathrm{GMR_C} = \sqrt[3]{r_{\mathrm{A}}r_{\mathrm{B}}r_{\mathrm{C}}} \tag{6.55}$$

and these $\{r_{\mathrm{A}}, r_{\mathrm{B}}, r_{\mathrm{C}}\}$ terms can be calculated as follows:

$$\begin{aligned} r_{\mathrm{A}} &= \sqrt{r_e D_{\mathrm{A_1A_2}}} \\ r_{\mathrm{B}} &= \sqrt{r_e D_{\mathrm{B_1B_2}}} \\ r_{\mathrm{C}} &= \sqrt{r_e D_{\mathrm{C_1C_2}}} \end{aligned} \tag{6.56}$$

where the subscripts $\mathrm{A_1}$ and $\mathrm{A_2}$ denote phase A of the first and second circuits of three-phase double-circuit transmission lines. Similar thing for phase B and phase C.

The preceding per-unit-length phase capacitance can be realized by replacing the vacuum permittivity ε_o with $\varepsilon_{\mathrm{air}}$ given in (6.31). Thus, (6.46) becomes:

$$X_C\left(f, \breve{T}, \breve{T}_{\mathrm{abs}}, \mathrm{RH}, \mathcal{P}_{\mathrm{ma}}, \mathcal{P}_{\mathrm{sw}}\right) = \frac{1}{2\pi f C\left[1 + \kappa\left(\breve{T} - \breve{T}_0\right)\right]} \tag{6.57}$$

Some studies show that the dielectric permittivity could be affected by snow. For example, the study reported in [5] shows that, for different frequency ranges, the dielectric permittivity can be affected by various snow parameters. Such parameters are density, shape and grain-size, liquid water content, shape and distribution of liquid inclusions and content of impurities. For the low-frequency range, the most effective parameters are the grain size and shape and snow density. This result is

emphasized by NASA in [66]. The other possible atmospheric condition that might have an impact on the dielectric permittivity is rain [61]. Also, it could be affected if a transmission line installed in a desert is hit by a strong sandstorm carrying charged sand particles [222, 398].

6.3.4 Conductance

The shunt conductance G is very small, and thus this distributed parameter is considered just for the long-line model[13]. However, we have to admit that almost all the references either neglect this parameter or taking it as a constant value. Based on our extensive search, only a few references, such as William H. Hayt in [163], describe this parameter with some mathematical equations. For two-wire system operated under low frequencies[14], the per-unit-length conductance can be calculated as follows:

$$G_{\text{L-L}} = \frac{\pi\sigma_0}{\cosh^{-1}\left[\frac{D}{2r}\right]} \quad [\text{℧/m}] \text{ or } [\text{S/m}] \tag{6.58}$$

where σ_0 is the nominal conductivity of the medium between the two conductors.

By analogy, the same steps applied to capacitance can also be applied here to calculate the conductance between one wire and the neutral:

$$G_{\text{AN}} = \frac{2\pi\sigma_0}{\ln\left[\frac{D}{r}\right]} \quad [\text{℧/m}] \text{ or } [\text{S/m}] \tag{6.59}$$

where $\ln\left[\frac{D}{r}\right] \approx \cosh^{-1}\left[\frac{D}{2r}\right]$ when $D \gg 2r$, which is the case faced here with real transmission lines.

Similar to (6.48), the conductance of a three-phase line could be estimated by the following modified formula:

$$G_{\text{AN}} = \frac{2\pi\sigma_o}{\ln\left[\frac{\text{GMD}}{r}\right]} \quad [\text{℧/m}] \text{ or } [\text{S/m}] \tag{6.60}$$

where GMD can be calculated by (6.49).

6.3.4.1 Frequency Effect

It could be strange to relate the system frequency here. From the preceding parameters, it has been seen that the series impedance can be markedly affected by f. As

[13] Sometimes, it is neglected even with the long line model [326].
[14] It is suitable for 50 - 60 Hz.

the line impedance increases the power loss increases as well. Thus, the conductor temperature $\breve{T}$ will raise causing the line to sag, which in turn raises the conductance.

6.3.4.2 Design Effect

It has been seen that the system frequency can indirectly affect the conductance of the line by affecting the conductor temperature. Thus, the other abandoned factors that might affect the line conductance are the type, material(s), and paint color of transmission line conductors. That is, based on the heat transfer between conductors and the surrounding medium, the **thermal conductivity** k (in W/m.K), the **heat transfer coefficient** h (in W/m^2.K) and the **grey body emissivity** ε[15] can also indirectly affect the conductance [143]. The relation is as follows: the heat gained from or dissipated to the air depends on $\{k, h, \varepsilon\}$, so $\breve{T}$ depends on these heat parameters.

6.3.4.3 Weather Effects

It is concluded that the conductance can be altered by $\breve{T}$. The basic mathematical equation of this temperature-dependent function can be expressed as follows [219]:

$$G(\breve{T}) = \frac{1}{\tilde{R}_0 \left[1 + \lambda \left(\breve{T} - \breve{T}_0\right)\right]} \tag{6.61}$$

where $\tilde{R}_0$ is the shunt resistance at $\breve{T}_0$ and f_0, and λ is the **shunt resistance temperature coefficient**.

Based on the earlier discussion, transmission line conductors tend to sag based on many other variables, such as ice, snow, sand, wind, cable material, cable type, cable age, etc. Thus, the simple equation given in (6.61) can further be realized. Add to that, the preceding frequency and design effects have just popped-up many other hidden and abandoned variables that might have some direct and indirect effects on the line conductance. Based on the heat transfer topic, transmission line conductors are subject to three different types of heat transfer called **radiation**, **convection**, and **conduction** [143]. Thus, the wind speed, wind direction, ambient temperature $\breve{T}_a$, atmospheric pressure $\mathcal{P}_a$, relative humidity RH, fog, aerosol, rain, snow (shape, size, weight, etc), dense charged sand particles (due to sandstorm), air pollution and

[15] Be note that ε is different than the permittivity of free space ε_0 and the permittivity of air ε_{air}.

the concentration of each content, type of land (dry soil, wet soil, near a river, lake, sea, etc), solar radiation intensity and many other hidden variables could contribute on the final value of G. The **mass transfer** topic also plays an important role here, because it explains many vital processes, such as **evaporation**, **absorption**, **drying**, and **precipitation**. That is, this topic explains the phase and movement of water and the status of water resources near transmission lines. Because many of these variables depend on the location, season and time, so these three variables could also be considered. Some pieces of the evidence to our claims are reported in [53, 191, 222, 283, 398].

By referring to the main subject of the conductance, this G represents the loss caused by the leakage current through the dielectric material separating the conductors. This dielectric material is simply the surrounding air containing ions. Thus, the conductivity of the air can be obtained by the number of ions per unit volume of air. For example, almost one century ago, Noto shows that the conductivity of clouds is far larger than that of dry air, and it varies based on the density of the cloud [283]. Because there are many similarities between clouds and fogs, so it is expected to see some similarities during foggy days [191]. This can be concluded by knowing that the corona losses under dry weather are much less than that during rainy or snowy days [156]. The study reported in [53] shows that the conductivity and the corona losses are proportional to the air density. It also shows that the conductivity increases when snow or dew forms on the surface of the conductor, which emphasizes some of our claims raised before. Thus, the conductivity decreases as the load current increases, because $\breve{T}$ is proportional to the load current. Furthermore, dense charged sand particles caused by sandstorms could affect the number of ions in the medium between conductors [222, 398]. More than that, the number of ions could be affected by a wide range of air pollutants, which in turn can affect the conductivity of the surrounding air [200]. Some mathematical equations to derive the conductivity σ_{air} as a function of some atmospheric variables in the absence and presence of aerosols are also given in [200], which can be considered as a good source for this realization work.

All these factors, variables, and uncertainties make predicting the actual conductance a very challenging task, and it might be the reason why almost all power system

references just neglect the parameter G or taking it as a constant value. Thus, there is a need to conduct more research on this topic where ML tools could provide some promising results[16].

6.4 Modeling Transmission Lines with Considering Temperature and Frequency Effects

In the first two sections, many transmission line models have been discussed. However, these models are valid just with nominal or steady-state distributed parameters. It has been seen, in the last section, that these parameters are not constants and they are subject to many dynamic changes due to operating conditions, weather conditions, surrounding environment and cable design/status/age. The goal of this section is to re-model transmission lines by just considering the temperature and frequency effects to have **temperature/frequency-based** (**TFB**) models. Once the idea is well captured, the same steps can be applied to any other variable.

6.4.1 Realizing Short-Length Transmission Line Model

The conventional ABCD parameters of this two-port network are given in (6.3). If $\breve{T}$ and ω $(= 2\pi f)$ are taken into account, then the receiving-end voltage V_R and current I_R should be modified as follows:

$$V_R(\omega, \breve{T}) = V_S(\omega, \breve{T}) - Z(\omega, \breve{T}) I_S(\omega, \breve{T}) \tag{6.62}$$

$$I_R(\omega, \breve{T}) = I_S(\omega, \breve{T}) \tag{6.63}$$

where the temperature/frequency-varying impedance $Z(\omega, \breve{T})$ can be calculated from the real and imaginary parts (i.e., $R(\omega, \breve{T})$ and $X_L(\omega, \breve{T})$) as shown in Figure 6.18.

[16] Chapter 10 presents some novel ML computing systems that can translate datasets as complicated nonlinear equations. Thus, it is very interesting if there is a way to translate the variation in conductivity as a mathematical model.

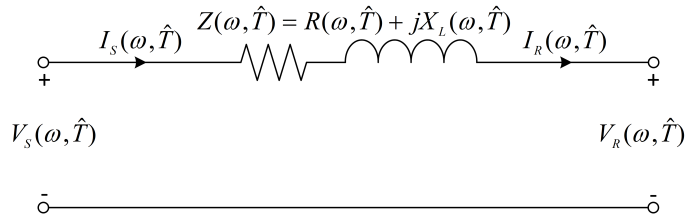


Figure 6.18: Realized Short-Length Transmission Line

However, since the temperature is not uniformly distributed along lines, the lumped impedance cannot be represented as a sum of per-unit-length impedances $Z = z\mathfrak{L}$ (where $\mathfrak{L}$ is the total length). Instead, it should be integrated as follows:

$$\begin{aligned} Z(\omega,\breve{T}) &= r_{\omega,0}\int_{\mathfrak{L}_{\min}}^{\mathfrak{L}_{\max}}\left\{1+\alpha\left[\breve{T}(x)-\breve{T}_0\right]\right\}dx \\ &\quad + j\omega l_0\int_{\mathfrak{L}_{\min}}^{\mathfrak{L}_{\max}}\left\{1+\beta\left[\breve{T}(x)-\breve{T}_0\right]\right\}dx \end{aligned} \tag{6.64}$$

where $r_{\omega,0}$ and l_0 are respectively the per-unit-length resistance and inductance at $\breve{T}_0$, and $\mathfrak{L} = \mathfrak{L}_{\max} - \mathfrak{L}_{\min}$. $\breve{T}(x)$ is a distance-based temperature function, which is hard to be obtained mathematically, especially with many uncertainties due to randomness and fuzziness. Therefore, (6.64) is discretized into N zones as follows:

$$Z(\omega,\breve{T}) \approx \sum_{k=1}^{N}\left[r_k\left(\omega,\breve{T}_k\right) + jx_{l,k}\left(\omega,\breve{T}_k\right)\right] \tag{6.65}$$

where r_k and $x_{l,k}$ are respectively the conductor resistance and inductive reactance at the kth zone, and it is assumed that the temperature $\breve{T}_k$ changes in steps from k to $k+1$.

6.4.2 Realizing Medium-Length Transmission Line Models

Similar to the previous model, the distributed parameters can be lumped. However, different models are used to represent medium lines, such as the $\{\Gamma, \text{T}, \text{T}, \Pi\}$ models and our M-model. Based on Figures 6.10-6.11, the performance of the first two models are bad. Thus, only the $\{\text{T}, \Pi, \text{M}\}$ models are covered in this realization section where the others can be realized by applying the same steps.

6.4.2.1 T-Model

The single-line diagram of this realized circuit is shown in Figure 6.19. For this model, the $\{A, B, C, D\}$ parameters given in (6.7) are realized as follows:

$$A(\omega, \breve{T}) = 1 + Z(\omega, \breve{T})Y(\omega, \breve{T})/2 \tag{6.66}$$

$$B(\omega, \breve{T}) = Z(\omega, \breve{T}) + Z^2(\omega, \breve{T})Y(\omega, \breve{T})/4 \tag{6.67}$$

$$C(\omega, \breve{T}) = Y(\omega, \breve{T}) \tag{6.68}$$

$$D(\omega, \breve{T}) = A(\omega, \breve{T}) \tag{6.69}$$

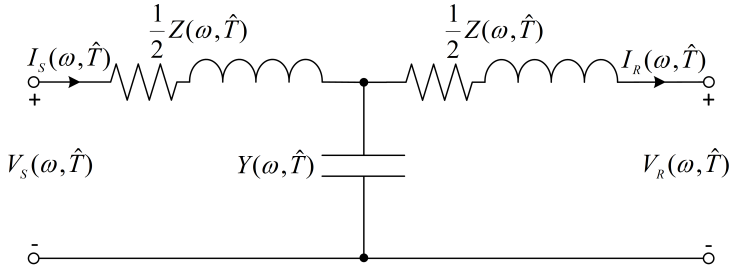


Figure 6.19: Realized Medium-Length Transmission Line (T-Circuit)

As can be seen, these two-port network parameters are not constant anymore. Thus, $V_R(\omega, \breve{T})$ and $I_R(\omega, \breve{T})$ become:

$$V_R(\omega, \breve{T}) = D(\omega, \breve{T})V_S(\omega, \breve{T}) - B(\omega, \breve{T})I_S(\omega, \breve{T}) \tag{6.70}$$

$$I_R(\omega, \breve{T}) = -C(\omega, \breve{T})V_S(\omega, \breve{T}) + A(\omega, \breve{T})I_S(\omega, \breve{T}) \tag{6.71}$$

The admittance $Y(\omega, \breve{T})$, given in (6.66)-(6.69), can be calculated by the same analogy of (6.65):

$$Y(\omega, \breve{T}) \approx \sum_{k=1}^{N} \left[g_k \left(\omega, \breve{T}_k \right) + j\omega c_k \left(\omega, \breve{T}_k \right) \right] \tag{6.72}$$

where g_k and c_k are respectively the shunt conductance and capacitance at the kth zone.

6.4.2.2 Π-Model

The single-line diagram of this realized circuit is shown in Figure 6.20. The expressions of A and D of this circuit are respectively similar to (6.66) and (6.69) of the

T-circuit, while B and C are expressed as follows:

$$B(\omega, \breve{T}) = Z(\omega, \breve{T}) \tag{6.73}$$

$$C(\omega, \breve{T}) = Y(\omega, \breve{T}) + Z(\omega, \breve{T})Y^2(\omega, \breve{T})/4 \tag{6.74}$$

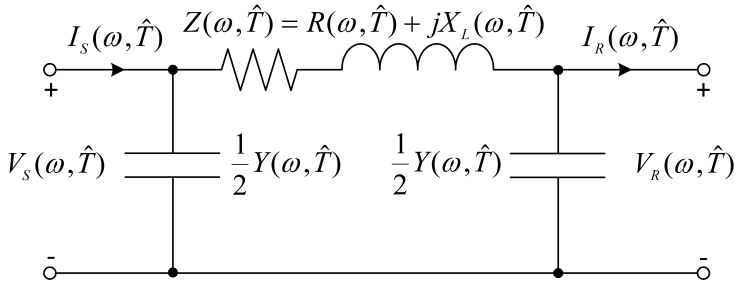


Figure 6.20: Realized Medium-Length Transmission Line (Π-Circuit)

Thus, $V_R(\omega, \breve{T})$ and $I_R(\omega, \breve{T})$ can be respectively calculated using (6.70) and (6.71), but after replacing A, B, C and D with (6.66), (6.73), (6.74) and (6.69), respectively.

6.4.2.3 M-Model

The single-line diagram of this realized circuit is shown in Figure 6.21. For this model, the $\{A, B, C, D\}$ parameters given in (6.12) are realized as follows:

$$A(\omega, \breve{T}) = 1 + \frac{Z(\omega, \breve{T})Y(\omega, \breve{T})}{6}\left[3 + \frac{Z(\omega, \breve{T})Y(\omega, \breve{T})}{6}\right] \tag{6.75}$$

$$B(\omega, \breve{T}) = Z(\omega, \breve{T})\left[1 + \frac{Z(\omega, \breve{T})Y(\omega, \breve{T})}{12}\right] \tag{6.76}$$

$$C(\omega, \breve{T}) = Y(\omega, \breve{T}) + \frac{Z(\omega, \breve{T})Y^2(\omega, \breve{T})}{9}\left[2 + \frac{Z(\omega, \breve{T})Y(\omega, \breve{T})}{12}\right] \tag{6.77}$$

$$D(\omega, \breve{T}) = A(\omega, \breve{T}) \tag{6.78}$$

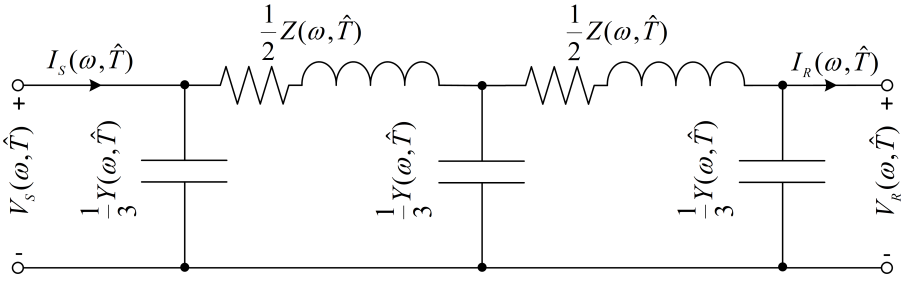


Figure 6.21: Realized Medium-Length Transmission Line (M-Circuit)

6.4.3 Realizing Long-Length Transmission Line and Its Equivalent Medium Line Circuits

For this model, the realized series resistance and inductive reactance and the realized shunt capacitive reactance and conductance should be distributed along the transmission line rather than being lumped as in the preceding models. This realized model is shown in Figure 6.22[17].

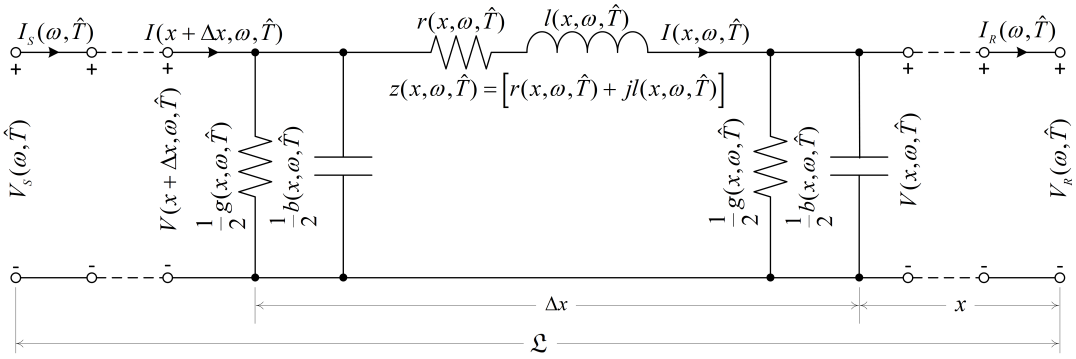


Figure 6.22: Realized Long-Length Transmission Line

The steady-state ABCD parameters given in (6.9) are realized as follows:

$$A(\omega,\breve{T}) = \cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right) \tag{6.79}$$

$$B(\omega,\breve{T}) = Z_c(\omega,\breve{T})\sinh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right) \tag{6.80}$$

$$C(\omega,\breve{T}) = \frac{1}{Z_c(\omega,\breve{T})}\sinh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right) \tag{6.81}$$

$$D(\omega,\breve{T}) = A(\omega,\breve{T}) \tag{6.82}$$

[17]Please note that the distributed parameters are denoted by small letters because the realized per-unit-length parameters are not uniform.

where the propagation constant γ given in (J.58) and the characteristic impedance Z_c given in (J.65) are converted to functions of angular frequency ω and conductor temperature $\breve{T}$. This is one of the important fixes that should be considered for [82, 219][18]. The first approach is to use the following definite integrals:

$$\gamma(\omega,\breve{T}) = \frac{1}{\mathfrak{L}}\int_{\mathfrak{L}_{\min}}^{\mathfrak{L}_{\max}} \sqrt{z\left(x,\omega,\breve{T}\right) y\left(x,\omega,\breve{T}\right)}\, dx \tag{6.83}$$

$$Z_c(\omega,\breve{T}) = \frac{1}{\mathfrak{L}}\int_{\mathfrak{L}_{\min}}^{\mathfrak{L}_{\max}} \sqrt{z\left(x,\omega,\breve{T}\right) / y\left(x,\omega,\breve{T}\right)}\, dx \tag{6.84}$$

For the same reason of replacing (6.64) with (6.65), the preceding very complicated integrations can be replaced with the following approximate equations:

$$\gamma(\omega,\breve{T}) \approx \frac{1}{N}\sum_{k=1}^{N} \sqrt{z_k\left(\omega,\breve{T}_k\right) y_k\left(\omega,\breve{T}_k\right)} \tag{6.85}$$

$$Z_c(\omega,\breve{T}) \approx \frac{1}{N}\sum_{k=1}^{N} \sqrt{z_k\left(\omega,\breve{T}_k\right) / y_k\left(\omega,\breve{T}_k\right)} \tag{6.86}$$

For the equivalent T-, Π- and M-circuits, the long line model is re-expressed as follows:

6.4.3.1 Equivalent T-Circuit

Let's consider the expressions of ABCD parameters given earlier as a two-port network solution of the long-line model. Comparing (6.81) with (6.68) gives:

$$C(\omega,\breve{T}) = \hat{Y}(\omega,\breve{T}) = \frac{1}{Z_c(\omega,\breve{T})}\sinh(\gamma(\omega,\breve{T})\mathfrak{L}) \tag{6.87}$$

Solving the above equation for $\hat{Y}(\omega,\breve{T})$ gives:

$$\hat{Y}(\omega,\breve{T}) = \frac{Y(\omega,\breve{T})\sinh(\gamma(\omega,\breve{T})\mathfrak{L})}{\gamma(\omega,\breve{T})\mathfrak{L}} \tag{6.88}$$

Now, comparing (6.79) with (6.66) gives:

$$A(\omega,\breve{T}) = 1 + \frac{1}{2}\hat{Z}(\omega,\breve{T})\hat{Y}(\omega,\breve{T}) = \cosh(\gamma(\omega,\breve{T})\mathfrak{L}) \tag{6.89}$$

Solving the above equation for $\frac{\hat{Z}(\omega,\breve{T})}{2}$ gives:

$$\frac{\hat{Z}(\omega,\breve{T})}{2} = \frac{Z(\omega,\breve{T})}{2}\cdot\frac{\tanh\left(\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}\right)}{\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}} \tag{6.90}$$

[18] The other weather variables should also be considered for more realized models.

Thus, (6.88) represents the shunt admittance, and (6.90) represents the left and right series impedance shown in Figure 6.19.

6.4.3.2 Equivalent Π-Circuit

A similar approach can be applied here where the line impedance is kept and the shunt admittance is equally divided between the two sides of the line. Comparing (6.80) with (6.73) gives:

$$B(\omega,\breve{T}) = \hat{Z}(\omega,\breve{T}) = Z_c(\omega,\breve{T})\sinh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right) \tag{6.91}$$

Solving the above equation for $\hat{Z}(\omega,\breve{T})$ gives:

$$\hat{Z}(\omega,\breve{T}) = \frac{Z(\omega,\breve{T})\sinh(\gamma(\omega,\breve{T})\mathfrak{L})}{\gamma(\omega,\breve{T})\mathfrak{L}} \tag{6.92}$$

Comparing (6.79) with (6.66) gives (6.89). By taking $Z_c(\omega,\breve{T}) = \frac{1}{Y_c(\omega,\breve{T})} \rightarrow \frac{Z(\omega,\breve{T})}{\gamma(\omega,\breve{T})\mathfrak{L}} = \frac{\gamma(\omega,\breve{T})\mathfrak{L}}{Y(\omega,\breve{T})}$, the last equation can be solved for $\frac{\hat{Y}(\omega,\breve{T})}{2}$ to give:

$$\frac{\hat{Y}(\omega,\breve{T})}{2} = \frac{Y(\omega,\breve{T})}{2}\cdot\frac{\tanh\left(\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}\right)}{\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}} \tag{6.93}$$

Thus, (6.92) represents the series impedance of the line, and (6.93) represents the left and right shunt admittances shown in Figure 6.20.

6.4.3.3 Equivalent M-Circuit

Deriving the equations of this model requires an additional effort with a special care. The derivation of the steady-stage model is given in Appendix M. Alternatively, by eliminating the effects of ω and $\breve{T}$, (6.88) and (6.90) are respectively equivalent to (6.13) and (6.14), and (6.92) and (6.93) are respectively equivalent to (6.15) and (6.16). Thus, by considering the effects of ω and $\breve{T}$, (6.17) and (6.18) become:

$$\frac{\hat{Z}(\omega,\breve{T})}{2} = \frac{-Z(\omega,\breve{T})}{4}\cdot\frac{\tanh\left(\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}\right)}{\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}}\left[1-\sqrt{5+4\cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right)}\right] \tag{6.94}$$

$$\frac{\hat{Y}(\omega,\breve{T})}{3} = \frac{Y(\omega,\breve{T})\coth\left(\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}\right)}{\frac{\gamma(\omega,\breve{T})\mathfrak{L}}{2}}\left[\frac{2-\sqrt{5+4\cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right)}+\cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right)}{3-\sqrt{5+4\cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right)}+2\cosh\left(\gamma(\omega,\breve{T})\mathfrak{L}\right)}\right] \tag{6.95}$$

6.4.4 Transforming the TFB-Models from the Phasor-Domain to the Time-Domain

To see the effects of $\breve{T}$ and frequency ω in the time-domain and along the line, the incident voltage v_1, reflected voltage v_2, incident current i_1 and reflected current i_2 can be calculated using the following mathematical expressions:

$$\begin{aligned} v_1(x,t,\omega,\breve{T}) &= \sqrt{2}e^{\bar{\alpha}(\omega,\breve{T})x}\Big[\psi_1(\omega,\breve{T})\cos\big(\omega t+\bar{\beta}(\omega,\breve{T})x\big) \\ &\qquad -\varphi_1(\omega,\breve{T})\sin\big(\omega t+\bar{\beta}(\omega,\breve{T})x\big)\Big] \quad (6.96) \\ v_2(x,t,\omega,\breve{T}) &= \sqrt{2}e^{-\bar{\alpha}(\omega,\breve{T})x}\Big[\psi_2(\omega,\breve{T})\cos\big(\omega t-\bar{\beta}(\omega,\breve{T})x\big) \\ &\qquad -\varphi_2(\omega,\breve{T})\sin\big(\omega t-\bar{\beta}(\omega,\breve{T})x\big)\Big] \quad (6.97) \\ i_1(x,t,\omega,\breve{T}) &= \sqrt{2}e^{\bar{\alpha}(\omega,\breve{T})x}\Big[\psi_3(\omega,\breve{T})\cos\big(\omega t+\bar{\beta}(\omega,\breve{T})x\big) \\ &\qquad -\varphi_3(\omega,\breve{T})\sin\big(\omega t+\bar{\beta}(\omega,\breve{T})x\big)\Big] \quad (6.98) \\ i_2(x,t,\omega,\breve{T}) &= -\sqrt{2}e^{-\bar{\alpha}(\omega,\breve{T})x}\Big[\psi_4(\omega,\breve{T})\cos\big(\omega t-\bar{\beta}(\omega,\breve{T})x\big) \\ &\qquad -\varphi_4(\omega,\breve{T})\sin\big(\omega t-\bar{\beta}(\omega,\breve{T})x\big)\Big] \quad (6.99) \end{aligned}$$

where the full derivation is given in Appendix O.

6.5 Applying the TFB-Models in Some Fundamental Power System Studies

To show the benefits of our temperature/frequency-based approach, this section provides numerical analysis of some fundamental studies covered in electric power systems engineering. For the sake of simplicity, the frequency effect on the resistance is neglected. Also, the shunt conductance is taken as a constant value.

6.5.1 Seeing the Effects of $\breve{T}$ and ω in the Phasor- and Time-Domains

To see the effects of $\breve{T}$ and ω on transmission lines, in both the phasor-domain and the time-domain, the following experiments are simulated using MATLAB 2016a:

6.5.1.1 Effects of $\breve{T}$ and ω in the Phasor-Domain

The parameters of this experiment are taken as follows: $\mathfrak{L}$ = 130 km, r = 0.036 Ω/km, l = 0.79577 mH/km, c = 0.011194 μF/km, g = 82.407 $p\mho$/km, f = 60 Hz, $V_{S(\text{L-L})}$ = 345 kV, I_S = 400 A, pf_S = 0.95, α_{aluminum} = 0.004308 1/°C (@ 20 °C).

Because the length is 130 km, so it is a medium-line transmission line. However, the long-line model is used to have higher accuracy by considering the negligible conductance. Also, this part of the experiment assumes that all the zone temperatures of the transmission line have the same value. Table 6.4 shows the temperature effect at different operating frequencies. As expected, R is linearly proportional to $\breve{T}$, while X_L and X_C are respectively proportional and inversely proportional to ω. Also, the receiving-end voltage magnitude decreases as $\breve{T} \uparrow$ and/or $\omega \uparrow$, while its angle increases as $\breve{T} \uparrow$ and/or $\omega \downarrow$. The receiving-end current is slightly affected by $\breve{T}$ and unseen in the table, while the effect of ω is 0.1 A for each 0.15 Hz $\rightarrow$ $I_R \propto \omega$. On the opposite side, the current angle θ_{I_R} can be significantly affected by both the conductor temperature $\breve{T}$ and the network frequency $\omega \rightarrow \theta_{I_R} \uparrow$ as $\breve{T} \uparrow$ and/or $\omega \downarrow$. Based on this, the receiving-end power factor decreases as $\breve{T} \uparrow$ and/or $\omega \uparrow$. Although the receiving-end real and reactive power are inversely proportional to $\breve{T}$, $P_R \propto^{-1} \omega$ and $Q_R \propto \omega$. The last column stands for the voltage regulation of the line in %, and it can be calculated as follows [326]:

$$\text{REG} = \frac{\frac{V_{S(\text{L-L})}}{|A(\omega,\breve{T})|} - V_{R(\text{L-L})}}{V_{R(\text{L-L})}} \times 100\% \tag{6.100}$$

From Table 6.4, the transmission line needs more voltage regulation as $\breve{T} \uparrow$ and/or $\omega \uparrow$. All these phenomena are graphically depicted in Figure 6.23.

6.5.1.2 Effects of $\breve{T}$ and ω in the Time-Domain

For this numerical experiment, the following long-line transmission line parameters are used: $\mathfrak{L}$ = 478 km, r = 0.0173 Ω/km, l = 1.0105 mH/km, c = 0.0117 μF/km, g = 78.331 $p\mho$/km, f = 50 Hz, $V_{S(\text{L-L})}$ = 765 kV, I_S = 937 A, pf_S = 0.89, α_{aluminum} = 0.004308 1/°C (@ 20 °C). If the preceding phasor/time-domain transformation is used in this experiment, then (6.96)-(6.99) can be implemented to generate Figure 6.24. This simulation is carried out by considering $\breve{T}$ = 50°C and fixed

Table 6.4: Temperature and Frequency Effects on the Receiving-End of the Transmission Line of the 1st Experiment (Phasor-Domain)

f (Hz)	$\breve{T}$ (°C)	R (Ω)	X_L (Ω)	X_C (MΩ)	$V_{R_{(L\text{-}L)}}$ (kV)	θ_{V_R} (°)	I_R (kA)	θ_{I_R} (°)	PF_R	P_R (MW)	Q_R (MVAR)	REG (%)
59.85	20	4.6800	38.9023	30.8826	330.7662	-4.1731	0.4418	-31.6314	0.8873	224.5952	116.7091	5.4234
	50	5.2848	38.9023	30.8826	330.3583	-4.1458	0.4418	-31.6226	0.8872	224.2753	116.6350	5.5536
	80	5.8897	38.9023	30.8826	329.9504	-4.1185	0.4418	-31.6139	0.8870	223.9554	116.5610	5.6840
60.00	20	4.6800	38.9998	30.8054	330.7317	-4.1844	0.4419	-31.6617	0.8872	224.5945	116.8035	5.4401
	50	5.2848	38.9998	30.8054	330.3238	-4.1571	0.4419	-31.6529	0.8870	224.2746	116.7292	5.5703
	80	5.8897	38.9998	30.8054	329.9159	-4.1297	0.4419	-31.6442	0.8869	223.9546	116.6550	5.7008
60.15	20	4.6800	39.0973	30.7286	330.6972	-4.1956	0.4420	-31.6919	0.8870	224.5939	116.8978	5.4568
	50	5.2848	39.0973	30.7286	330.2892	-4.1683	0.4420	-31.6832	0.8869	224.2738	116.8233	5.5870
	80	5.8897	39.0973	30.7286	329.8813	-4.1410	0.4420	-31.6744	0.8867	223.9538	116.7489	5.7175

along the transmission line, and the measurement is taken from the center-point (i.e., $x = \mathfrak{L}/2 = 478/2 = 239$ km).

6.5.2 Power Flow Analysis

There are some studies reported in the literature that attempted to solve power flow (PF) problems by considering the surrounding weather conditions. For example, by referring to [138], the classical Newton-Raphson (NR) method is modified so the solution can be updated from the jth iteration through the following process:

$$\begin{bmatrix} \delta^{j+1} \\ |V^{j+1}| \\ \breve{T}^{j+1} \end{bmatrix} = \begin{bmatrix} \delta^{j} \\ |V^{j}| \\ \breve{T}^{j} \end{bmatrix} + \mathcal{J}\left(\delta^{j}, \left|V^{j}\right|, \breve{T}^{j}\right)^{-1} \cdot \begin{bmatrix} \Delta P^{j} \\ \Delta Q^{j} \\ \Delta H^{j} \end{bmatrix} \tag{6.101}$$

where $|V|$ and δ are respectively the bus voltage magnitudes and angles, and H is the temperature difference vector of branches (it is described in [138]). The Jacobian matrix is denoted by $\mathcal{J}$, which can be expressed as follows:

$$\mathcal{J}\left(\delta, |V|, \breve{T}\right) = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |V|} & \frac{\partial P}{\partial \breve{T}} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |V|} & \frac{\partial Q}{\partial \breve{T}} \\ \frac{\partial H}{\partial \delta} & \frac{\partial H}{\partial |V|} & \frac{\partial H}{\partial \breve{T}} \end{bmatrix} \tag{6.102}$$

It is clear that the conventional Jacobian matrix is expanded from just four sub-matrices $\{\mathcal{J}_1 \rightarrow \mathcal{J}_4\}$ to nine sub-matrices $\{\mathcal{J}_1 \rightarrow \mathcal{J}_9\}$. The dimension of this modified square matrix is $(\mathfrak{n} + \mathfrak{m} + \mathfrak{h} - 1)$ where $\{\mathfrak{n}, \mathfrak{m}, \mathfrak{h}\}$ stand for the number of buses, the number of PQ buses, and the number of temperature-dependent branches,

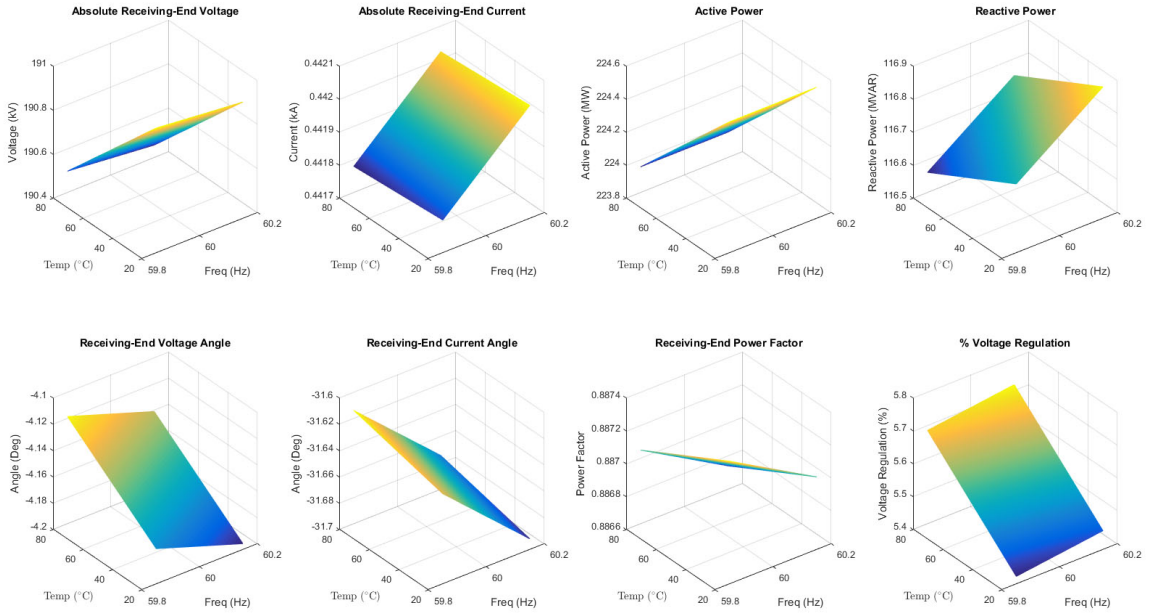


Figure 6.23: 3D Effects of $\breve{T}$ and ω on the Transmission Line of the 1$^{\text{st}}$ Experiment (Phasor-Domain)

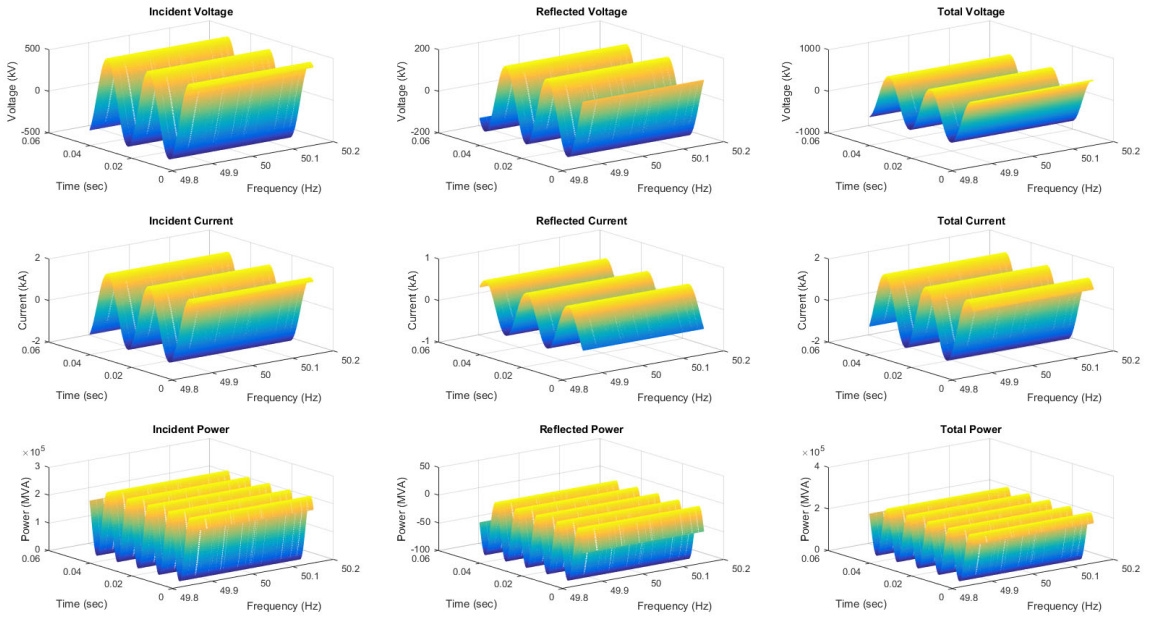


Figure 6.24: 3D Plots of the Incident, Reflected and Total Measurements of v, i and $v \times i$ of the 2$^{\text{nd}}$ Experiment (Time-Domain)

respectively. Because any line could be affected by $\breve{T}$, so $\mathfrak{h}$ is likely to be equal to the number of branches ß, which is really high. This means that the dimension of the modified Jacobian matrix is much larger than the conventional one, and thus the technique proposed in [138] can be considered as a highly time-consuming approach. In that study, the realized resistance has many relaxations, such as the skin effect phenomenon, spiraling and bundle conductor effects. Also, neglecting the effects of $\breve{T}$ and ω on inductance, capacitance, and conductance (which are covered in our TFB technique) could lead to some errors in the final results. Thus, the feasibility and optimality criteria could be affected. More than that, this technique is valid just for solving PF problems, so the researchers are forced to develop their new temperature-based mathematical models for each specific application (PF, ELD, PSS, etc).

Using the TFB technique can preserve the conventional mathematical models of many power system studies where the feasibility, optimality, and processing speed criteria are kept without any change. Therefore, with TFB, the generality of the conventional NR method can be preserved:

$$\begin{bmatrix} \delta^{j+1} \\ |V^{j+1}| \end{bmatrix} = \begin{bmatrix} \delta^{j} \\ |V^{j}| \end{bmatrix} + \mathcal{J}\left(\delta^{j}, |V^{j}|\right)^{-1} \cdot \begin{bmatrix} \Delta P^{j} \\ \Delta Q^{j} \end{bmatrix} \tag{6.103}$$

where the Jacobian matrix is defined as follows:

$$\mathcal{J}\left(\delta, |V|, \breve{T}\right) = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |V|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |V|} \end{bmatrix} \tag{6.104}$$

The only changes here are the elements of the nodal admittance matrix Y_{BUS} because the distributed series and shunt parameters are affected by $\breve{T}$ and ω.

In this section, the conventional NR method is used to solve the PF problem of the IEEE 14-bus test system shown in Figure 6.25. The full information about this system is given in [26]. In PF problems, any one of the short-, medium-, and long-length transmission line models could be used. Based on the information about this test system, there are 15 out of 20 branches are temperature-dependent. Thus, if someone still wants to use (6.101) instead of (6.103), then he/she should deal with a Jacobian matrix of $[37 \times 37]$ (without including the other effects) instead of $[22 \times 22]$ as with the TFB approach.

Now, suppose that the conductors are made of aluminum metal and the resistance is modeled at 20°C during the winter season. The temperature coefficient aluminum is

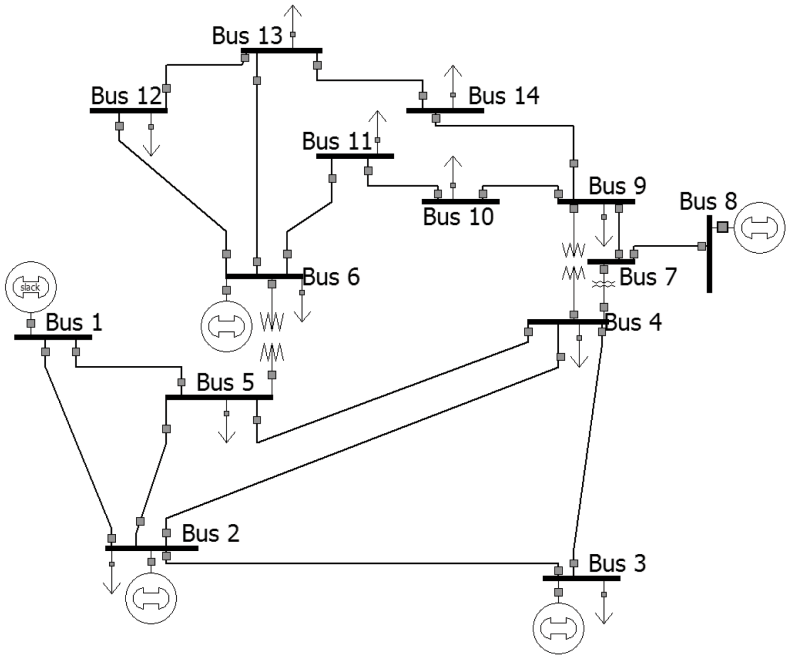


Figure 6.25: Single-Line Diagram of the IEEE 14-Bus Test System

$\alpha_{\text{aluminum}} = 0.004308$ 1/°C. Also, suppose that the peak temperature of the conductors during the summer season is 80°C, and the distortion in the network frequency is 59.85 Hz. For simplicity, and because of the lack of information, $\{\beta, \kappa, \lambda\}$ are taken equal to zero. In this study, four cases are covered as follows:

- Case No.1: $\breve{T} = 20$°C and $f = 60$ Hz $\forall$ N zones[19]
- Case No.2: $\breve{T} = 80$°C and $f = 60$ Hz $\forall$ N zones
- Case No.3: $\breve{T} = 20$°C and $f = 59.85$ Hz $\forall$ N zones
- Case No.4: $\breve{T} = 80$°C and $f = 59.85$ Hz $\forall$ N zones

Tables 6.5-6.8 show the PF results of the NR method without/with implementing the TFB-model. It is obvious that the voltage magnitude $|V|$ of the PQ buses remain close to 1 p.u. for all the cases. However, they are slightly affected by $\breve{T}$ and ω. Similar thing for the voltage angle δ_V of both the PQ and the PV buses. The changes in Y_{BUS} due to $\breve{T}$ and ω also affect the final settings of the five generators and synchronous compensators connected to the network. For the real power, it can be observed that the slack generator compensates for the losses in the network due to the change in

[19] i.e., the normal case without employing TFB.

Table 6.5: Power Flow Results Obtained by NR for Case No.1 (Default)

Bus No.	**Type**	$\|\mathbf{V}\|^a$	$\delta_{\mathbf{V}}$	$\mathbf{P_L}$	$\mathbf{Q_L}$	$\mathbf{P_G}$	$\mathbf{Q_G}$
1	Slack	1.06000	0.00			232.39	-16.55
2	PV	1.04500	-4.98	21.70	12.70	40.00	43.56
3	PV	1.01000	-12.73	94.20	19.00	0.00	25.07
4	PQ	1.01767	-10.31	47.80	-3.90		
5	PQ	1.01951	-8.77	7.60	1.60		
6	PV	1.07000	-14.22	11.20	7.50	0.00	12.73
7	PQ	1.06152	-13.36				
8	PV	1.09000	-13.36			0.00	17.62
9	PQ	1.05593	-14.94	29.50	16.60		
10	PQ	1.05099	-15.10	9.00	5.80		
11	PQ	1.05691	-14.79	3.50	1.80		
12	PQ	1.05519	-15.08	6.10	1.60		
13	PQ	1.05038	-15.16	13.50	5.80		
14	PQ	1.03553	-16.03	14.90	5.00		

[a] $|V|$ in per-unit, δ_V in degrees, P in MW, and Q in MVAr.

$\breve{T}$. This phenomenon can be clearly concluded by comparing $P_{G,1}$ of Tables 6.5-6.7. On the opposite side, the reactive power Q_G of all the machines are highly affected by $\breve{T}$ and slightly affected by ω. However, this is not a final judgment, because the step-change in $\breve{T}$ is higher than that in ω. In contrast, R are much smaller than X_L.

Before ending this section, someone may argue about the feasibility of counting these changes. For example, the contingency analysis can sacrifice the solution quality to have faster algorithms, such as DC and AC-DC PF algorithms. It is partially correct for that specific application, but what about the other applications and studies where high-quality PF solutions are required, such as fault analysis (FA), system stability, automation and control, and relays coordination.

6.5.3 Network Power Losses

This study has many applications in electric power systems, such as OPF, ELD, UC, performance monitoring of transmission lines, materials and superconductivity, planning and future expansions, etc. Starting from complex power loss S_L, this quantity is split into two parts: active P_L and reactive Q_L.

Table 6.6: Power Flow Results Obtained by NR for Case No.2

Bus No.	**Type**	$\|\mathbf{V}\|$	$\delta_{\mathbf{V}}$	$\mathbf{P_L}$	$\mathbf{Q_L}$	$\mathbf{P_G}$	$\mathbf{Q_G}$
1	Slack	1.06000	0.00			236.36	-33.19
2	PV	1.04500	-5.27	21.70	12.70	40.00	49.56
3	PV	1.01000	-13.28	94.20	19.00	0.00	34.45
4	PQ	1.01361	-10.68	47.80	-3.90		
5	PQ	1.01610	-9.07	7.60	1.60		
6	PV	1.07000	-14.59	11.20	7.50	0.00	14.34
7	PQ	1.05927	-13.73				
8	PV	1.09000	-13.73			0.00	19.02
9	PQ	1.05327	-15.30	29.50	16.60		
10	PQ	1.04800	-15.44	9.00	5.80		
11	PQ	1.05493	-15.13	3.50	1.80		
12	PQ	1.05296	-15.41	6.10	1.60		
13	PQ	1.04736	-15.47	13.50	5.80		
14	PQ	1.02992	-16.34	14.90	5.00		

Table 6.7: Power Flow Results Obtained by NR for Case No.3

Bus No.	**Type**	$\|\mathbf{V}\|$	$\delta_{\mathbf{V}}$	$\mathbf{P_L}$	$\mathbf{Q_L}$	$\mathbf{P_G}$	$\mathbf{Q_G}$
1	Slack	1.06000	0.00			232.39	-16.63
2	PV	1.04500	-4.97	21.70	12.70	40.00	43.58
3	PV	1.01000	-12.69	94.20	19.00	0.00	25.07
4	PQ	1.01768	-10.29	47.80	-3.90		
5	PQ	1.01953	-8.75	7.60	1.60		
6	PV	1.07000	-14.19	11.20	7.50	0.00	12.69
7	PQ	1.06154	-13.33				
8	PV	1.09000	-13.33			0.00	17.66
9	PQ	1.05596	-14.90	29.50	16.60		
10	PQ	1.05102	-15.06	9.00	5.80		
11	PQ	1.05693	-14.75	3.50	1.80		
12	PQ	1.05521	-15.04	6.10	1.60		
13	PQ	1.05041	-15.12	13.50	5.80		
14	PQ	1.03558	-15.99	14.90	5.00		

Table 6.8: Power Flow Results Obtained by NR for Case No.4

Bus No.	**Type**	$\|\mathbf{V}\|$	$\delta_{\mathbf{V}}$	$\mathbf{P_L}$	$\mathbf{Q_L}$	$\mathbf{P_G}$	$\mathbf{Q_G}$
1	Slack	1.06000	0.00			236.36	-33.31
2	PV	1.04500	-5.26	21.70	12.70	40.00	49.60
3	PV	1.01000	-13.25	94.20	19.00	0.00	34.47
4	PQ	1.01363	-10.65	47.80	-3.90		
5	PQ	1.01612	-9.05	7.60	1.60		
6	PV	1.07000	-14.55	11.20	7.50	0.00	14.30
7	PQ	1.05929	-13.69				
8	PV	1.09000	-13.69			0.00	19.05
9	PQ	1.05330	-15.27	29.50	16.60		
10	PQ	1.04803	-15.40	9.00	5.80		
11	PQ	1.05495	-15.09	3.50	1.80		
12	PQ	1.05297	-15.37	6.10	1.60		
13	PQ	1.04739	-15.43	13.50	5.80		
14	PQ	1.02997	-16.30	14.90	5.00		

Table 6.9: Reactive Power Losses Across the Transformers of the IEEE 14-Bus System

Case No[a]	$\breve{T}$ (° C)	f (Hz)	$Q_L^{\text{Bus 4-7}}$	$Q_L^{\text{Bus 4-9}}$	$Q_L^{\text{Bus 5-6}}$
1	20	60	1.703	1.305	4.421
2	80	60	1.736	1.301	4.446
3	20	59.85	1.700	1.301	4.412
4	80	59.85	1.733	1.297	4.437

[a] $Q_L^{\text{Bus i-j}}$ in MVAr.

6.5.3.1 Losses Across Transformers

In transformers, Q_L is absorbed by shunt reactance as "magnetizing" and by series reactance as "leakage flux", while its resistance absorbs P_L as "heat". In the literature, there are some mathematical equations that can be used, with considering $\breve{T}$ and ω effects, to calculate P_L and Q_L. For online monitoring, P_L and Q_L can be easily calculated by subtracting power outputs $\{P_{\text{out}}, Q_{\text{out}}\}$ from inputs $\{P_{\text{in}}, Q_{\text{in}}\}$:

$$P_L = P_{\text{in}} - P_{\text{out}} \tag{6.105}$$

$$Q_L = Q_{\text{in}} - Q_{\text{out}} \tag{6.106}$$

With considering just series reactance (as with the IEEE 14-bus test system), Q_L

of the three transformers shown in Figure 6.25 are tabulated in Table 6.9 for the four cases mentioned before. From the preceding table, it can be noticed that the losses of the 4^{th} case can be predicted by knowing the normal losses and the losses of each individual effect as follows:

$$Q_L^{\breve{T},f} = Q_L^{\breve{T}_0,f} + Q_L^{\breve{T},f_0} - Q_L^{\breve{T}_0,f_0} \tag{6.107}$$

But, this is not a general formula. It might not be valid for other test systems, especially if the other transformer parameters and the skin effect phenomenon are not neglected. More research is required to clarify this point.

6.5.3.2 Losses Across Lines

A similar thing could happen in lines where P_L and Q_L can be calculated from their on-line power flow data. However, if the TFB-model is implemented, then these actual losses can be estimated with fewer errors; which also leads to state estimation applications. In the literature, many methods are used to estimate line losses where each one has its own strengths and weaknesses. For example, the exact transmission loss formula (or George's equation) for the line between the ith bus and the jth bus (for all the network buses $\rightarrow$ "NB") are given below:

$$P_L = \sum_{i=1}^{\text{NB}} \sum_{j=1}^{\text{NB}} P_i \hat{B}_{ij} P_j \tag{6.108}$$

$$Q_L = \sum_{i=1}^{\text{NB}} \sum_{j=1}^{\text{NB}} Q_i \hat{C}_{ij} Q_j \tag{6.109}$$

where $\{P_i, P_j, Q_i, Q_j\}$ are the net real and reactive power "injected to" or "absorbed from" the ith and jth buses, respectively. $\hat{B}_{ij}$ and $\hat{C}_{ij}$ are respectively called the $\hat{B}$- and $\hat{C}$-coefficients[20]. There are many ways to determine these coefficients by using current or power quantities [210]. For example, they can be obtained as follows:

$$\hat{B}_{ij} = \frac{R_{ij}}{|V_i|\,|V_j|} \cdot \frac{\cos(\theta_i - \theta_j)}{\cos(\phi_i)\cos(\phi_j)} \tag{6.110}$$

$$\hat{C}_{ij} = \frac{X_{ij}}{|V_i|\,|V_j|} \cdot \frac{\cos(\theta_i - \theta_j)}{\sin(\phi_i)\sin(\phi_j)} \tag{6.111}$$

[20]The hat accents are used in this chapter just to differentiate between them and the ABCD parameters.

Table 6.10: $\breve{T}$ and ω Effects on the PV Buses of the WSCC 9-Bus Test System

PV Bus	Case No.1		Case No.2		Case No.3		Case No.4	
No[a]	$\|V\|$	δ_V	$\|V\|$	δ_V	$\|V\|$	δ_V	$\|V\|$	δ_V
2	18.450	9.351	18.450	9.349	18.450	9.329	18.450	9.327
3	14.145	5.142	14.145	5.133	14.145	5.130	14.145	5.121

[a] $|V|$ in kV and δ_V in degrees.

where θ_i and θ_j are respectively the current angles at the ith and jth buses. R_{ij} and X_{ij} are respectively the resistance and reactance of the i-jth line. ϕ_i and ϕ_j are respectively the power angles at the ith and jth buses, and they can be calculated as:

$$\phi_i = \tan^{-1}\left(\frac{Q_i}{P_i}\right) \tag{6.112}$$

$$\phi_j = \tan^{-1}\left(\frac{Q_j}{P_j}\right) \tag{6.113}$$

Because the current angles are not known during solving PF problems, so they can be calculated based on the known voltage angles from the PF stage and the known power angles from (6.112) and (6.113) as follows:

$$\theta_i = \delta_i - \phi_i \tag{6.114}$$

$$\theta_j = \delta_j - \phi_j \tag{6.115}$$

That is, the effects of $\breve{T}$ and f should be accounted in the PF stage, and the solution obtained from that stage can be utilized to find $\hat{B}$- and $\hat{C}$-coefficients. Thus, the whole process is affected by the conductor temperatures $\breve{T}$ and the network frequency f. Although these innovative ways can accelerate getting very good solutions without consuming extra CPU time, they are based on some assumptions, such as the generators are operated with fixed power factors and their busbars have constant voltage magnitudes and angles [123, 210]. To validate these two assumptions, the two PV buses of the **Western System Coordinating Council** (**WSCC**) 9-bus test system shown in Figure 6.26 are analyzed. The full information about this test system is given in [26]. Table 6.10 gives a clear picture of the effects of $\breve{T}$ and ω. It can be seen that the voltage magnitudes of PV buses are not sensitive to both $\breve{T}$ and ω because they are fixed during the PF stage. However, their corresponding voltage angles are affected by both $\breve{T}$ and ω. Thus, there are two common approaches to deal

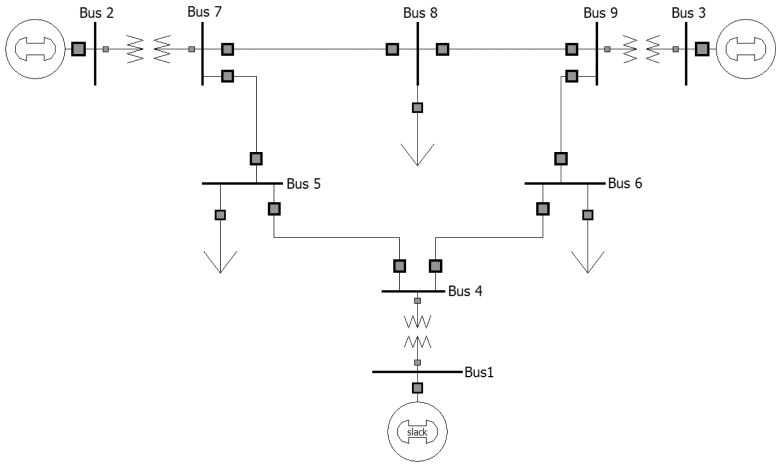


Figure 6.26: Single-Line Diagram of the WSCC 9-Bus Test System

Table 6.11: $\breve{T}$ and ω Effects on the Losses of the WSCC 9-Bus System

Branch[a]		**Case No.1**		**Case No.2**		**Case No.3**		**Case No.4**	
From	To	P_L	Q_L	P_L	Q_L	P_L	Q_L	P_L	Q_L
1	4	0.000	3.147	0.000	3.330	0.000	3.144	0.000	3.327
2	7	0.000	15.820	0.000	15.812	0.000	15.780	0.000	15.773
3	9	0.000	4.103	0.000	4.112	0.000	4.092	0.000	4.101
4	5	0.280	-16.142	0.372	-15.940	0.280	-16.098	0.373	-15.896
4	6	0.145	-15.616	0.194	-15.475	0.145	-15.574	0.194	-15.433
5	7	2.160	-20.556	2.706	-20.519	2.160	-20.501	2.707	-20.464
6	9	1.464	-30.904	1.834	-30.807	1.465	-30.821	1.834	-30.724
7	8	0.505	-12.141	0.635	-12.153	0.505	-12.109	0.635	-12.121
8	9	0.073	-21.342	0.093	-21.326	0.073	-21.285	0.094	-21.270
Total		4.627	-93.631	5.834	-92.966	4.628	-93.372	5.837	-92.707

[a] P_L in MW and Q_L in MVAr.

with this accuracy issue, either using highly accurate transmission loss formulas or using online loss calculations. The first one still is a dialectical point in the literature, while the other one cannot be used for many power system studies, such as fault and contingency analysis. The other approach is to extract the estimated losses from the static PF stage, but after being modified by the TFB-model[21]. Table 6.11 shows the PF losses in the branches of Figure 6.26 for all the four cases.

It is important to note that there are some power losses in other power system

[21] AI can also be used to estimate power losses regardless of the assumptions. For example, Chapter 9 presents one method based on ANNs.

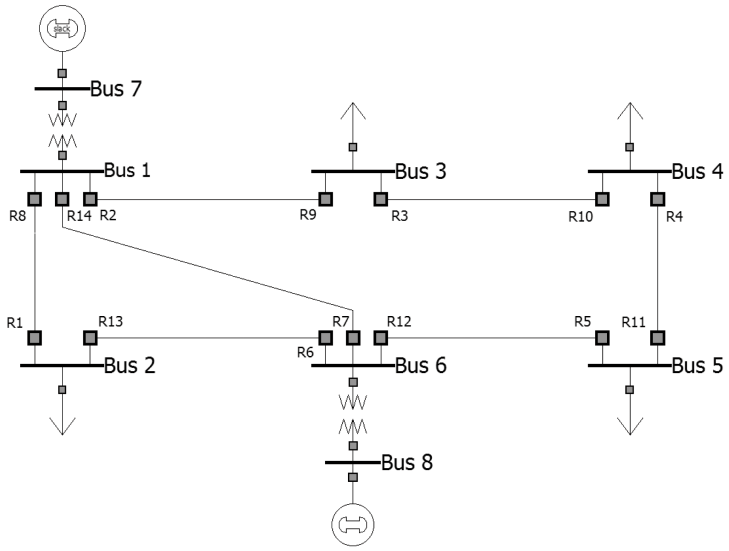


Figure 6.27: Single-Line Diagram of the IEEE 8-Bus Test System

components (including loads, generators, battery banks, etc), which might be considered in future work. Further discussion about P_L is covered in the next section.

6.5.4 Short-Circuit Analysis

It has been found that $\breve{T}$ and ω could have significant effects on the power flow analysis. The PF stage is considered as a backbone of many power system studies [124, 326]. One of the well-known studies is fault analysis (FA). As a rule of thumb, changing the impedance of any circuit means changing the current flowing through that circuit. This is why the fault current magnitude is proportional to the fault location; which is also one of the prospective applications of the TFB-model. To prove the significant effects of $\breve{T}$ and ω on the FA stage, the IEEE 8-bus test system shown in Figure 6.27 is analyzed by measuring the 3-phase (3ϕ) short-circuit current near each DOCR indicated in that network. The full information about this test system is given in [25]. Table 6.12 shows the short-circuit current I_{sc} seen by all primary/backup (P/B) relay pairs, after being stepped down by their CTs, where R^{pr} and R^{bc} mean the primary and backup DOCRs, respectively. It has been found that this stage can be slightly affected by ω. However, this observation cannot be generalized because this test system is small and mainly used to optimize the relay settings [280]. On the opposite side, $\breve{T}$ can deviate I_{sc} more than 10 A in some

Table 6.12: 3ϕ Faults of the IEEE 8-Bus Test System for Different Cases of $\breve{T}$ and ω

P/B DOCR Pairs			**Case No.1'** "I_{sc} (A)"		**Case No.2** "I_{sc} (A)"		**Case No.3** "I_{sc} (A)"		**Case No.4** "I_{sc} (A)"	
Primary	Backup	CTR	R^{pr}	R^{bc}	R^{pr}	R^{bc}	R^{pr}	R^{bc}	R^{pr}	R^{bc}
R_1	R_6	1200:5	1025.31	1025.31	1022.57	1022.57	1025.65	1025.65	1022.9	1022.9
R_2	R_1	1200:5	1394.57	446.33	1397.6	444.52	1394.89	446.39	1397.89	444.58
R_2	R_7	1200:5	1394.57	897.37	1397.6	893.21	1394.89	897.48	1397.89	893.3
R_3	R_2	800:5	803.6	803.6	808.24	808.24	803.99	803.99	808.64	808.64
R_4	R_3	1200:5	1664.99	531.79	1656.58	536.25	1664.95	532.04	1656.53	536.53
R_5	R_4	1200:5	1248.84	1248.84	1240.22	1240.22	1248.92	1248.92	1240.28	1240.28
R_6	R_5	1200:5	1797.39	934.45	1794.27	926.69	1797.71	934.56	1794.5	926.78
R_6	R_{14}	1200:5	1797.39	589.09	1794.27	591.77	1797.71	589.25	1794.5	591.94
R_7	R_5	800:5	1686.73	934.45	1680.81	926.69	1686.95	934.56	1680.96	926.78
R_7	R_{13}	800:5	1686.73	293.15	1680.81	294.66	1686.95	293.24	1680.96	294.75
R_8	R_7	1200:5	1568.21	897.37	1566.07	893.21	1568.56	897.48	1566.44	893.3
R_8	R_9	1200:5	1568.21	825.24	1566.07	820.25	1568.56	825.37	1566.44	820.38
R_9	R_{10}	800:5	1315.31	1315.31	1310.35	1310.35	1315.53	1315.53	1310.56	1310.56
R_{10}	R_{11}	1200:5	1865.82	601.26	1861.59	608.11	1865.83	601.62	1861.62	608.48
R_{11}	R_{12}	1200:5	898.86	898.86	908.4	908.4	899.29	899.29	908.86	908.86
R_{12}	R_{13}	1200:5	1256.05	293.15	1267.01	294.66	1256.39	293.24	1267.43	294.75
R_{12}	R_{14}	1200:5	1256.05	589.09	1267.01	591.77	1256.39	589.25	1267.43	591.94
R_{13}	R_8	1200:5	824.81	824.81	824.55	824.55	825.16	825.16	824.91	824.91
R_{14}	R_1	800:5	1291.36	446.33	1294.34	444.52	1291.64	446.39	1294.67	444.58
R_{14}	R_9	800:5	1291.36	825.24	1294.34	820.25	1291.64	825.37	1294.67	820.38

branches, such as that of R_{12} (when it acts as a primary DOCR). Although 10 A could be a very small deviation, compared with values above 1 kA, neglecting that effect could lead to some violations or errors in fault locations and the ORC problems.

6.5.5 Transient Stability

To have a brief idea about the effects of $\breve{T}$ and ω on the stability of electric power systems, the single machine infinite bus (SMIB) problem shown in Figure 6.28 is given as a simple example of transient stability. Many classical SMIB problems are modeled based on an assumption that all the electric power components (including generators, transformers, and lines) are lossless, so the active power delivered by that machine P_G can be calculated by using the following popular equation:

$$P_G = \frac{|E_G|\ |E_\infty|}{X_{\text{tot}}} \sin \delta_G \tag{6.116}$$

where E_G and E_∞ are respectively the generator internal voltage and the infinite bus voltage, and X_{tot} is the total inductive reactance.

By knowing the terminal voltage V_t and the current I_G delivered by that generator,

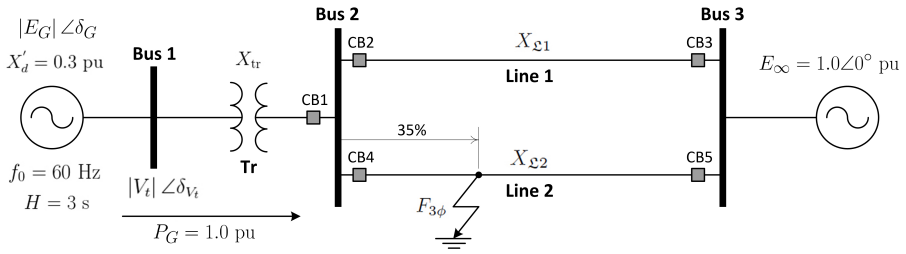


Figure 6.28: Classical Single Machine Infinite Bus (SMIB)

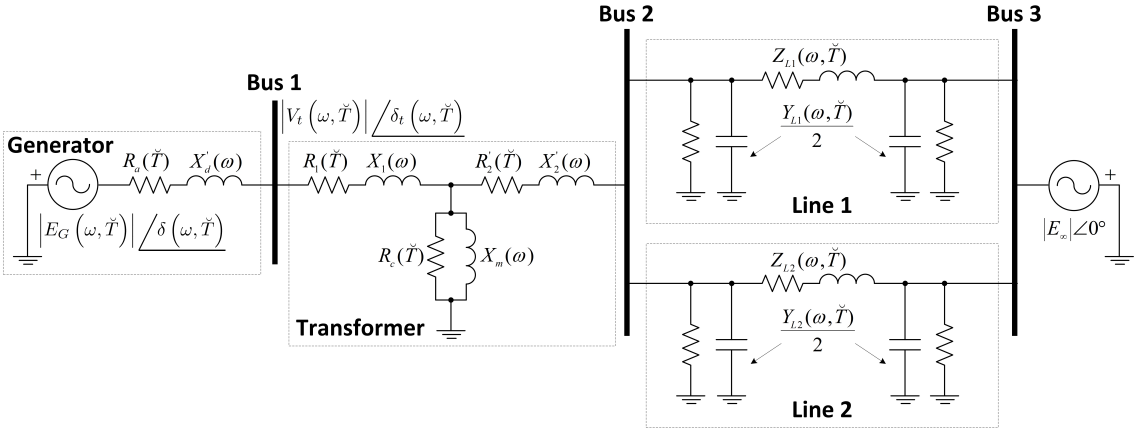


Figure 6.29: Single Machine Infinite Bus (SMIB) Modeled with TFB

the following equation can be used to calculate E_G:

$$E_G = V_t + jI_G X_d' \tag{6.117}$$

If the TFB-model is considered during solving that SMIB problem, then all the series impedances and shunt admittances should be considered. Thus, the simple circuit given in Figure 6.28 can be realized to be as that shown in Figure 6.29. The realistic transformer T-circuit shown in Figure 6.29 can be simplified by shifting the magnetizing elements to any side (i.e., to have a Γ- or Ꟊ-circuit) and then finding the equivalent series impedance. By employing the realized transmission line parameters of the TFB-model, the effects of $\breve{T}$ and ω can also be included[22] in the two Π-circuits

[22]Because the infinite bus represents an external network, so the system frequency of the generator, transformer, and parallel lines is fixed. The doubt here is when there is a sudden shortage of power in that external network. For example, losing one power station, having a trip of a large capacity thermal unit, or shedding/adding a large load element. Thus, the system frequency will fluctuate. Of course, the analysis is done by fixing that frequency during the simulation period. But, the constant value of that frequency could be 60 Hz or any value within the permissible limits, which is taken here as $f = 59.85$ Hz.

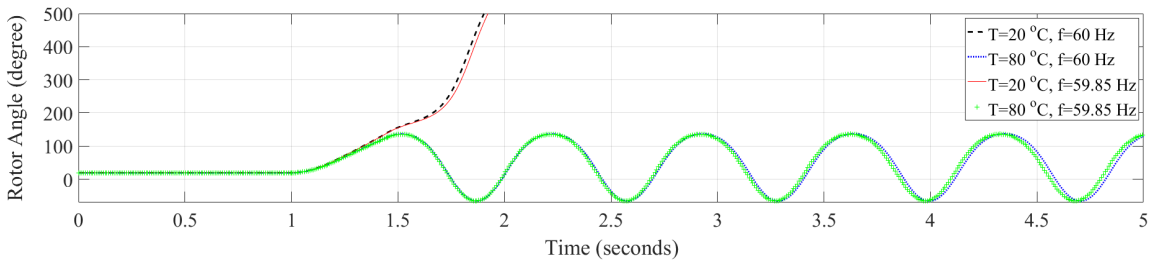


Figure 6.30: Simulation Results of the Generator Rotor Angle of the SMIB Problem When T_{clear}=460 Millie-Seconds

shown in Figure 6.29. It has to be noted that the transformer has different design and materials than that of transmission lines, so different resistance temperature coefficients αs should be used [138]. In addition, (6.116)-(6.117) are not valid with the TFB-model anymore, and hence they should be re-derived again based on these non-neglected elements (i.e., generator resistance, transformer resistance and magnetizing conductance/susceptance, line series resistance, and line shunt charging capacitance and conductance). Now, let's simulate the SMIB network shown in Figure 6.29 with the following parameters:

- ***Generator:*** H=3 s (*inertia constant*), $D = 0$ (*damping*), R_a=0.006 pu (*stator resistance*), and $X_d^{'}$=0.3 pu (*direct axis transient reactance*).
- ***Transformer:*** $R_{\text{tr}}^{\text{eq}}$=0.002777 pu (*series resistance*), $X_{\text{tr}}^{\text{eq}}$=0.07675 pu (*series reactance*), G_c=0.000774 pu (*magnetizing conductance*), and B_m=-0.00090 pu (*magnetizing susceptance*).
- ***Transmission Lines*** *(identical parallel lines):* R=0.04701 pu (*lumped series resistance*), X= 0.17103 pu (*lumped series reactance*), B=0.012800 pu (*lumped shunt charging*), and G=0.000095 pu (*shunt conductance*).

The classical GENCLS model is used to simulate the above SMIB problem with a bolted fault made on line 2 and away from bus 2 by 35%. The faulty line is completely isolated when the fault period reaches T_{clear}[23]. This can be done by opening the circuit breakers CB4 and CB5 shown in Figure 6.28. That fault is cleared after 0.46 seconds of its inception, and then the faulty line is returned into service after waiting 30

[23]i.e., the time required to open both end circuit breakers.

Table 6.13: $\breve{T}$ and ω Effects on T_{cr}, $|V_t|$ and δ_{V_t} of the SMIB Problem

Variable	**Case No.1**	**Case No.2**	**Case No.3**	**Case No.4**
T_{cr} (ms)	416.1465	478.4825	424.4994	493.0005
$\|V_t\|$ (pu)	1.100327	1.106437	1.100172	1.106283
δ_{V_t} (deg)	7.659409	7.410749	7.639265	7.390706
Additional Chance	—	14.979%	2.007%	18.468%

seconds. To preserve the system stability, it has to ensure that the clearing time T_{clear} should not exceed the critical time T_{cr}[24]. Table 6.13 shows the values of T_{cr}, $|V_t|$ and δ_{V_t} to remain the system stable within a 60 seconds frame; the time required to complete the simulation. These values are recorded for all the cases of $\breve{T}$ and ω. Now, let's set the clearing time to T_{clear}=460 ms to clear that fault for all the cases. Figure 6.30 shows the behavior of the generator rotor angle within five seconds. As expected, because the critical clearing time of the 1$^{\text{st}}$ and 3$^{\text{rd}}$ cases are less than 460 ms, so their stability cannot be preserved within that opening speed of CB4 and CB5.

Tables 6.12-6.13 and Figure 6.30 give a solid conclusion that there are possibilities to have significant impacts of $\breve{T}$ and ω on the protection stage. This phenomenon is covered in Chapter 8 for one of the realized optimal relay coordination studies.

6.6 Modeling the M-Circuit for Transmission Lines with Sag

In the previous sections, it has been seen that the distributed parameters of transmission lines are subject to many weather and system variables. These dynamic changes make cables to sag as illustrated in Figure 6.31. The innovative M-model shown in Figure 6.8 can be applied here to effectively account for these changes by just varying the two series impedances and the admittance at the center, while the admittances at the sending-end and receiving-end terminals are kept at their nominal values without any change. The new M-circuit is shown in Figure 6.32. To do that, there are two possible scenarios:

[24] i.e., the maximum allowable chance to clear the fault before losing the synchronism.

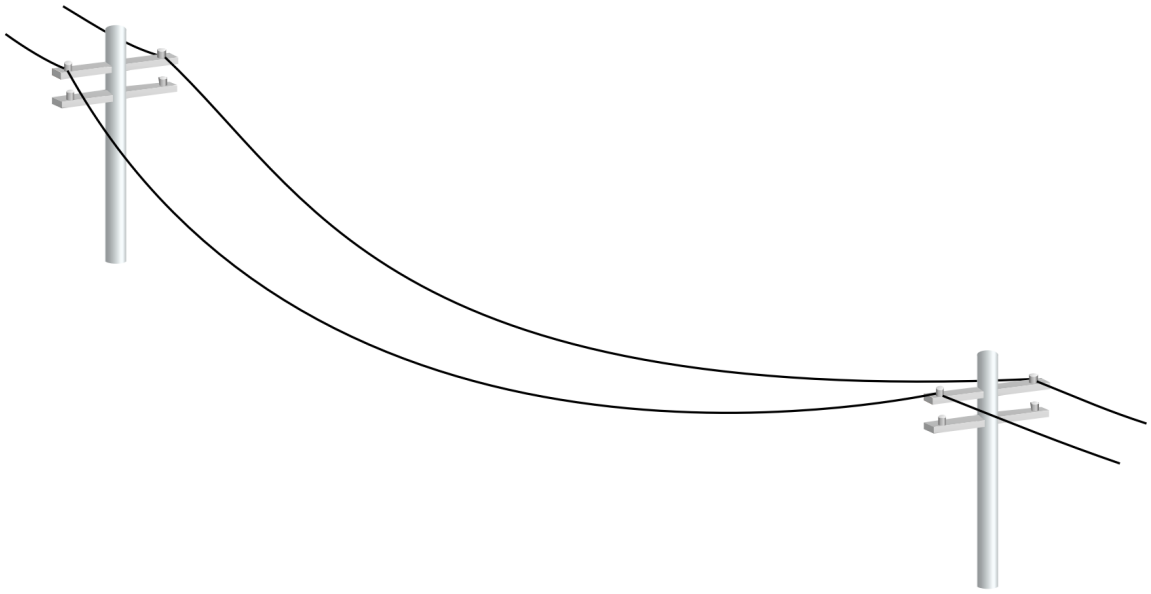


Figure 6.31: Sag in Overhead Transmission Lines

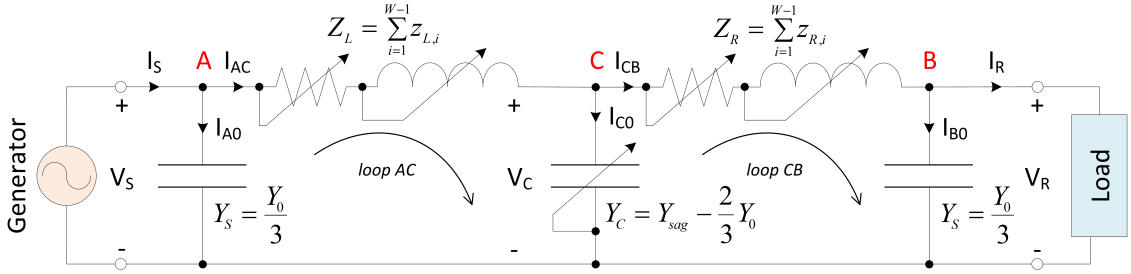


Figure 6.32: Modified M-Model to Account the Sag Effects

6.6.1 Inclined-Spans Scenario

This is the hardest scenario where the height of each two towers are not equal or the earth's surface is not flat[25]. However, the model created with inclined-spans is more general and thus it can be used as a basis to find the other models.

The scenario of inclined-spans due to unequal tower heights is illustrated in Figure 6.33. As can be seen from that illustration, there are W towers with equal spans and unequal sags. If the M-circuit shown in Figure 6.32 is applied, then there are two approaches to model the left impedance Z_L and the right impedance Z_R:

[25]There is also one other possible situation that acts like inclined-spans, which happens when there is a huge amount of snow, ice or sand accumulated on the ground. This situation frequently happens in very cold and hot countries.

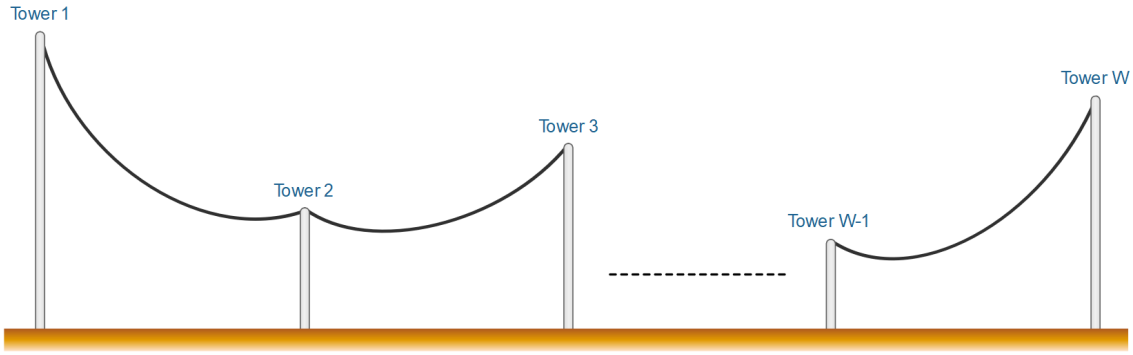


Figure 6.33: Inclined-Spans in Overhead Transmission Lines (Unequal Tower Heights)

- The two impedances are equal:

$$Z_L = Z_R = \frac{1}{2}\sum_{i=1}^{W-1} z_i \tag{6.118}$$

  where W is the number of towers installed between the sending-end and the receiving-end busbars, and z_i is the total impedance of the cable connected between the ith and $(i+1)$th towers.

- The two impedances are unequal:

$$Z_L = \sum_{i=1}^{W-1} z_{L,i} \tag{6.119}$$

$$Z_R = \sum_{i=1}^{W-1} z_{R,i} \tag{6.120}$$

  where $z_{L,i}$ is the impedance between the ith tower and the sag point, and $z_{R,i}$ is the impedance between the sag point and the $(i+1)$th tower.

The M-model of the first approach is similar to the M-model of the leveled-spans scenario, which is easy to solve. The second approach is more general and thus it is explained first. The two-port network of this approach can be established by the following matrix elements:

$$A = 1 + Z_L Y_C + Y_S \left(Z_L + Z_R + Z_L Z_R Y_C\right) \tag{6.121}$$

$$B = Z_L + Z_R + Z_L Z_R Y_C \tag{6.122}$$

$$C = Y_C + Y_S \left[2 + Y_C \left(Z_L + Z_R\right)\right] + Y_S^2 \left(Z_L + Z_R + Z_L Z_R Y_C\right) \tag{6.123}$$

$$D = A \tag{6.124}$$

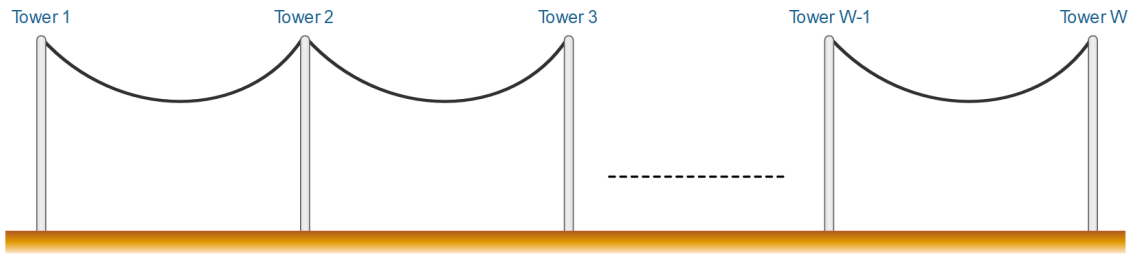


Figure 6.34: Leveled-Spans in Overhead Transmission Lines

The full derivation is given in Appendix P. The ABCD parameters of the nominal M-model given in (6.12) can be easily retrieved from (6.121)-(6.124) by taking $Z_L = Z_R = Z_0/2$ and $Y_C = Y_S = Y_0/3$. It has to be noted that the series impedance Z and the shunt admittance Y of the nominal M-model shown in Figure 6.8 are taken here as Z_0 and Y_0; i.e., as pre-sag lumped parameters, which are shown in Figure 6.32.

6.6.2 Leveled-Spans Scenario

This scenario is illustrated in Figure 6.34. It happens when all the towers have the same height with the same **span** and all the conductors are subject to the same operating condition (i.e., metrology data, system frequency, load current, and cable design/status/age).

Because the impedance between the ith tower and the sag point is equal to the impedance between that point and the $(i+1)$th tower, so Z_L and Z_R given in (6.119)-(6.120) are equal and similar to (6.118). Again, the variation in the series impedance is compensated by these two arm impedances, and the variation in the shunt admittance is compensated by the center element Y_C. The two-end admittances are kept at their nominal values; i.e. $Y_S = Y_0/3$.

The two-port network of this scenario can be easily obtained by equating Z_L and Z_R. For not messing up, let's call them $Z_L = Z_R = Z_V$. Thus, replacing Z_L and Z_R with Z_V in (6.121)-(6.124) yields:

$$A = 1 + Z_V\left(Y_C + 2Y_S + Z_V Y_C Y_S\right) \tag{6.125}$$

$$B = Z_V\left(2 + Z_V Y_C\right) \tag{6.126}$$

$$C = \left(1 + Z_V Y_S\right)\left[Y_C + Y_S\left(2 + Z_V Y_C\right)\right] \tag{6.127}$$

$$D = A \tag{6.128}$$

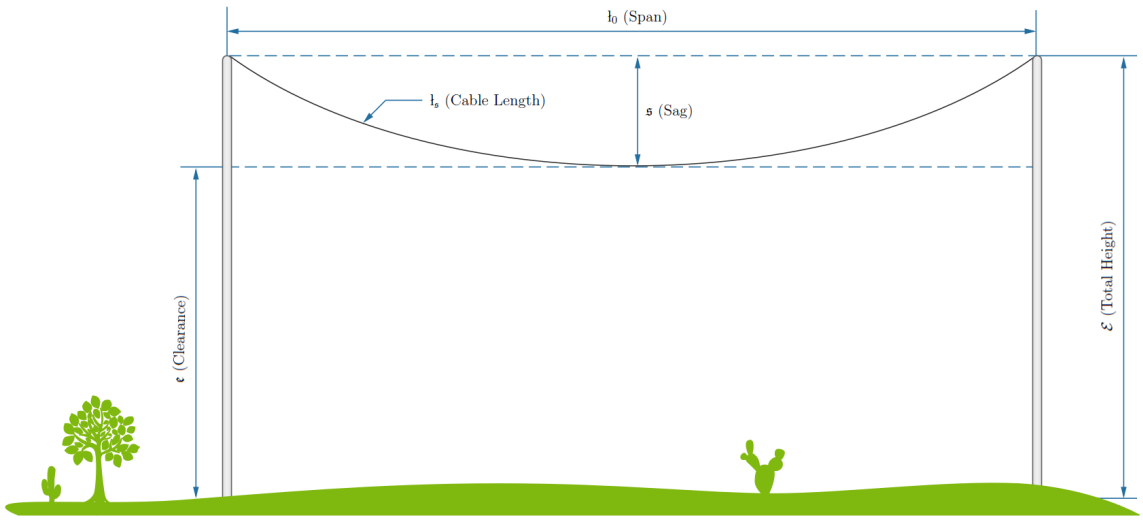


Figure 6.35: Cable Connected Between Two Identical Towers (Leveled-Spans)

Again, the ABCD parameters of the nominal M-model given in (6.12) can be easily retrieved from (6.125)-(6.128) by taking $Z_V = Z_0/2$ and $Y_C = Y_S = Y_0/3$.

6.7 Innovative Ways to Determine the Lumped-Parameters of Transmission Lines with Sag Using the M-Model

In the preceding section, the lumped-parameters $\{Z_L, Z_R, Y_C, Y_S\}$ are supposed to be known. The main question that might be raised here is: *How to calculate these parameters?* The procedure given here is based on the leveled-spans scenario. However, it is also applicable to the inclined-spans scenario.

6.7.1 Calculating Z_V

This part has been partially covered in (6.118)-(6.120). First of all, it has to be noted that the per-unit-length (Δx) is regularly taken in many references as 1 km or 1 mile. Here, we are dealing with the "**tower-to-tower**" span and not the "per-unit-length". Thus, for not messing up, the span is denoted by $ł_0$ and the total cable length is denoted by $ł_s$. The ith tower-to-tower cable is shown in Figure 6.35.

By comparing Figure 6.35 with Figure 6.33 and Figure 6.34, the total length of a

transmission line without sag can be calculated as follow:

$$\mathfrak{L}_0 = \mathfrak{l}_0 \, (W - 1) \tag{6.129}$$

With the leveled-spans scenario (i.e., uniform sags):

$$\mathfrak{L}_\mathfrak{s} = \mathfrak{l}_\mathfrak{s} \, (W - 1) \tag{6.130}$$

With the inclined-spans scenario (i.e., non-uniform sags)[26]:

$$\mathfrak{L}_\mathfrak{s} = \sum_{i=1}^{W-1} \mathfrak{l}_{\mathfrak{s}_i} \tag{6.131}$$

where $\mathfrak{l}_{\mathfrak{s}_i}$ is the total length of the ith sector; or in other words, the length of the cable installed between the ith and $(i+1)$th towers.

For the leveled-spans scenario, all the $(W-1)$ sectors have the same impedance:

$$z = z_0 \left(\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0} \right) \tag{6.132}$$

where z_0 is the sector impedance before sag, and the ratio $\mathfrak{l}_\mathfrak{s} \, / \, \mathfrak{l}_0$ can be further explained as follows:

$$\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0} = \frac{\mathfrak{L}_\mathfrak{s}}{\mathfrak{L}_0} \tag{6.133}$$

By combining the above equations, the sector impedance can be calculated for the leveled-spans scenario as follows:

$$z = z_0 \left(\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0} \right) = z_0 \left(\frac{\mathfrak{L}_\mathfrak{s}}{\mathfrak{L}_0} \right) = 2 z_V \tag{6.134}$$

where z_V is a half sector impedance.

Therefore, the total transmission line impedance during sag can be calculated as follows:

$$Z = z \, (W - 1) = z_0 \left(\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0} \right) (W - 1) = z_0 \left(\frac{\mathfrak{L}_\mathfrak{s}}{\mathfrak{L}_0} \right) (W - 1) = Z_0 \left(\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0} \right) = Z_0 \left(\frac{\mathfrak{L}_\mathfrak{s}}{\mathfrak{L}_0} \right) \tag{6.135}$$

Finally, the two series impedances Z_V can be calculated as follows:

$$Z_V = \frac{Z}{2} = \frac{z}{2} \, (W - 1) = z_0 \left(\frac{\mathfrak{l}_\mathfrak{s}}{2\mathfrak{l}_0} \right) (W - 1) = z_0 \left(\frac{\mathfrak{L}_\mathfrak{s}}{2\mathfrak{L}_0} \right) (W - 1) \tag{6.136}$$

[26] This equation is given just to show the differences between the leveled- and inclined-spans. Only the first one will be covered here.

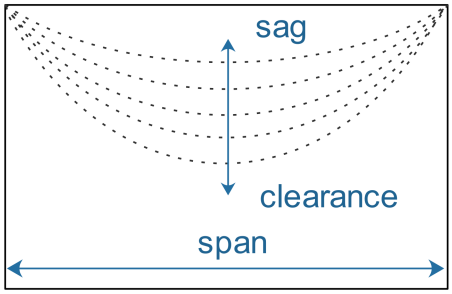


Figure 6.36: Illustrated Variation in Sag and Clearance (Leveled-Spans)

6.7.2 Calculating $\mathbf{Y_C}$

As said before, the two-end shunt admittances Y_S are not changed, and they are equal to the initial value $Y_0/3$. Thus, the **slack admittance** located at the center can be calculated as follows:

$$Y_C = Y - 2Y_S = Y - \frac{2}{3}Y_0 \tag{6.137}$$

where Y and Y_0 are respectively the total line admittance before and during sag.

The total initial admittance Y_0 is known, but *how to find* Y_S*?* For the series impedance, the ratio between z_0 and z can be calculated by dividing the **actual cable length** $ł_s$ by the **ideal cable length** or span $ł_0$. For the shunt capacitance, we are dealing with a varying area below the cable where the sag increases as the **clearance** decreases and vice versa. This phenomenon is illustrated in Figure 6.36.

This actual sector of the leveled-spans scenario can be simplified as an M-circuit as shown in Figure 6.37. It is clear that the difference in the shunt admittance can be compensated by the slack admittance Y_C. Thus, to calculate the actual shunt admittance, two innovative approaches are offered here.

6.7.2.1 Approach No.1

In this approach, the difference in shunt admittance can be accounted for by thinking of it as an **equal-area criterion problem**. For not messing up with the equal-area criterion used in the stability subject, we call it the "**line wings problem** (**LWP**)". The goal here is to find the value of y_2[27]. The parabolic curve can be explained by different approaches [156, 161, 170]. The most popular one is called the **Catenary's**

[27] y_2 is a point on the y-axis of Figure 6.37, not a per-unit-length or tower-to-tower admittance.

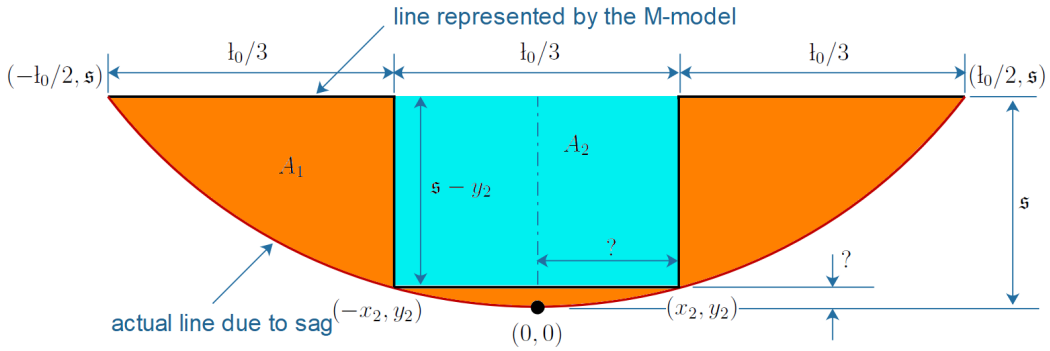


Figure 6.37: Simplifying Actual Sectors to Act as M-Circuits (Leveled-Spans)

shape parabolic equation, which can be mathematically expressed as follows [156]:

$$y(x) = \mathfrak{e}\left[\cosh\left(\frac{x}{\mathfrak{e}}\right) - 1\right] \tag{6.138}$$

where $\mathfrak{e}$ is the clearance, which is shown in Figure 6.35. The independent variable x is any point located between the sag point and the right or left tower. The sag point is considered as the origin point $(0, 0)$, and thus $\mathfrak{s}$ can be calculated as follows:

$$\mathfrak{s} = y\left(\mathfrak{l}_0/2\right) = \mathfrak{e}\left[\cosh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) - 1\right] \tag{6.139}$$

By referring to Figure 6.35, the total area below an **ideal sector** is:

$$A_{\text{total}_0} = \mathcal{E}\mathfrak{l}_0 \tag{6.140}$$

With the presence of sag, a part of A_{total_0} will be eliminated. Because the sag considered here is uniform, so the eliminated area can be calculated as follows:

$$A_{\text{dead}} = \mathfrak{s}\mathfrak{l}_0 - 2\int_0^{\mathfrak{l}_0/2} y(x)\, dx = \mathfrak{s}\mathfrak{l}_0 - 2\mathfrak{e}\int_0^{\mathfrak{l}_0/2}\left[\cosh\left(\frac{x}{\mathfrak{e}}\right) - 1\right]\, dx \tag{6.141}$$

where, from Figure 6.37, y_2 can be determined from (6.138) as follows:

$$y_2 = y\left(x_2\right) = \mathfrak{e}\left[\cosh\left(\frac{x_2}{\mathfrak{e}}\right) - 1\right] \tag{6.142}$$

The goal of Figure 6.37 is to force A_2 to equal A_{dead} so the M-circuit can be built. The area of the blue rectangle equals:

$$A_2 = 2x_2(\mathfrak{s} - y_2) \tag{6.143}$$

From Figure 6.37, it is obvious that A_2 is within the eliminated area A_{dead}. Thus, $A_2 < A_{\text{dead}}$. To ensure $A_2 = A_{\text{dead}}$, three possible methods can be applied, which are graphically described in Figure 6.38. The first one is achieved by fixing the width and increasing the height of the rectangle, and the second one is by increasing the width and fixing the height. The third method is by increasing both the width and height. The last two methods violate the assumption that $Y_S = Y_0/3$, because $2x_2 > \text{ł}_0/3$ and thus $\text{ł}_0/2 - x_2 < \text{ł}_0/3$. Therefore, when the second or third method is used, then Y_S should be adjusted too. Instead, the first method can be used to ensure $Y_S = Y_0/3$. Thus, from Figure 6.38a, (6.143) should be modified to:

$$A_2 = 2x_2(\mathfrak{s} + y_2) = \frac{\text{ł}_0}{3}(\mathfrak{s} + y_2) \tag{6.144}$$

and y_2 can be calculated by equating A_2 and A_{dead} as follows:

$$y_2 = \frac{3}{\text{ł}_0}\left\{\mathfrak{s}\text{ł}_0 - 2\mathfrak{e}\int_0^{\text{ł}_0/2}\left[\cosh\left(\frac{x}{\mathfrak{e}}\right) - 1\right]\, dx\right\} - \mathfrak{s} \tag{6.145}$$

Based on our observations, it has been found that $y_2 \approx \mathfrak{s}$ for many random values of ł_0 and $\mathfrak{e}$.

After finding the height y_2, the slack admittance Y_C can be determined from the equations of the transmission line parameters. For example, by referring to (6.47)-(6.56) and their corresponding references, the shunt capacitance can be calculated for the new height y_2 (instead of $\mathcal{E}$) and the new length $\text{ł}_0/3$ (instead of ł_0).

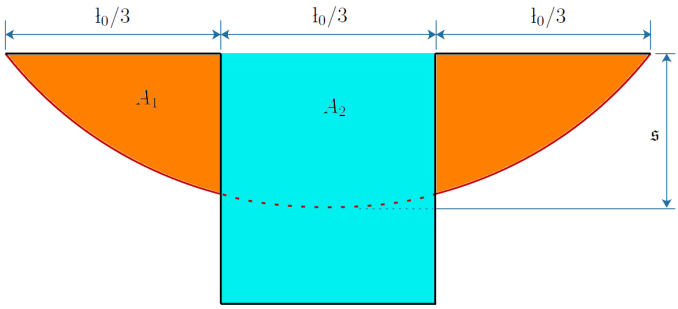
6.7.2.2 Approach No.2

This method is much simpler than the preceding one. It can estimate the shunt admittance directly without referring to any equation of (6.47)-(6.56). The procedure is similar to that used with the series impedance, but by using the ratio between A_{total_0} and $A_{\text{total}_\mathfrak{s}}$ instead of the ratio between $\text{ł}_\mathfrak{s}$ and ł_0. The term $A_{\text{total}_\mathfrak{s}}$ can be calculated as follows:

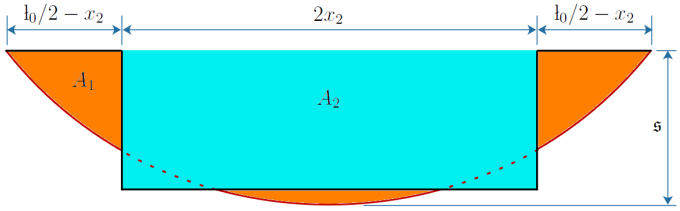
$$A_{\text{total}_\mathfrak{s}} = A_{\text{total}_0} - A_{\text{dead}} \tag{6.146}$$

Figure 6.39 describes the whole process. The effective area is $A_{\text{total}_\mathfrak{s}}$, and the eliminated area is A_{dead}. With ideal transmission lines, there is no sag[28], so the

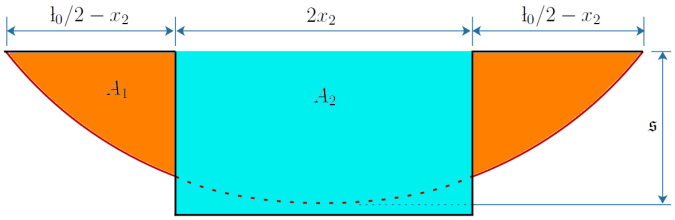
[28]Both approaches can also work with real transmission lines by just replacing the span ł_0 with the lowest actual cable length $\text{ł}_{\mathfrak{s}_0}$; i.e., when the cable has the lowest possible sag.



(a) Method No.1



(b) Method No.2



(c) Method No.3

Figure 6.38: Possible Ways to Compensate for the Variation in the Shunt Admittance

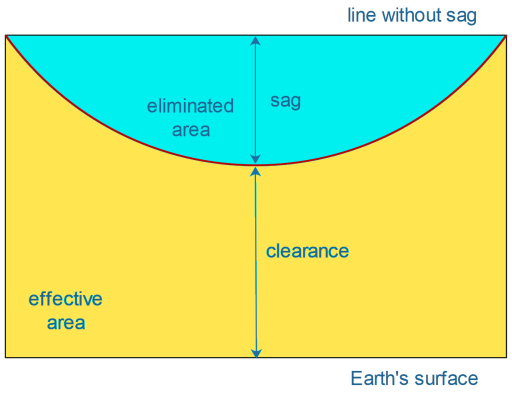


Figure 6.39: Effective Area of Overhead Cables with Sag

eliminated area equals zero and thus the effective area becomes A_{total_0}. Thus, the total shunt admittance of real transmission lines can be calculated as follows:

$$Y = Y_0 \left(\frac{A_{\text{total}_0}}{A_{\text{total}_s}} \right) \tag{6.147}$$

Once Y is determined, the slack admittance Y_C can be calculated from (6.137) where $Y_S = Y_0/3$.

6.8 Estimating the Parameters of Sag Transmission Lines

In Section 6.4, the distributed parameters $\{R, X_L, X_C, G\}$ have been modeled with considering the effects of conductor temperature $\breve{T}$ and system frequency ω, which were extensively discussed in Section 6.3. In Section 6.6, the inclined- and leveled-spans sagging phenomena have been realized by using a modified version of the M-model. Some innovative methods to approximate the series impedances and the shunt admittances of the M-circuit have been discussed in Section 6.7. However, except the resistance temperature coefficient α, the other temperature coefficients $\{\beta, \kappa, \lambda\}$ presented in Section 6.3 still are unknown. Even in the literature, there are just a few studies focus on these coefficients by assuming that they are known and available [12, 82, 219]. We have tried our best to find a reference that gives some ways to estimate these coefficients, but - unfortunately - we failed. This motivates us to derive our own estimations towards these coefficients, which are the core of this section. They can be used as bases to develop more advanced approximators.

Before starting, it has to be remembered that the per-unit-length Δx is replaced with the span[29] $ł_0$. Also, the symbol x is used to describe the points within $ł_0$. The other thing is that the following estimations are derived solely from the effect of the conductor temperature $\breve{T}$ because it is the most significant variable that represents many weather and system variables. Although the leveled-spans scenario is discussed here, the inclined-spans scenario can also be solved by applying the same steps with some essential corrections.

[29] We also call it the ideal cable length and the tower-to-tower length.

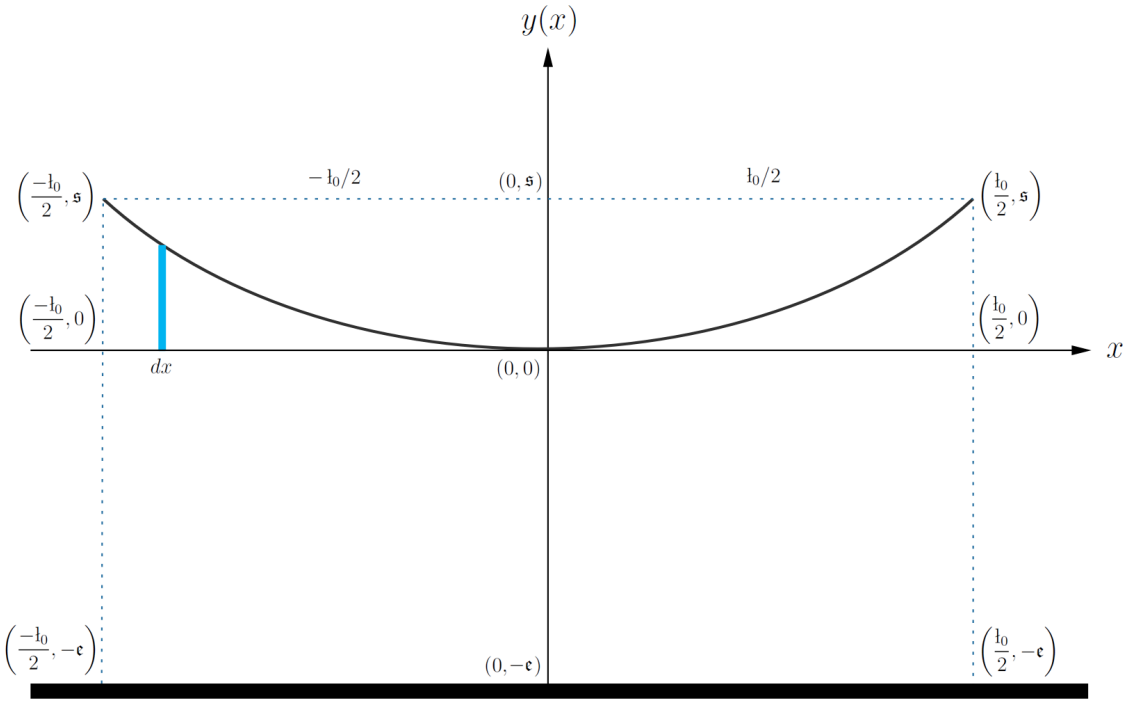


Figure 6.40: Mathematical Model of the Leveled-Span Sector

6.8.1 Estimating the Cable Length

By referring to Figure 6.35, the parabolic curve can be mathematically explained by the Catenary's shape parabolic equation and many other equations. This mathematical model is shown in Figure 6.40 for any function $y(x)$. In this section, the Catenary's equation given in (6.138) is used to estimate the height of the sagged cable at any point lies between the left and right towers.

Based on (6.138), $\mathfrak{l}_\mathfrak{s}$ can be determined by finding the total length of $y(x)$ using the "arc length" technique as follows:

$$\begin{aligned}
\mathfrak{l}_\mathfrak{s} &= 2\int_0^{\mathfrak{l}_0/2} \sqrt{1+\left[y'(x)\right]^2}\, dx = 2\int_0^{\mathfrak{l}_0/2} \sqrt{1+\left\{\frac{d}{dx}\left[\mathfrak{e}\cosh\left(\frac{x}{\mathfrak{e}}\right)-\mathfrak{e}\right]\right\}^2}\, dx \\
&= 2\int_0^{\mathfrak{l}_0/2} \sqrt{1+\sinh^2\left(\frac{x}{\mathfrak{e}}\right)}\, dx = 2\int_0^{\mathfrak{l}_0/2} \sqrt{\cosh^2\left(\frac{x}{\mathfrak{e}}\right)}\, dx = 2\int_0^{\mathfrak{l}_0/2} \cosh\left(\frac{x}{\mathfrak{e}}\right)\, dx \\
&= 2\left[\mathfrak{e}\sinh\left(\frac{x}{\mathfrak{e}}\right)\right]_0^{\mathfrak{l}_0/2} = 2\mathfrak{e}\left[\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) - \cancel{\sinh(0)}\right] = 2\mathfrak{e}\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right)
\end{aligned} \tag{6.148}$$

where the span $\mathfrak{l}_0$ is constant and known, and the clearance $\mathfrak{e}$ depends on the sag $\mathfrak{s}$

by the following relationship:

$$\mathfrak{e} = \mathcal{E} - \mathfrak{s} \tag{6.149}$$

where $\mathcal{E}$ is constant and known.

This means that the cable length $\mathfrak{l}_\mathfrak{s}$ can be expressed as a function of $\mathfrak{s}$:

$$\therefore \quad \boxed{\mathfrak{l}_\mathfrak{s} = 2\left(\mathcal{E} - \mathfrak{s}\right)\sinh\left[\frac{\mathfrak{l}_0}{2\left(\mathcal{E} - \mathfrak{s}\right)}\right]} \tag{6.150}$$

6.8.2 Estimating the Sag Point

Practically, the sag point $\mathfrak{s}$ can be precisely calculated via some instrument devices. But, *what if the cables are not equipped with such devices?* The following equations show how to approximate $\mathfrak{s}$ based on the conductor temperature $\breve{T}$.

First, assume that the shunt distributed parameters are very small and can be neglected. Thus, the short-length transmission line model can be applied. By referring to (6.19) and (6.27), $R(\breve{T})$ can be expressed as follows [12, 13]:

$$R(\breve{T}) = R_0\left[1 + \alpha\left(\breve{T} - \breve{T}_0\right)\right] \tag{6.151}$$

By using the ratio between the sag and no-sag conditions given in (6.135) on the real part of Z, $R(\breve{T})$ can also be expressed as follows:

$$R(\breve{T}) \approx R_0\left(\frac{\mathfrak{l}_\mathfrak{s}}{\mathfrak{l}_0}\right) \tag{6.152}$$

Equating (6.151) with (6.152) gives:

$$\mathfrak{l}_\mathfrak{s} \approx \mathfrak{l}_0\left[1 + \alpha\left(\breve{T} - \breve{T}_0\right)\right] \tag{6.153}$$

where α is known and can be easily found for each specific material.

Equating (6.150) with (6.153) gives:

$$2\left(\mathcal{E} - \mathfrak{s}\right)\sinh\left[\frac{\mathfrak{l}_0}{2\left(\mathcal{E} - \mathfrak{s}\right)}\right] \approx \mathfrak{l}_0\left[1 + \alpha\left(\breve{T} - \breve{T}_0\right)\right] \tag{6.154}$$

which can be solved numerically to find $\mathfrak{s}$.

The other straightforward way is to use the following formula given in p.16 of [170]:

$$\mathfrak{l}_\mathfrak{s} \approx \mathfrak{l}_0 + \frac{8\mathfrak{s}^2}{3\mathfrak{l}_0} \tag{6.155}$$

Substituting (6.155) in (6.153) for $\text{ł}_\text{ś}$ yields:

$$\begin{aligned}
\text{ł}_0 + \frac{8\text{ś}^2}{3\text{ł}_0} &\approx \text{ł}_0\left[1+\alpha\left(\breve{T}-\breve{T}_0\right)\right] \\
\cancel{3\text{ł}_0^2} + 8\text{ś}^2 &\approx 3\text{ł}_0^2\left[\cancel{1}+\alpha\left(\breve{T}-\breve{T}_0\right)\right] = \cancel{3\text{ł}_0^2} + 3\alpha\text{ł}_0^2\left(\breve{T}-\breve{T}_0\right) \\
8\text{ś}^2 &\approx 3\alpha\text{ł}_0^2\left(\breve{T}-\breve{T}_0\right)
\end{aligned}$$

$$\therefore \boxed{\text{ś} \approx \frac{\text{ł}_0}{4}\sqrt{6\alpha\left(\breve{T}-\breve{T}_0\right)}} \tag{6.156}$$

Because many factors might affect the actual value of ś, so the error associated with (6.156) could be reduced by taking the average of ś at different values of $\breve{T}$.

6.8.3 Estimating the Inductance Temperature Coefficient

By referring to Section 6.3, the following general equation can be derived for X_L:

$$X_L(\breve{T}) = X_{L_0}\left[1+\beta\left(\breve{T}-\breve{T}_0\right)\right] \tag{6.157}$$

where β is unknown and there is a lack of information about it [12, 13, 82, 219].

By using the same ratio given in (6.152), $X_L(\breve{T})$ can be approximated as follows:

$$X_L(\breve{T}) \approx X_{L_0}\left(\frac{\text{ł}_\text{ś}}{\text{ł}_0}\right) \tag{6.158}$$

Equating (6.157) with (6.158) gives:

$$\frac{\text{ł}_\text{ś}}{\text{ł}_0} \approx 1+\beta\left(\breve{T}-\breve{T}_0\right) \tag{6.159}$$

Thus, solving the last equation for β gives:

$$\therefore \boxed{\beta \approx \frac{\text{ł}_\text{ś}\,/\,\text{ł}_0 - 1}{\breve{T}-\breve{T}_0}} \tag{6.160}$$

6.8.4 Estimating the Capacitance Temperature Coefficient

Let's recall (6.46):

$$X_C(\breve{T}) = \frac{1}{2\pi f C_0\left[1+\kappa\left(\breve{T}-\breve{T}_0\right)\right]} \tag{6.46}$$

Again, there is a lack of information about κ [12,13,82,219]. For the shunt parameters, the approach given in (6.147) can be used to approximate $X_C(\breve{T})$ as follows:

$$X_C(\breve{T}) \approx \left(\frac{1}{2\pi f C_0}\right)\left(\frac{A_{\text{total}_0}}{A_{\text{total}_s}}\right) \tag{6.161}$$

where A_{total_0} and A_{total_s} can be calculated by (6.140) and (6.146), respectively.

By referring to Figure 6.35 and Figure 6.40, the simpler way to calculate A_{total_s} is as follows:

$$\begin{aligned} A_{\text{total}_s} &= \mathfrak{e}\mathfrak{l}_0 + 2\int_0^{\mathfrak{l}_0/2} y(x)\,dx = \mathfrak{e}\mathfrak{l}_0 + 2\mathfrak{e}\int_0^{\mathfrak{l}_0/2}\left[\cosh\left(\frac{x}{\mathfrak{e}}\right) - 1\right]\,dx \\ &= \cancel{\mathfrak{e}\mathfrak{l}_0} + 2\mathfrak{e}\int_0^{\mathfrak{l}_0/2}\cosh\left(\frac{x}{\mathfrak{e}}\right)\,dx - \cancel{2\mathfrak{e}\int_0^{\mathfrak{l}_0/2} dx} = 2\mathfrak{e}\left[\mathfrak{e}\sinh\left(\frac{x}{\mathfrak{e}}\right)\right]_0^{\mathfrak{l}_0/2} \\ &\therefore \boxed{A_{\text{total}_s} = 2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right)} \end{aligned} \tag{6.162}$$

Substituting (6.140) and (6.162) in (6.161) for A_{total_0} and A_{total_s}, respectively, and then equating (6.161) with (6.46) yields:

$$\begin{aligned} \frac{\mathcal{E}\mathfrak{l}_0}{2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right)} &\approx \frac{1}{1+\kappa\left(\breve{T}-\breve{T}_0\right)} \\ \mathcal{E}\mathfrak{l}_0 + \mathcal{E}\mathfrak{l}_0\kappa\left(\breve{T}-\breve{T}_0\right) &\approx 2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) \\ \therefore \boxed{\kappa \approx \frac{2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) - \mathcal{E}\mathfrak{l}_0}{\mathcal{E}\mathfrak{l}_0\left(\breve{T}-\breve{T}_0\right)}} \end{aligned} \tag{6.163}$$

6.8.5 Estimating the Conductance Temperature Coefficient

Recalling (6.61):

$$G(\breve{T}) = \frac{1}{\tilde{R}_0\left[1+\lambda\left(\breve{T}-\breve{T}_0\right)\right]} \tag{6.61}$$

By applying the same ratio used in (6.161):

$$G(\breve{T}) \approx \left(\frac{1}{\tilde{R}_0}\right)\left(\frac{A_{\text{total}_0}}{A_{\text{total}_s}}\right) \tag{6.164}$$

Equating (6.61) with (6.164) gives:

$$\begin{aligned} \frac{\mathcal{E}\mathfrak{l}_0}{2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right)} &\approx \frac{1}{1+\lambda\left(\breve{T}-\breve{T}_0\right)} \\ \mathcal{E}\mathfrak{l}_0 + \mathcal{E}\mathfrak{l}_0\lambda\left(\breve{T}-\breve{T}_0\right) &\approx 2\mathfrak{e}^2\sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) \end{aligned}$$

$$\therefore \quad \boxed{\lambda \approx \frac{2\mathfrak{e}^2 \sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) - \mathcal{E}\mathfrak{l}_0}{\mathcal{E}\mathfrak{l}_0\left(\breve{T} - \breve{T}_0\right)}} \tag{6.165}$$

Because G is very small and sensitive to many operating and weather conditions, so the approximation given in (6.165) could be further enhanced by multiplying it by a weight ξ_1 and the result is added to an intercept ξ_0 as follows:

$$\therefore \quad \boxed{\lambda \approx \xi_0 + \xi_1 \left[\frac{2\mathfrak{e}^2 \sinh\left(\frac{\mathfrak{l}_0}{2\mathfrak{e}}\right) - \mathcal{E}\mathfrak{l}_0}{\mathcal{E}\mathfrak{l}_0\left(\breve{T} - \breve{T}_0\right)}\right]} \tag{6.166}$$

where the coefficients ξ_0 and ξ_1 are respectively the intercept and slope of a curve fitted by simple linear regression analysis.

The observations required to conduct that analysis can be generated by measuring $\mathfrak{e}$, or $\mathfrak{s}$ if (6.149) is used, at different $\breve{T}$. The response can be the actual readings measured at the sending-end and receiving-end terminals.

6.8.6 Estimating All the Parameters Without Knowing the Conductor Temperature

From (6.153), (6.156), (6.160), (6.163), (6.165) and (6.166), it is clear that all the estimations are based on the conductor temperature $\breve{T}$. It is assumed to be known and given. In reality, measuring $\breve{T}$ is not a usual practice. Instead, that variable can be estimated by involving the heat transfer topic where the network and metrology variables and the cable design, status, and age all are required. The cable information is already known. The metrology variables (ambient temperature, atmospheric pressure, humidity, wind speed, wind direction, etc) can be easily obtained from many weather stations and satellites. The operating variables (system frequency, voltage, current, power, etc) can be extracted from the two fundamental measurements received from the corresponding CT and PT.

6.8.7 Estimating All the Parameters Without Knowing the Conductor Temperature or Involving the Heat Transfer Topic

Similar to (6.27), the thermal expansion topic tells us that when a material is exposed to a temperature change ($\Delta\breve{T} = \breve{T} - \breve{T}_0$) it expands or contracts according to [102]:

$$\Delta\mathfrak{L} = \eta\mathfrak{L}_0\Delta\breve{T} \tag{6.167}$$

where $\Delta\mathfrak{L}$ = the change in length due to $\Delta\breve{T}$

$\mathfrak{L}_0$ = the initial length measured at $\breve{T}_0$

η = the thermal expansion coefficient

This equation can be used as a basis to estimate $\mathfrak{l}_\mathfrak{s}$, $\mathfrak{s}$, β, $\mathfrak{e}$, $A_{\text{total}_\mathfrak{s}}$, κ and λ. First, let's start estimating the cable length during sag. The cable of each sector is hanged between two fixed towers. Despite the effects of snow, ice, sandstorm, wind speed and wind direction, the sag point $\mathfrak{s}$ is significantly affected by the span $\mathfrak{l}_0$ and the physical properties of the cable. Such properties are the cable weight, material types and design (AAC, AAAC, ACAR, and ACSR), elasticity, status, and age. Thus, even with fixing $\mathfrak{l}_0$ and $\breve{T}$ and neglecting the other weather effects, each cable has its own sag profile. Based on (6.138) and Figure 6.36, the expansion in the cable length $\mathfrak{l}_\mathfrak{s}$ is nonlinear. This means that the higher-order terms of (6.167) should be included to explain the nonlinearity[30]. Thus, by expanding (6.167) and substituting (6.133):

$$\mathfrak{l}_\mathfrak{s} = \mathfrak{l}_0\left(1 + \eta_1\Delta\breve{T} + \eta_2\Delta\breve{T}^2 + \cdots + \eta_k\Delta\breve{T}^k\right) \tag{6.168}$$

The dissipated heat is governed by:

$$P = 3\left|I\right|^2 R \tag{6.169}$$

where the resistance R increases as I increases due to the self-heating phenomenon.

This means that there is a correlation between the active power P flowing in the cable and the conductor temperature $\breve{T}$. Thus, (6.168) could be re-expressed with other coefficients as follows:

$$\begin{aligned}\mathfrak{l}_\mathfrak{s} &\approx \mathfrak{l}_0\left(1 + \varsigma_1\Delta P + \varsigma_2\Delta P^2 + \cdots + \varsigma_\mathfrak{p}\Delta P^\mathfrak{p}\right)\\ &\approx \mathfrak{l}_0\left(1 + \varsigma_1 P_L + \varsigma_2 P_L^2 + \cdots + \varsigma_\mathfrak{p} P_L^\mathfrak{p}\right)\end{aligned} \tag{6.170}$$

[30] Exactly, similar to (6.26).

Based on (6.105), the active power loss P_L across the entire line can be calculated as follows:

$$P_L = |P_S| - |P_R| \tag{6.171}$$

where P_S and P_R are respectively the sending-end and receiving-end active power, which can be calculated from the solution of the two-port network of any transmission line model presented before. The absolute notation is used because the sign of each power flow depends on its direction.

Because $\breve{T}$ indirectly affects X_L and X_C, as respectively seen in (6.157) and (6.46), so $\mathfrak{l}_{\mathfrak{s}}$ can also be estimated by Q_L as follows:

$$\mathfrak{l}_{\mathfrak{s}} \approx \mathfrak{l}_0 \left(1 + \zeta_1 Q_L + \zeta_2 Q_L^2 + \cdots + \zeta_{\mathfrak{q}} Q_L^{\mathfrak{q}}\right) \tag{6.172}$$

Similar to (6.171), by referring to (6.106), the reactive power loss Q_L across the entire line can be calculated as follows:

$$Q_L = |Q_S| - |Q_R| \tag{6.173}$$

where Q_S and Q_R are respectively the sending-end and receiving-end reactive power.

The error associated with (6.170) and (6.172) could be reduced by taking the average as follows:

$$\frac{\Delta\mathfrak{l}}{\mathfrak{l}_0} \approx \frac{\varsigma_1 P_L + \zeta_1 Q_L + \varsigma_2 P_L^2 + \zeta_2 Q_L^2 + \cdots + \varsigma_{\mathfrak{p}} P_L^{\mathfrak{p}} + \zeta_{\mathfrak{q}} Q_L^{\mathfrak{q}}}{2} \tag{6.174}$$

where $\Delta\mathfrak{l} = \mathfrak{l}_{\mathfrak{s}} - \mathfrak{l}_0$.

By distributing 1/2 for each term, (6.174) can be re-expressed as follows:

$$\boxed{\mathfrak{l}_{\mathfrak{s}} = \mathfrak{l}_0 \left(1 + \tilde{\varsigma}_1 P_L + \tilde{\zeta}_1 Q_L + \tilde{\varsigma}_2 P_L^2 + \tilde{\zeta}_2 Q_L^2 + \cdots + \tilde{\varsigma}_{\tilde{\mathfrak{p}}} P_L^{\tilde{\mathfrak{p}}} + \tilde{\zeta}_{\tilde{\mathfrak{q}}} Q_L^{\tilde{\mathfrak{q}}}\right)} \tag{6.175}$$

where $\tilde{\varsigma}_i = \varsigma_i \,/\, 2$ and $\tilde{\zeta}_j = \zeta_j \,/\, 2$.

Equating (6.168) with (6.175) gives:

$$\eta_1 \Delta\breve{T} + \cdots + \eta_k \Delta\breve{T}^k \approx \tilde{\varsigma}_1 P_L + \tilde{\zeta}_1 Q_L + \cdots + \tilde{\varsigma}_{\tilde{\mathfrak{p}}} P_L^{\tilde{\mathfrak{p}}} + \tilde{\zeta}_{\tilde{\mathfrak{q}}} Q_L^{\tilde{\mathfrak{q}}} \tag{6.176}$$

Thus, the change in temperature $\Delta\breve{T}$ can be estimated by any order as follows:

$$\begin{aligned} a\eta\Delta\breve{T} &\approx \check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L \\ &\approx \check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L + \check{\varsigma}_2 P_L^2 + \check{\zeta}_2 Q_L^2 \\ &\approx \check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L + \check{\varsigma}_2 P_L^2 + \check{\zeta}_2 Q_L^2 + \cdots + \check{\varsigma}_{\check{\mathfrak{p}}} P_L^{\check{\mathfrak{p}}} + \check{\zeta}_{\check{\mathfrak{q}}} Q_L^{\check{\mathfrak{q}}} \end{aligned} \tag{6.177}$$

where a is an external constant.

Based on this, by taking $a = \sqrt{6\alpha}/4$ for (6.156), the temperature-free estimation of $\mathfrak{s}$ is:

$$\therefore \quad \boxed{\mathfrak{s} \approx \mathfrak{h}_0 \sqrt{\check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L + \check{\varsigma}_2 P_L^2 + \check{\zeta}_2 Q_L^2 + \cdots + \check{\varsigma}_{\check{\mathfrak{p}}} P_L^{\check{\mathfrak{p}}} + \check{\zeta}_{\check{\mathfrak{q}}} Q_L^{\check{\mathfrak{q}}}}} \tag{6.178}$$

To estimate the inductance temperature coefficient β, (6.160) is re-expressed as follows:

$$\beta \approx \frac{\mathfrak{h}_{\mathfrak{s}} \,/\, \mathfrak{h}_0 - 1}{\check{T} - \check{T}_0} \approx \frac{\mathfrak{h}_{\mathfrak{s}} - \mathfrak{h}_0}{\mathfrak{h}_0 \left(\check{T} - \check{T}_0\right)} \approx \frac{\Delta\mathfrak{h} \,/\, \mathfrak{h}_0}{\check{T} - \check{T}_0} \tag{6.179}$$

This is a ratio between (6.175) and (6.177) without the constant terms:

$$\therefore \quad \boxed{\beta \approx \frac{\tilde{\varsigma}_1 P_L + \tilde{\zeta}_1 Q_L + \tilde{\varsigma}_2 P_L^2 + \tilde{\zeta}_2 Q_L^2 + \cdots + \tilde{\varsigma}_{\tilde{\mathfrak{p}}} P_L^{\tilde{\mathfrak{p}}} + \tilde{\zeta}_{\tilde{\mathfrak{q}}} Q_L^{\tilde{\mathfrak{q}}}}{\check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L + \check{\varsigma}_2 P_L^2 + \check{\zeta}_2 Q_L^2 + \cdots + \check{\varsigma}_{\check{\mathfrak{p}}} P_L^{\check{\mathfrak{p}}} + \check{\zeta}_{\check{\mathfrak{q}}} Q_L^{\check{\mathfrak{q}}}}} \tag{6.180}$$

To estimate $A_{\text{total}_{\mathfrak{s}}}$ without knowing the conductor temperature $\check{T}$, the clearance $\mathfrak{e}$ must be re-expressed by substituting (6.178) in (6.149) for $\mathfrak{s}$ as follows:

$$\therefore \quad \boxed{\mathfrak{e} \approx \mathcal{E} - \mathfrak{h}_0 \sqrt{\check{\varsigma}_1 P_L + \check{\zeta}_1 Q_L + \check{\varsigma}_2 P_L^2 + \check{\zeta}_2 Q_L^2 + \cdots + \check{\varsigma}_{\check{\mathfrak{p}}} P_L^{\check{\mathfrak{p}}} + \check{\zeta}_{\check{\mathfrak{q}}} Q_L^{\check{\mathfrak{q}}}}} \tag{6.181}$$

If the sigma-notation is used, then the last equation can be re-expressed as follows:

$$\therefore \quad \boxed{\mathfrak{e} \approx \mathcal{E} - \mathfrak{h}_0 \sqrt{\sum_{i=1}^{\check{\mathfrak{p}}} \left(\check{\varsigma}_i P_L^i\right) + \sum_{j}^{\check{\mathfrak{q}}} \left(\check{\zeta}_j Q_L^j\right)}} \tag{6.182}$$

Substituting (6.181) in (6.162) for $\mathfrak{e}$ yields:

$$\boxed{A_{\text{total}_{\mathfrak{s}}} = 2 \left[\mathcal{E} - \mathfrak{h}_0 \sqrt{\sum_{i=1}^{\check{\mathfrak{p}}} \left(\check{\varsigma}_i P_L^i\right) + \sum_{j}^{\check{\mathfrak{q}}} \left(\check{\zeta}_j Q_L^j\right)}\right]^2 \times \sinh \left[\frac{\mathfrak{h}_0 \,/\, 2}{\mathcal{E} - \mathfrak{h}_0 \sqrt{\sum_{i=1}^{\check{\mathfrak{p}}} \left(\check{\varsigma}_i P_L^i\right) + \sum_{j}^{\check{\mathfrak{q}}} \left(\check{\zeta}_j Q_L^j\right)}}\right]} \tag{6.183}$$

As can be seen, the effective area depicted in Figure 6.39 can be estimated from active and reactive power losses, which can be easily measured from both end CTs and PTs[31].

[31] Modern remotely-accessed numerical relays and phasor measurements units (PMUs) could also be utilized to measure online values of P_L and Q_L.

Recalling (6.163):

$$\kappa \approx \frac{2\mathfrak{e}^2 \sinh\left(\frac{\mathfrak{h}_0}{2\mathfrak{e}}\right) - \mathcal{E}\mathfrak{h}_0}{\mathcal{E}\mathfrak{h}_0\left(\breve{T} - \breve{T}_0\right)} \approx \frac{\frac{2}{\mathcal{E}\mathfrak{h}_0}\mathfrak{e}^2 \sinh\left(\frac{\mathfrak{h}_0}{2\mathfrak{e}}\right) - 1}{\breve{T} - \breve{T}_0} \tag{6.184}$$

Thus, by replacing $\mathfrak{e}$ with (6.182), or $2\mathfrak{e}^2 \sinh\left(\frac{\mathfrak{h}_0}{2\mathfrak{e}}\right)$ with (6.183), and then replacing $\breve{T} - \breve{T}_0$ with (6.177) after taking $a\eta = 1$, the preceding κ-estimator[32] can be re-expressed by the following temperature-free formula:

$$\boxed{\begin{aligned}\kappa \approx & \frac{\frac{2}{\mathcal{E}\mathfrak{h}_0}\left[\mathcal{E} - \mathfrak{h}_0\sqrt{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)}\right]^2}{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)} \\ & \times \sinh\left[\frac{\mathfrak{h}_0 \,/\, 2}{\mathcal{E} - \mathfrak{h}_0\sqrt{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)}}\right] \\ & - \left[\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)\right]^{-1}\end{aligned}} \tag{6.185}$$

Similar thing for the conductance temperature coefficient λ. If (6.166) is used, then λ can be estimated by using the following temperature-free formula:

$$\boxed{\begin{aligned}\lambda \approx \xi_0 + \xi_1 \Bigg\{ & \frac{\frac{2}{\mathcal{E}\mathfrak{h}_0}\left[\mathcal{E} - \mathfrak{h}_0\sqrt{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)}\right]^2}{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)} \\ & \times \sinh\left[\frac{\mathfrak{h}_0 \,/\, 2}{\mathcal{E} - \mathfrak{h}_0\sqrt{\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)}}\right] \\ & - \left[\sum_{i=1}^{\breve{\mathfrak{p}}}\left(\breve{\zeta}_i P_L^i\right) + \sum_{j}^{\breve{\mathfrak{q}}}\left(\breve{\zeta}_j Q_L^j\right)\right]^{-1} \Bigg\}\end{aligned}} \tag{6.186}$$

As can be clearly seen from (6.175), (6.178), (6.180)-(6.183), (6.185) and (6.186), all the sag line parameters could be estimated without using any instrument devices,

[32] i.e., the capacitance temperature coefficient.

knowing the temperature of conductors, or applying complicated heat transfer models. All we need is to regress these parameters with active and reactive power losses. These P_L and Q_L should be measured at different operating conditions and times[33].

Again, these formulas could be considered as a basis or foundation for future studies on the same topic. For example, in Chapter 10, we will present some novel ML computing systems that could be used here to generate more advanced approximated functions. They could even be highly complicated nonlinear equations with the preceding independent variables $\{\breve{T}, P_L, Q_L, \cdots\}$.

[33]In Chapter 5, it has been said that the weather conditions change with time. These changes could happen within minutes, hours, days, weeks, or even months like seasons.

Chapter 7

Realizing Economic Load Dispatch Problems

The economic load dispatch (ELD) package in energy management systems (EMS) is very important for any electric power system. Ensuring optimal operation can save millions of dollars. Based on this, many techniques have been presented in the literature and promoted as the best ELD solvers with some numerical results posted as evidence of their superiority. Chapter 3 covers the classical formulation of ELD problems with some numerical experiments using one of our most powerful optimization algorithms. However, still, there is a doubt about the ELD mathematical optimization model itself. For example, the dynamic changes of the system and surrounding weather conditions are not accounted for, and thus the actual active and reactive power losses are shifted away from their nominal or steady-state values. Also, all the known ELD solvers are built based on an assumption that each generating unit is connected to one specific busbar, while in reality multiple units could be connected together to one **common busbar** as can be seen in most thermal power stations. Furthermore, some thermal units are operated by a mixture of different fuels, which means that the present multi-fuel ELD models are not valid anymore. Add to that, there are many other abandoned and hidden phenomena that need to be either modeled as an extension to the current objective function(s) or as additional design constraints. Such phenomena are the efficiency degrading of power equipment, the fuel and power losses in power stations, the impacts of delayed maintenance, the correct fuel-cost of spinning reserve units, the dependency of **steam turbines** (**STs**) on **gas turbines** (**GTs**) when they are operated as **combined-cycle power plants** (**CCPPs**), etc. Therefore, the whole ELD model needs to be revisited and modified with some essential corrections and realistic constraints, which is the main goal of this chapter. If this non-ordinary realistic ELD model is correctly coded, then the profitability index could be increased by detecting realistic solutions that satisfy both the optimality and feasibility criteria.

7.1 Economic Load Dispatch Using the TFB Model

The relationship between the TFB-model and ELD problems can be summarized in (3.16), specifically in P_L, because the losses in the network depend on: 1. the system and metrology variables, and 2. the model used to represent lines; as seen before in Chapter 6. This leads us to a very serious fact where (3.16) is an equality constraint, so any significant change in $\breve{T}$ and/or f could result in getting non-feasible solutions. For optimizing ELD problems, it is common to use Kron's loss formula given in (3.18). This formula is a compact version of (6.108) where only generator buses are taken into account to reduce the computation time [123].

To simulate the effects of $\breve{T}$ and f on any ELD problem, the B-coefficients of Kron's loss formula should be updated for any dynamic change of $\breve{T}$ or f, so the corrected values of P_L and Q_L can be attained; as highlighted before in Table 6.11.

Unfortunately, the popular ELD test systems, given in the literature, have insufficient information[1] that can be utilized for other studies. Thus, to see the contribution of employing TFB in solving ELD problems, the fuel-cost functions used in the IEEE 3-unit ELD test system, which is described in [28, 348], are used for the three units of the WSCC 9-bus test system given in [26]. The following search space is used: $P_1 \in [55, 450]$, $P_2 \in [60, 240]$, and $P_3 \in [25, 90]$ - all in MW. The MpBBO-SQP algorithm, presented in Chapters 2 and 3, is selected here to solve this new realistic ELD problem with using the same initialization parameters and a high enough number of generations and population size to get rid of all the sensitivity effects of the algorithm parameters[2], so the global optimum solution can be attained.

As can be clearly seen from Table 7.1, $\breve{T}$ has a significant effect on the network losses, which in turn gives a doubt whether (3.16) is satisfied or not. By comparing the first and third cases, it can be observed that the frequency effect on the ELD solutions is very small. This can also be observed by comparing the second and fourth cases. This is logical, because: 1. $P_L \propto 3|I|^2R$ where the effect of f on the resistance is neglected in this simulation, and 2. the practical range of f is very close to its nominal frequency f_0[3].

[1] Such as per-unit length series impedance and reactance and shunt capacitance and conductance.

[2] Because the main goal of this section is to study the effects of $\breve{T}$ and f.

[3] i.e., based on the standard frequencies, $f \rightarrow f_0 = 50$ or 60 Hz. We take $f_0 = 50$ Hz and $f = 59.85$ Hz, so $\Delta f = 0.15$ Hz.

Table 7.1: $\breve{T}$ and f Effects on the Solution Quality of the 3-Unit ELD Problem

Unit No.	**Case[a] No.1**	**Case No.2**	**Case No.3**	**Case No.4**
P_1 (MW)	154.7331	154.7363	154.7331	154.749
P_2 (MW)	72.9764	73.6899	72.9771	73.6781
P_3 (MW)	90	90	90	90
Total Power (MW)	317.7095	318.4262	317.7102	318.4271
Losses (MW)	2.7095	3.4262	2.7102	3.4271
Min. Cost ($/hr)	3730.8199	3741.5683	3730.8303	3741.6194

[a]These four cases are covered in Chapter 6, which are respectively defined as follows: 1. $\breve{T} = 20°$C and $f = 60$ Hz, 2. $\breve{T} = 80°$C and $f = 60$ Hz, 3. $\breve{T} = 20°$C and $f = 59.85$ Hz, and 4. $\breve{T} = 80°$C and $f = 59.85$ Hz.

7.2 Fuel Cost Modeling for Spinning Reserve Thermal Generating Units

Compared with solar and wind energy, the conventional production of fossil-based turbines (i.e. STs and GTs) is massive, especially if they are operated as CCPPs. For example, the following models are some commercial heavy-duty GTs and STs available in the market [16]:

- **Gas Turbines:**
  - SIEMENS — SGT5-8000H (450 MW)
  - General Electric — 9HA.01 (446 MW)
  - General Electric — 9HA.02 (557 MW)
  - Mitsubishi Hitachi — M701J Series (480-490 MW)
  - Ansaldo Energia — GT36 S5 (500 MW)
- **Steam Turbines:**
  - SIEMENS — SST-6000 (300-1200 MW)
  - SIEMENS — SST-9000 (1000-1900 MW)
  - General Electric — STF-D1050 (1200 MW)
  - General Electric — STF-D850 (1000 MW)
  - Mitsubishi Hitachi — HP-IP-2(3)LP (1200 MW)
  - Ansaldo Energia — RT30 (150-1000MW)

Based on this fact, a small deviation in calculating fuel consumption can lead to a significant error in estimating the actual operating cost. This section shows that the existing polynomial-based fuel-cost functions cannot explain the exact variability of the fuel-cost relation if the corresponding unit is running but not connected to the grid yet. Thus, the main goal of this section is to correct the existing fuel-cost functions of thermal units so that the operating costs of uncommitted running units are also considered. It proposes a piecewise fuel-cost function to calculate the operating cost based on the unit status; whether it is connected to the grid or not. Different regression-based models are investigated to determine the most appropriate approximation. They are analyzed based on a real dataset gathered from a **distributed control system** (**DCS**) of a single-fuel 75 MW base-load GT.

7.2.1 The Technical Problem Associated with the Classical Fuel-Cost Models

As said before, the classical fuel-cost functions are formulated in Chapter 3. The deficiency of these functions occurs when the ith unit is running but unconnected to the grid. That is, the ith unit is consuming fuel with zero production of P_i and Q_i. With classical fuel-cost functions, the current answer is to set $P_i = Q_i = 0$ in C_i^a and C_i^r equations, so:

$$C_i^a\left(P_i\right) = \alpha_0 \tag{7.1}$$

$$C_i^r\left(Q_i\right) = \beta_0 \tag{7.2}$$

These two intercepts (i.e., α_0 and β_0) only explain the active and reactive operating costs during the **full-speed no-load** (**FSNL**) condition. Unfortunately, the thermal units can be operated at different speeds. This means that the fuel consumption is not constant when $P_i = Q_i = 0$. This serious claim leads us to investigate more about this hidden phenomenon that could be faced in many real electric power stations. First, let's consider the simplified version of the General Electric Mark V control system shown in Figure 7.1 [2].

As can be seen from the preceding simplified schematic diagram, six essential control modes are available to operate GTs: 1. temperature, 2. speed, 3. acceleration rate, 4. start-up, 5. shut-down, and 6. manual. This diagram can be further

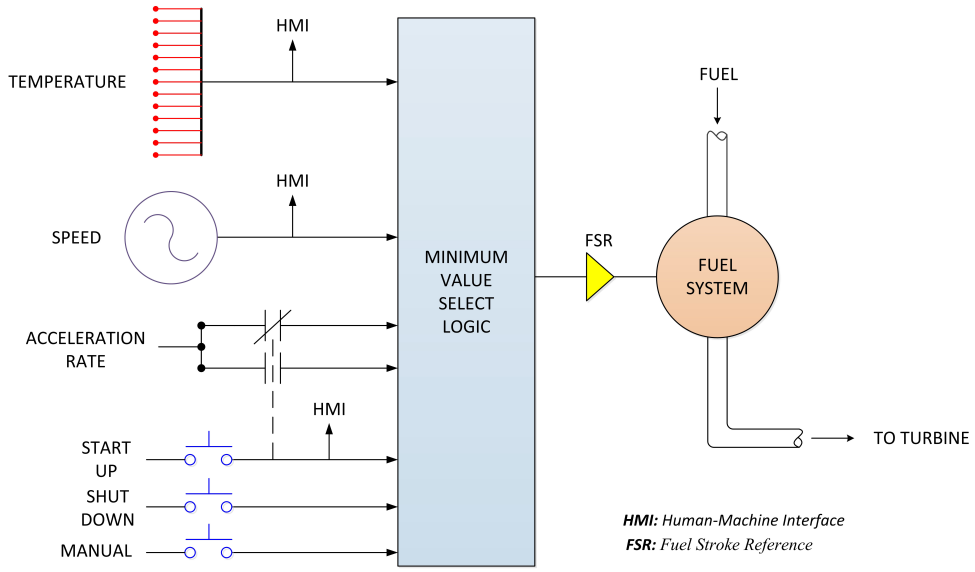


Figure 7.1: Simplified Schematic Diagram of GE Mark V Control System

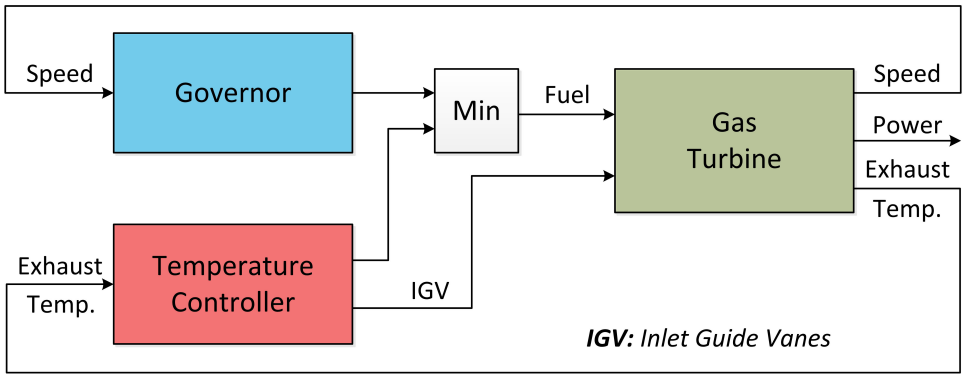


Figure 7.2: Simplified Governor and Temperature Control Loops

simplified by focusing only on the temperature and speed controls as explained in [62] and depicted in Figure 7.2.

In the literature, the cold and warm start-up operating costs are treated in the UC analysis. These two costs are explained in [386]. Also, **original equipment manufacturers** (**OEMs**) have their own special formulas used to calculate what are called **factored hours**, **equivalent operating hours** (**EOH**), and **equivalent operating cycles** (**EOC**) to define the maintenance interval of each machine [188, 190, 380]. These formulas can be embedded to enhance the operating cost calculation accuracy because the start-up and shut-down phases are not equal to the normal

operating time, and thus more weights must be given to the operating costs of these two phases. This can be observed by looking at the equations of the cold and warm start-up operating costs given in [386].

The logical question that might be raised here is whether the technical problem explained in the first paragraphs of this subsection can be tackled by using these start-up/shut-down equations. The start-up operating cost of the ith unit can be explained if its equation is considered during searching for optimal operating solutions. However, this equation does not help if the ith unit passed its start-up phase but still not connected yet to the grid. Similarly, if the shut-down operating cost is mathematically formulated in the optimization problem, then it does not help when the ith unit is disconnected from the grid but the shut-down phase is not initiated yet. Therefore, there is a need to correct the existing fuel-cost equations to consider the operating cost of the ith unit during both the connection and disconnection conditions.

7.2.2 Proposed Spinning Reserve Mathematical Models

The control diagrams shown in Figures 7.1 and 7.2 reveal a fact that the independent variables P_i and Q_i can be used to calculate C_i^a and C_i^r, respectively, if the ith turbine is maintained at FSNL and both P_i and Q_i are equal to or bigger than 0. However, this statement is not correct if the preceding ith unit is operated by a mixture of multiple fuels[4] [11].

One of the possible approaches is to use piecewise functions to calculate the operating costs of the ith unit as follows:

$$C_i^a\left(\varpi_i, P_i\right) = \begin{cases} f_i^a\left(\varpi_i\right), & \varpi_i \neq \text{FSNL or } P_i < 0 \\ g_i^a\left(P_i\right), & \text{otherwise} \end{cases} \tag{7.3}$$

$$C_i^r\left(\varpi_i, Q_i\right) = \begin{cases} f_i^r\left(\varpi_i\right), & \varpi_i \neq \text{FSNL or } Q_i < 0 \\ g_i^r\left(Q_i\right), & \text{otherwise} \end{cases} \tag{7.4}$$

where f_i^a and f_i^r are respectively the active and reactive fuel-cost functions of the ith turbine when it is not connected to the grid yet. g_i^a and g_i^r are respectively the active and reactive fuel-cost functions when the ith turbine is connected to the grid. ϖ_i is the mechanical speed of the ith turbine; which is commonly measured in **revolutions per minute** (**r.p.m.**).

[4] It will be explained later in a separate section of this chapter.

The second term of each piecewise function means:

- $g_i^a\left(P_i\right) = C_i^a\left(P_i\right)$ that is presented in (3.1), (3.3), (3.5), (3.7), (3.9), and (3.11).
- $g_i^r\left(Q_i\right) = C_i^r\left(Q_i\right)$ that is presented in (3.2), (3.4), (3.6), (3.8), (3.10), and (3.12).

Be note that the piecewise functions presented in (7.3) and (7.4) are not related to those presented in (3.11) and (3.12). The purpose of the former piecewise functions is to linearize the curves of $C_i^a\left(P_i\right)$ and $C_i^r\left(Q_i\right)$ by using multiple first-order polynomial equations. Thus, if this is the case of $C_i^a\left(P_i\right)$, then (7.3) becomes:

$$C_i^a\left(\varpi_i, P_i\right) = \begin{cases} f_i^a\left(\varpi_i\right) = \begin{cases} \zeta_{1,0} + \zeta_{1,1}\ \varpi_i; & \varpi_i \in \left[\varpi_i^{\min}, \varpi_i^{Z_1^{\min}}\right) \\ \zeta_{2,0} + \zeta_{2,1}\ \varpi_i; & \varpi_i \in \left[\varpi_i^{Z_1^{\min}}, \varpi_i^{Z_1^{\max}}\right) \\ \zeta_{3,0} + \zeta_{3,1}\ \varpi_i; & \varpi_i \in \left[\varpi_i^{Z_1^{\max}}, \varpi_i^{Z_2^{\min}}\right), \\ \quad\vdots & \\ \zeta_{m,0} + \zeta_{m,1}\ \varpi_i; & \varpi_i \in \left[\varpi_i^{Z_m^{\max}}, \varpi_i^{\max}\right] \end{cases} & \begin{array}{c} \varpi_i \neq \text{FSNL} \\ \text{or} \\ P_i < 0 \end{array} \\ g_i^a\left(P_i\right) = \begin{cases} \alpha_{1,0} + \alpha_{1,1} P_i; & P_i \in \left[P_i^{\min}, P_i^{Z_1^{\min}}\right) \\ \alpha_{2,0} + \alpha_{2,1} P_i; & P_i \in \left[P_i^{Z_1^{\min}}, P_i^{Z_1^{\max}}\right) \\ \alpha_{3,0} + \alpha_{3,1} P_i; & P_i \in \left[P_i^{Z_1^{\max}}, P_i^{Z_2^{\min}}\right), \\ \quad\vdots & \\ \alpha_{k,0} + \alpha_{k,1} P_i; & P_i \in \left[P_i^{Z_k^{\max}}, P_i^{\max}\right] \end{cases} & \text{otherwise} \end{cases} \tag{7.5}$$

A similar thing can be applied to (7.4). The goal of this study is to find a proper regression-based model of f_i^a using a real GT dataset. To accurately and precisely develop a fuel-cost function for uncommitted running thermal units, the following possible polynomial regression models are investigated:

$$\dot{V}\left(\varpi\right) = \gamma_0 + \gamma_1\ \varpi \tag{7.6}$$

$$\dot{V}\left(\varpi\right) = \gamma_0 + \gamma_1\ \varpi + \gamma_2\ \varpi^2 \tag{7.7}$$

$$\dot{V}\left(\varpi\right) = \gamma_0 + \gamma_1\ \varpi + \gamma_2\ \varpi^2 + \gamma_3\ \varpi^3 \tag{7.8}$$

where $\dot{V}$ is the fuel volumetric flow-rate and $\{\gamma_0, \gamma_1, \gamma_2, \gamma_3\}$ are the linear polynomial coefficients of the regression models.

Table 7.2: Technical Specifications of the Real Single-Fuel-Based Gas Turbine Used in This Experiment

Manufacturer	BBC[a]
GT Model	GT13DM
Fuel Type	Khuff Gas
Fuel Pressure	$\approx$ 21 bar
Rated Speed	3000 r.p.m. (FSNL)
Generator Model	3-Phase WG19L-080LL
Rated Power	125 MVA
Rated Voltage	10.5 kV
Rated Frequency	50 Hz

[a]BBC was a Swiss group of electrical engineering companies. In 1988, BBC merged with a Swedish industrial company called ASEA to form ABB Group. In 2000, ABB's conventional systems of power generation was sold to a French multinational company called ALSTOM. In the late of 2015, the energy division of ALSTOM was acquired by General Electric (GE) - an American multinational conglomerate corporation.

7.2.3 Numerical Experiment and Discussion

Table 7.2 shows the technical specifications of the real turbo-generator used in this numerical experiment. Figure 7.3 is a real photograph of the gas turbine used in this experiment. By logging-in the "**performance test data**" page of the DCS shown in Figure 7.4, a real dataset composed of 52 samples is collected from one **human-machine interface** (**HMI**) located in the **main control room** (**MCR**). Figure 7.5 shows three scatter-plots to describe the relationship between the mechanical speed of the **prime-mover** (in r.p.m.) and the volumetric flow-rate of the fuel gas (in **normal cubic meter per hour**, $\mathbf{Nm^3/h}$ $\rightarrow$ normal means: 0°C and 1 atm).

The curves of Figures 7.5b-7.5c are almost identical. This duplicated pattern happens because the relationship between the turbine speed and the operating time is almost linear; as obviously observed in Figure 7.5a.

There are two main methods to calculate the fuel price per hour (\$/h), either by measuring its: 1. volumetric flow-rate, or 2. heat-rate. If the fuel chemical and physical properties are given, then Nm^3/h can be directly converted to the **British thermal unit per hour** (**Btu/h**). Therefore, the cost function f^a can be easily

Figure 7.3: Real Photograph of the 125 MVA Gas Turbine Used in This Experiment

derived if the fuel volumetric flow-rate $\dot{V}$ is precisely and accurately determined by knowing the turbine speed ϖ; as described in (7.6)-(7.8).

If the fuel-cost curve is taken as the operating cost of the ith unit, then $f^a(\varpi)$ can be obtained as [11]:

$$f^a(\varpi) = \xi \, \dot{V}(\varpi) \tag{7.9}$$

where ξ is the fuel price rate in \$/m^3. It can also be converted in case the fuel cost is calculated based on \$/Btu criterion instead of \$/m^3. Other interchangeable units can also be used.

The preceding equation can be further modified to:

$$f^a(\varpi) = \phi + \varphi \left[\xi \, \dot{V}(\varpi) \right] \tag{7.10}$$

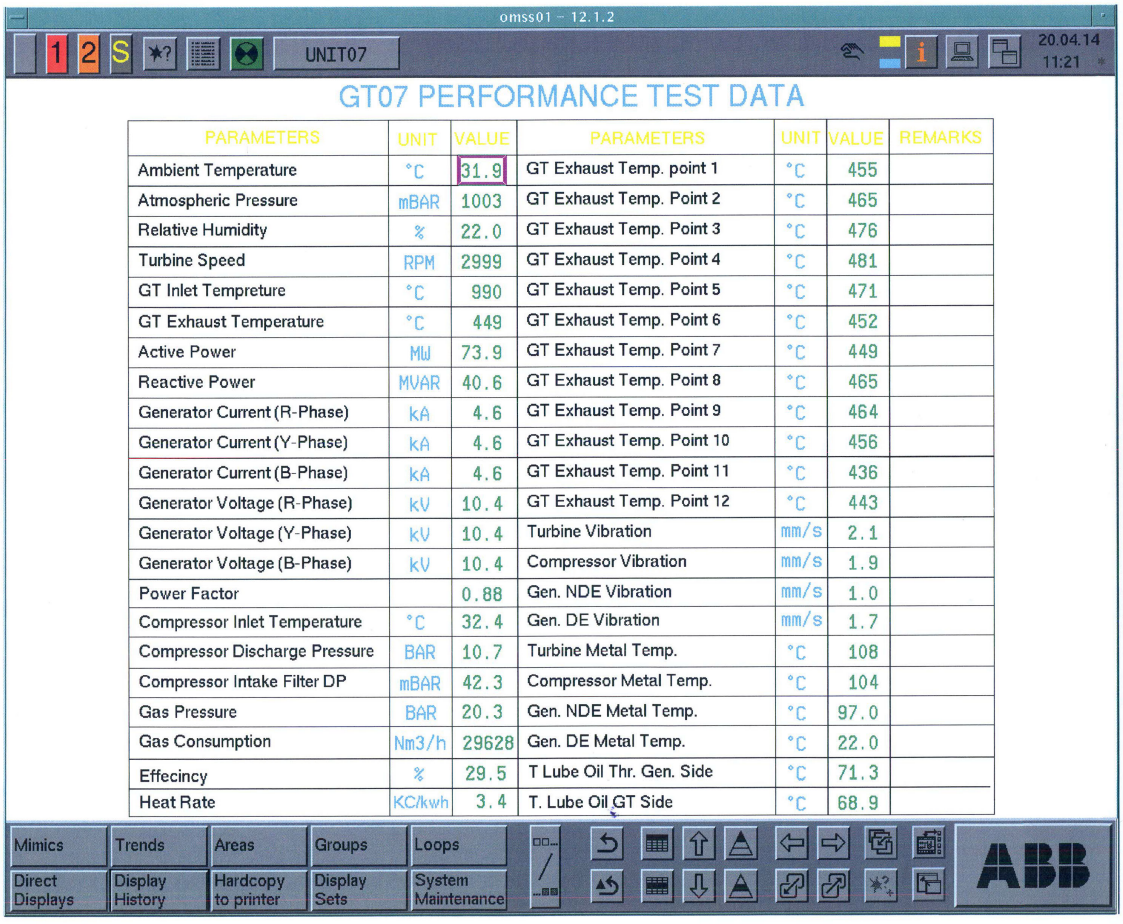
where ϕ is the initial cost in \$/h and φ is the operational pricing gain "unitless".

Or, it can even be expanded to be in a quadratic form of $\dot{V}(\varpi)$ as follows:

$$f^a(\varpi) = \phi + \varphi \left[\xi \, \dot{V}(\varpi) \right] + \psi \left[\xi \, \dot{V}(\varpi) \right]^2 \tag{7.11}$$

where ψ is the second-order pricing coefficient in h/\$.

Thus, the mystery key here is $\dot{V}$. Knowing this variable leads to determining the operating cost of the turbo-generator during both connection and disconnection

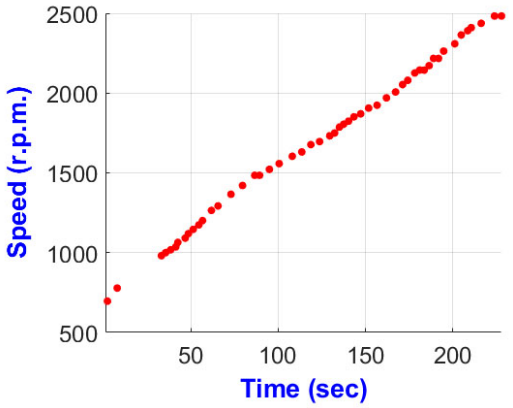


PARAMETERS	UNIT	VALUE	PARAMETERS	UNIT	VALUE	REMARKS
Ambient Temperature	°C	31.9	GT Exhaust Temp. point 1	°C	455	
Atmospheric Pressure	mBAR	1003	GT Exhaust Temp. Point 2	°C	465	
Relative Humidity	%	22.0	GT Exhaust Temp. Point 3	°C	476	
Turbine Speed	RPM	2999	GT Exhaust Temp. Point 4	°C	481	
GT Inlet Tempreture	°C	990	GT Exhaust Temp. Point 5	°C	471	
GT Exhaust Temperature	°C	449	GT Exhaust Temp. Point 6	°C	452	
Active Power	MW	73.9	GT Exhaust Temp. Point 7	°C	449	
Reactive Power	MVAR	40.6	GT Exhaust Temp. Point 8	°C	465	
Generator Current (R-Phase)	kA	4.6	GT Exhaust Temp. Point 9	°C	464	
Generator Current (Y-Phase)	kA	4.6	GT Exhaust Temp. Point 10	°C	456	
Generator Current (B-Phase)	kA	4.6	GT Exhaust Temp. Point 11	°C	436	
Generator Voltage (R-Phase)	kV	10.4	GT Exhaust Temp. Point 12	°C	443	
Generator Voltage (Y-Phase)	kV	10.4	Turbine Vibration	mm/s	2.1	
Generator Voltage (B-Phase)	kV	10.4	Compressor Vibration	mm/s	1.9	
Power Factor		0.88	Gen. NDE Vibration	mm/s	1.0	
Compressor Inlet Temperature	°C	32.4	Gen. DE Vibration	mm/s	1.7	
Compressor Discharge Pressure	BAR	10.7	Turbine Metal Temp.	°C	108	
Compressor Intake Filter DP	mBAR	42.3	Compressor Metal Temp.	°C	104	
Gas Pressure	BAR	20.3	Gen. NDE Metal Temp.	°C	97.0	
Gas Consumption	Nm3/h	29628	Gen. DE Metal Temp.	°C	22.0	
Effecincy	%	29.5	T Lube Oil Thr. Gen. Side	°C	71.3	
Heat Rate	KC/kwh	3.4	T. Lube Oil GT Side	°C	68.9	

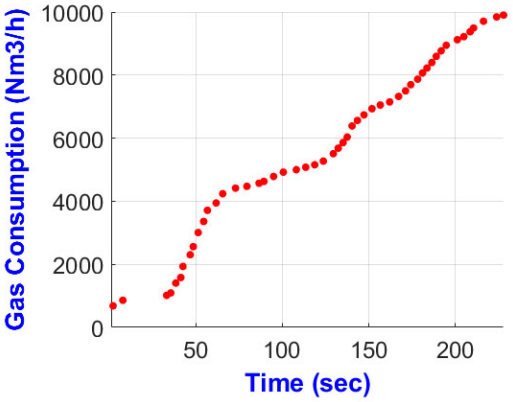
Figure 7.4: Actual Input/Output Readings of the 125 MVA Gas Turbine Monitored by ABB's ProControl-14 DCS

conditions. If $\dot{V}$ is known, then there is no need to do any regression analysis, and thus any one of (7.9)-(7.11) can be used directly. However, in case $\dot{V}$ is unknown, inaccessible, restricted, or if it belongs to external suppliers, then there is a need to do a regression analysis to correctly and precisely estimate $\dot{V}$ based on another highly correlated variable(s). In this study, the turbine mechanical speed ϖ is used for this purpose. From (7.9)-(7.11), because both $f^a(\varpi)$ and $\dot{V}(\varpi)$ are functions of ϖ, so $f^a(\varpi)$ can be directly expressed as a function of ϖ instead of $\dot{V}(\varpi)$. This option can also be used if we do not want to estimate the operating cost based on an estimated volumetric flow-rate (i.e., double estimation). However, real power stations can only estimate f^a if $\dot{V}$ is known[5]. Practically speaking, real power stations can

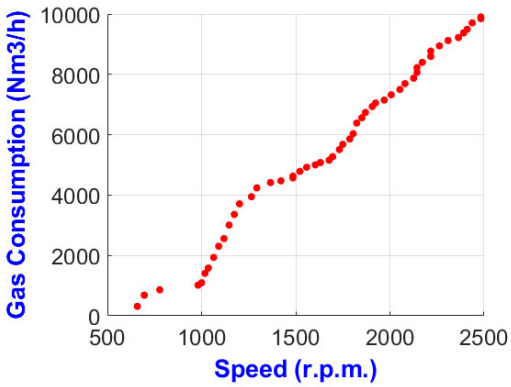
[5]It has to be said that f^a can also be estimated if Btu/h is known. However, the reading Btu/h



(a) Speed vs. Time



(b) Volumetric Flow-Rate vs. Time



(c) Volumetric Flow-Rate vs. Speed

Figure 7.5: Scatter-Plots Between Operating Time, Turbine Speed, and Fuel Volumetric Flow-Rate ($P = Q = 0 \; \forall$ samples)

only calculate $\dot{V}$ from fuel **flowmeters**[6]. Thus, in both approaches, f^a depends on $\dot{V}$. This means that without $\dot{V}$, the actual response of f^a cannot be determined, and thus it is hard to regress f^a by taking ϖ as a predictor.

The polynomial equations given in (7.6)-(7.8) are regressed using MATLAB 2017b. The results obtained from these three regression analyses are shown in Figure 7.6. It is obvious that the fitted linear equation shown in Figure 7.6a is the most significant regression model. The evidence is: 1. the smallest tStat's p-values, 2. the smallest root mean squared error (RMSE), 3. the biggest adjusted R-squared (R^2_{adj}), 4. the lowest **degrees of freedom** (**DF**), and 5. its F-statistic scored the biggest value with the lowest p-value[7]. Therefore, (7.6) is selected to estimate $\dot{V}$ based on ϖ. The regression analysis shows the following numerical result:

$$\dot{V}(\varpi) = -3491.9 + 5.404\ \varpi \tag{7.12}$$

The actual data, fitted curve and confidence bounds are shown in Figure 7.7. To check whether the residuals follow the normal distribution or not, the Anderson-Darling's test is applied using the following two hypotheses:

H_0 : *the residuals follow the normal distribution*, versus

H_1 : *the residuals do not follow the normal distribution*

The p-value of this test is 0.1096, which means that it fails to reject the null hypothesis H_0. Thus, the normality test is passed.

As per [128], the US natural gas industrial price is 3.89 USD per thousand cubic feet (3.89 \$/1000 ft^3) for September, 2017[8]. Thus, with 1 ft = 0.3048 m, ξ is 0.137374054 \$/m^3. For that price, substituting (7.12) in (7.9) for $\dot{V}$ yields:

$$\begin{aligned} f^a(\varpi) &= 0.137374054\,(-3491.9 + 5.404\ \varpi) \\ &= -479.6965 + 0.7424\ \varpi \end{aligned} \tag{7.13}$$

Suppose that, at least, three technicians (instrumentation and control (C&I), electrical, and mechanical) and one operator should attend to safely operate that turbo-generator. If their total average pay is: \$(29.40 + 20.68 + 20.09 + 28.71) per

itself cannot be determined if $\dot{V}$ is unknown.

[6]These volumetric flow-rate signals are sent from the field to DCS via **flow transmitters**.

[7]If it is compared with the constant model: $\dot{V} = \gamma_0$.

[8]It is the same data used in our study published in [16].

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)    -3491.9       160.24    -21.792    3.4933e-27
    x1               5.404     0.091013     59.376    4.9437e-48


Number of observations: 52, Error degrees of freedom: 50
Root Mean Squared Error: 332
R-squared: 0.986,  Adjusted R-Squared 0.986
F-statistic vs. constant model: 3.53e+03, p-value = 4.94e-48
```

(a) Linear Regression Analysis

```
Linear regression model:
    y ~ 1 + x1 + x1^2

Estimated Coefficients:
                   Estimate         SE         tStat       pValue
                   ___________    ___________    _______    __________

    (Intercept)       -3367.6         477.16    -7.0576    5.4021e-09
    x1                 5.2349        0.61725      8.481    3.5466e-11
    x1^2           5.1857e-05    0.00018723    0.27698       0.78296


Number of observations: 52, Error degrees of freedom: 49
Root Mean Squared Error: 335
R-squared: 0.986,  Adjusted R-Squared 0.985
F-statistic vs. constant model: 1.73e+03, p-value = 3.56e-46
```

(b) Quadratic Regression Analysis

```
Linear regression model:
    y ~ 1 + x1 + x1^2 + x1^3

Estimated Coefficients:
                   Estimate          SE         tStat       pValue
                   ___________    ___________    _______    _________

    (Intercept)       -3218.5           1331     -2.418     0.01945
    x1                 4.9111         2.7654     1.7759    0.082086
    x1^2           0.0002673      0.0018025     0.1483      0.88273
    x1^3          -4.4688e-08    3.7181e-07    -0.12019     0.90484


Number of observations: 52, Error degrees of freedom: 48
Root Mean Squared Error: 339
R-squared: 0.986,  Adjusted R-Squared 0.985
F-statistic vs. constant model: 1.13e+03, p-value = 1.67e-44
```

(c) Cubic Regression Analysis

Figure 7.6: Results of the Three Polynomial Regression Models

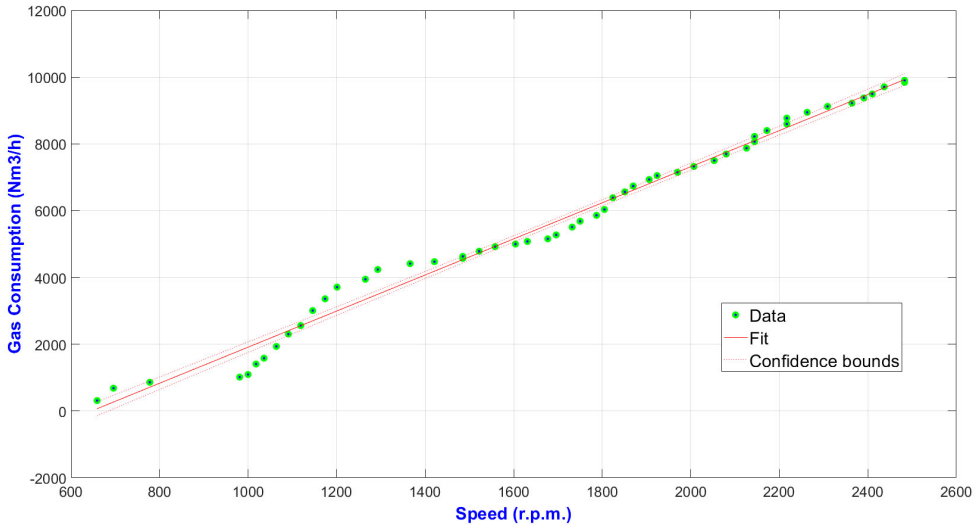


Figure 7.7: Actual Data, Fitted Curve, and Confidence Bounds of the 1st Order Regression Model Given in (7.6)

hour [303], then $\phi = 98.88$ \$/h for (7.10). Assume that the operational pricing gain $\varphi = 1.25$ is used to account the interest, depreciation, maintenance cost, etc, then (7.10) becomes:

$$f^a(\varpi) = 98.88 + 1.25\,(-479.6965 + 0.7424\ \varpi) = -500.7406 + 0.928\ \varpi \tag{7.14}$$

If $\psi = 4 \times 10^{-5}$ h/\$ is used, then (7.11) can be applied to calculate the operating cost[9] as follows:

$$\begin{aligned} f^a(\varpi) &= 98.88 + 1.25\,(-479.6965 + 0.7424\ \varpi) + 4 \times 10^{-5}\,(-479.6965 + 0.7424\ \varpi)^2 \\ &= -491.5363 + 0.8995\ \varpi + 2.2046 \times 10^{-5}\ \varpi^2 \end{aligned} \tag{7.15}$$

The plots of (7.13)-(7.15) are shown in Figure 7.8. If this turbo-generator is considered as the ith unit, then any one of (7.13)-(7.15) can be substituted in (7.3) for $f_i^a(\varpi_i)$. Thus, the ith operating cost C_i^a can be calculated during both the connection and disconnection conditions. Suppose that (7.11), which is numerically determined in (7.15) using (7.12), is adopted with (3.7) to calculate the operating cost of the ith running unit. Based on this, if (7.6) is selected, then (7.3) can be modified to:

$$C_i^a(\varpi_i, P_i) = \begin{cases} \phi + \varphi \cdot \xi\,(\gamma_0 + \gamma_1 \varpi) + \psi \cdot \xi^2\,(\gamma_0 + \gamma_1 \varpi)^2, & \varpi_i \neq \text{FSNL or } P_i < 0 \\ \alpha_0 + \alpha_1 P_i + \alpha_2 P_i^2, & \text{otherwise} \end{cases} \tag{7.16}$$

[9] ψ could be defined to cover the network losses and other negligible and unconsidered costs.

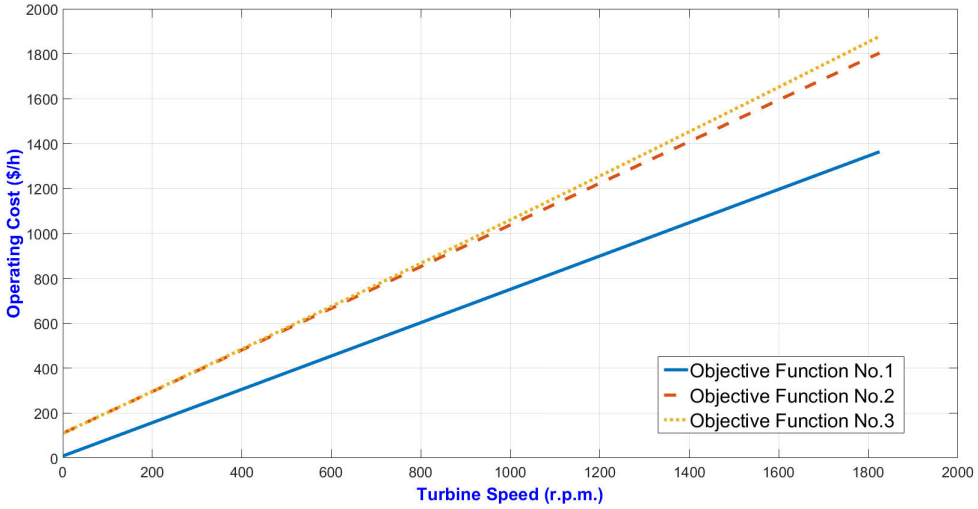


Figure 7.8: Graphical Representation of $f^a(\varpi)$ Using (7.13)-(7.15)

7.3 Revisiting ELD Models from Real Power Stations Point of View

Based on our observations in different real power stations, the existing ELD optimization models cannot be used; or, in other words, they are not well-compatible with actual equipment. The generating units of most, if not all, thermal power stations are synchronized and connected to some common busbars to feed their grids with the required electricity. Sometimes, if a power station is small or if it contains just small-capacity units, then the whole electricity produced by that power station is fed to the network via just one single busbar. Yes, these units could be treated as a single equivalent unit [123]. But, it is well known that each unit has its own characteristics and specifications. Even if all these units are from the same manufacturer[10], these units have different performance due to many practical reasons, such as [7, 11, 196]:

1. Fuel assigned to each machine could be gas, distillate, or a mixture of both.
2. Even if only fuel gas is used, it could be a floating ratio of different fuel gases available in the market, such as a mixture of **Khuff** and **Residual gases**. Thus, if fuel mixtures are applied to some units, then the classical *one*-dimensional fuel-cost functions cannot be used anymore[11].

[10] For example, GE, SIEMENS, Mitsubishi Hitachi, Ansaldo Energia, etc.
[11] This essential point will be discussed later, in the next subsection, in more detail.

3. Some units could be just commissioned from a **major overhaul** (or **C-inspection**), while some others could be weary or need some maintenance jobs.
4. Some units could not be operated to their full capacity due to defective burners, unaligned prime-mover, unwell coated blades, vibration, the combustor behavior monitored by the **pulsation probe**, etc.
5. Maximum power output is suppressed due to winding resistance or external problems with other equipment, like its **cooling system**, **excitation system**, or **step-up transformer**.
6. Some units are forced to be stopped or partially operated based on strict orders received from the planning department and/or the **system control**.

Based on this, it is impossible to consider these synchronized units as a single equivalent unit to make real networks compatible with the existing mathematical optimization models. If so, then the error definitely will exist, and there will be a doubt about the feasibility and optimality of the results obtained from these incorrectly coded units.

This section tries to formulate a more realistic mathematical model to ensure that the ELD solutions are compatible with the real configurations of modern power stations. Also, it tries to cover some hidden practical constraints and phenomena faced in real-world problems. Such an essential adjustment can effectively reduce the total technical challenges that might be faced in EMS if someone wants to apply the existing classical ELD optimizers. Thus, it can be said that this section has two main contributions, which are summarized as follows:

1. The ELD design function is revised and corrected to consider many real technical problems.
2. Some hidden operational and administrative restrictions/obstacles are modeled as new constraints.

7.3.1 Corrected Formulation of ELD Problems

The classical ELD model is formulated in Chapter 3. The first point that has to be seriously taken into account is the fuel itself. Each ith unit could have a fuel type different from others. This means that the ith unit can have multiple fuel-cost curves.

If J fuels are assigned to the ith unit, then the operating cost of the ith unit should be converted from being a scalar value to a vector. Without considering the valve-point loading effect, this operating cost can be mathematically expressed as follows:

$$\bar{C}_i\left(P_i\right) = \min\left[\bar{C}_{i,1}(P_{i,1}), \bar{C}_{i,2}(P_{i,2}), \cdots, \bar{C}_{i,J}(P_{i,J})\right] \tag{7.17}$$

Let's assume that the jth fuel is the best type. Thus, similar to (3.14), the sinusoidal term should be included if the ith unit is realized with the valve-point loading effect as follows:

$$C_i\left(P_i\right) = \bar{C}_i\left(P_i\right) + \left|e_{i,j} \times \sin\left[f_{i,j} \times \left(P_{i,j}^{\min} - P_{i,j}\right)\right]\right| \tag{7.18}$$

where $e_{i,j}$ and $f_{i,j}$ are the valve-point loading effect coefficients of the ith unit operated by the jth fuel type.

By referring to the preceding fuel-cost functions and the single-fuel cost functions presented in Chapter 3, the classical multi-fuel cost function of the ith unit operated by J fuels can be formulated as follows:

$$C_i = \begin{cases} \alpha_{i,1} + \beta_{i,1}P_{i,1} + \gamma_{i,1}P_{i,1}^2 + \left|e_{i,1}\sin\left[f_{i,1}\left(P_{i,1}^{\min} - P_{i,1}\right)\right]\right|, & P_{i,1}^{\min} \leqslant P_{i,1} \leqslant P_{i,1}^{\max} \\ \alpha_{i,2} + \beta_{i,2}P_{i,2} + \gamma_{i,2}P_{i,2}^2 + \left|e_{i,2}\sin\left[f_{i,2}\left(P_{i,1}^{\min} - P_{i,2}\right)\right]\right|, & P_{i,2}^{\min} \leqslant P_{i,2} \leqslant P_{i,2}^{\max} \\ \vdots & \\ \alpha_{i,J} + \beta_{i,J}P_{i,J} + \gamma_{i,J}P_{i,J}^2 + \left|e_{i,J}\sin\left[f_{i,J}\left(P_{i,J}^{\min} - P_{i,J}\right)\right]\right|, & P_{i,J}^{\min} \leqslant P_{i,J} \leqslant P_{i,J}^{\max} \end{cases} \tag{7.19}$$

It is expressed as a piecewise function to explain the variability of P_i due to the variation in the **chemical composition**. This simplified multi-fuel cost function works fine if all the units are fed with one fuel type. However, based on our long observations in different real power stations, it has been found that many thermal generating units are operated by a mixture of multiple fuels. This phenomenon can be graphically described in Figure 7.9. If the ith unit is fed with a mixture of M fuels, then the preceding hidden phenomenon can be translated into the following mathematical model:

$$P_i = f\left(\dot{V}_i^{M_i}\right) \quad , \quad \text{where} \quad M_i \leqslant J \tag{7.20}$$

where $\dot{V}_i^{M_i}$ is the total volumetric flow-rate of the M_i fuels assigned to the ith unit to generate P_i megawatts.

Realistically speaking, each ith unit could be operated by multiple fuels, and these fuels could be different from those of other units. For the sake of simplicity, suppose

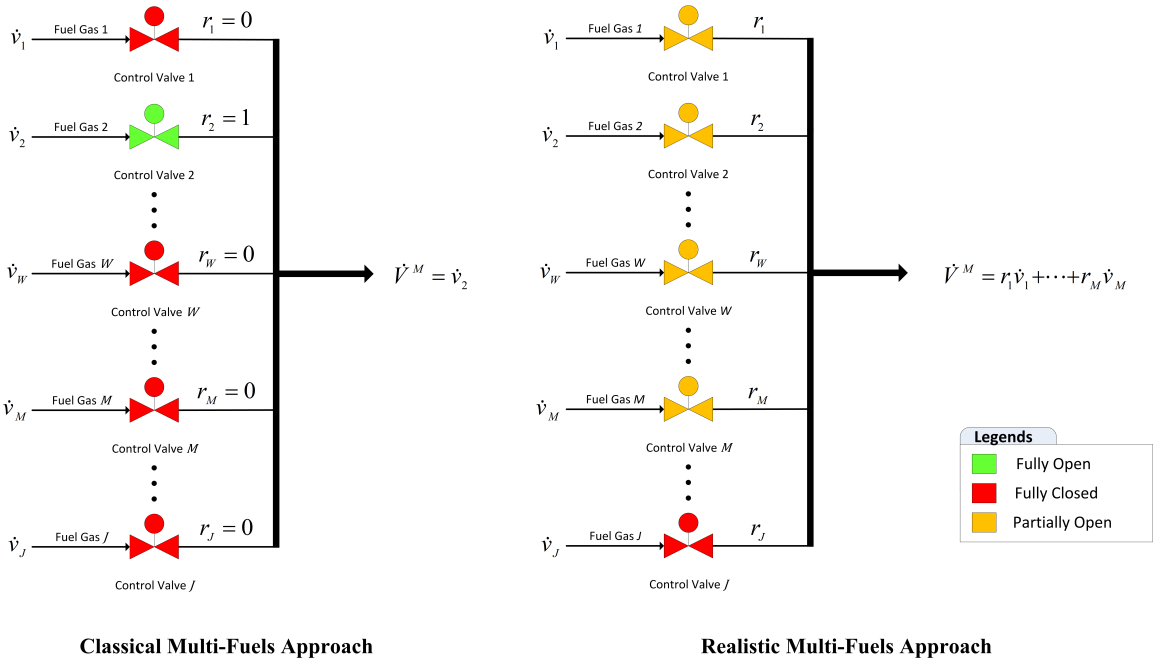


Figure 7.9: Real Piping and Instrumentation Diagram (P&ID) of Multi-Fuel Units

that all the multi-fuel units are operated by the same group of fuels denoted by M (i.e., $M = M_1 = M_2 = \cdots = M_i = \cdots = M_n$) and $M = J$. Based on Figure 7.9, $\dot{V}_i$ can be calculated for J fuels as follows:

$$\dot{V}_i = r_{i,1}\dot{v}_{i,1} + r_{i,2}\dot{v}_{i,2} + \cdots + r_{i,W}\dot{v}_{i,W} + \cdots + r_{i,J}\dot{v}_{i,J} \tag{7.21}$$

where $\dot{v}_{i,W}$ is the volumetric flow-rate of the Wth fuel assigned to the ith unit, and $r_{i,W}$ is the ratio of the Wth fuel selected for that unit. Thus, it is important to ensure that the sum of all the ratios equals one:

$$r_{i,1} + r_{i,2} + \cdots + r_{i,W} + \cdots + r_{i,J} = 1 \tag{7.22}$$

It has to be said that each fuel mixture has its own chemical composition. This fact is very important because it tells us that the relation is nonlinear even with mixtures made of just two fuel gases. This claim is proved via an experiment conducted on a real GT[12]. Thus, for this realistic situation, (7.19) is not valid anymore.

This study focuses on just two fuel gases supplied from two different sources with different properties. Therefore, substituting (7.22) in (7.21) for these two fuel gases

[12]It will be seen later.

yields:

$$\begin{aligned} \dot{V}_i &= r_{i,1}\dot{v}_{i,1} + r_{i,2}\dot{v}_{i,2} \\ &= \begin{cases} r_i\dot{v}_{i,1} + (1-r_i)\dot{v}_{i,2}, & \text{if} \quad r_{i,1} = r_i \\ (1-r_i)\dot{v}_{i,1} + r_i\dot{v}_{i,2}, & \text{if} \quad r_{i,2} = r_i \end{cases} \end{aligned} \tag{7.23}$$

From (7.23), three possible scenarios can be highlighted here as follows:

- **Binary Ratio** $\rightarrow$ $r_i = \{0, 1\}$:
  This is the classical multi-fuel ELD model presented in the literature [94, 216], which is the simplest scenario. It is hard to accept this scenario as a general multi-fuel-based ELD problem because the chance to have a binary ratio r_i for all power stations is very low.

- **Fixed Ratio** $\rightarrow$ $r_i \in [0, 1]$:
  This scenario is more realistic where P_i can be adjusted to its set-point by opening or closing the **control valves** (**CVs**) of both fuels, so their flow-rates (i.e., $\dot{v}_{i,1}$ and $\dot{v}_{i,2}$) can controlled by a predefined ratio r_i. But again, this scenario cannot be accepted as a general multi-fuel-based ELD problem because r_i is controlled by many factors. For instance, the current fuel price in the market, its availability and its burning quality have direct influences on r_i. Moreover, because the ratio r_i is a floating fraction, so there are almost semi-infinite fuel mixtures; if someone still wants to use the traditional expression explained in (7.19).

- **Variable Ratio** $\rightarrow$ $r_i \in [0, 1]$:
  This is the most realistic scenario that can cover all real multi-fuel-based ELD problems. The preceding two scenarios are just two special cases. Thus, this scenario should be considered for r_i of (7.23). Embedding it means a major improvement to the existing classical ELD mathematical models.

Equations (7.21)-(7.23) give a solid conclusion that infinite ratios between 0 and 1 can be used for the ith unit. Based on this, the dimension of the classical ELD model is open, and thus the vector lengths of $\{\alpha, \beta, \gamma, e, f\}$ are not constant anymore. Furthermore, each ratio r_i can affect the power output, operating cost, and emission

rates of the ith unit. Therefore, it is very hard to find an optimal solution to such a problem, and it needs a highly accurate formula from the corresponding original equipment manufacturer (OEM) to describe the exact relationship between the power output and the chemical composition of each fuel. For example, the natural gas has some percentages of methane, ethane, nitrogen, propane, carbon dioxide, butane, pentane, etc [137]. If the best fuel ratios of (7.21) are chosen to make (7.17) exist at C_i^* (P_i^*), then the objective function becomes:

$$\text{OBJ} = \min \sum_{i=1}^{n} C_i^* \left(P_i^*\right) \tag{7.24}$$

The second challenge that almost exists in all real power networks is about the busbars and whether they are energized by one individual unit or a group of synchronized units. Classical ELD optimizers work based on an assumption that each ith unit is connected to one specific busbar. Practically speaking, many, or even most of, units are connected to some common busbars as a group of synchronized units. For instance, a power station built based on two combined-cycle blocks, each powered by two gas turbines (160 MW GT - 13E2) and one steam turbine (160 MW ST - DKZ), is offered by ALSTOM[13]. Such power station can supply national grids by 960 MW through one common busbar (i.e., from both blocks) or two common busbars (i.e., from each block). The other example from the same company is a power station built based on two combined cycle blocks where each block contains three gas turbines (97 MW GT - 13D2) and one steam turbine (117 MW ST - DK50) as shown in Figure 7.10. Thus, total power of 816 MW can be supplied via one common busbar or two busbars with 408 MW for each. Figure 7.11 gives an additional illustration of this configuration. Some busbars could be energized by one or multiple units synchronized within an acceptable tolerance of frequency Δf.

Now, consider the general configuration of the ith power station shown in Figure 7.12. To make the classical ELD model valid, the power output shared by the ith power station is treated as a net power output supplied to the ith busbar after subtracting the total power consumed by the auxiliary equipment of the ith power

[13]French multinational company acquired by GE in November 2015

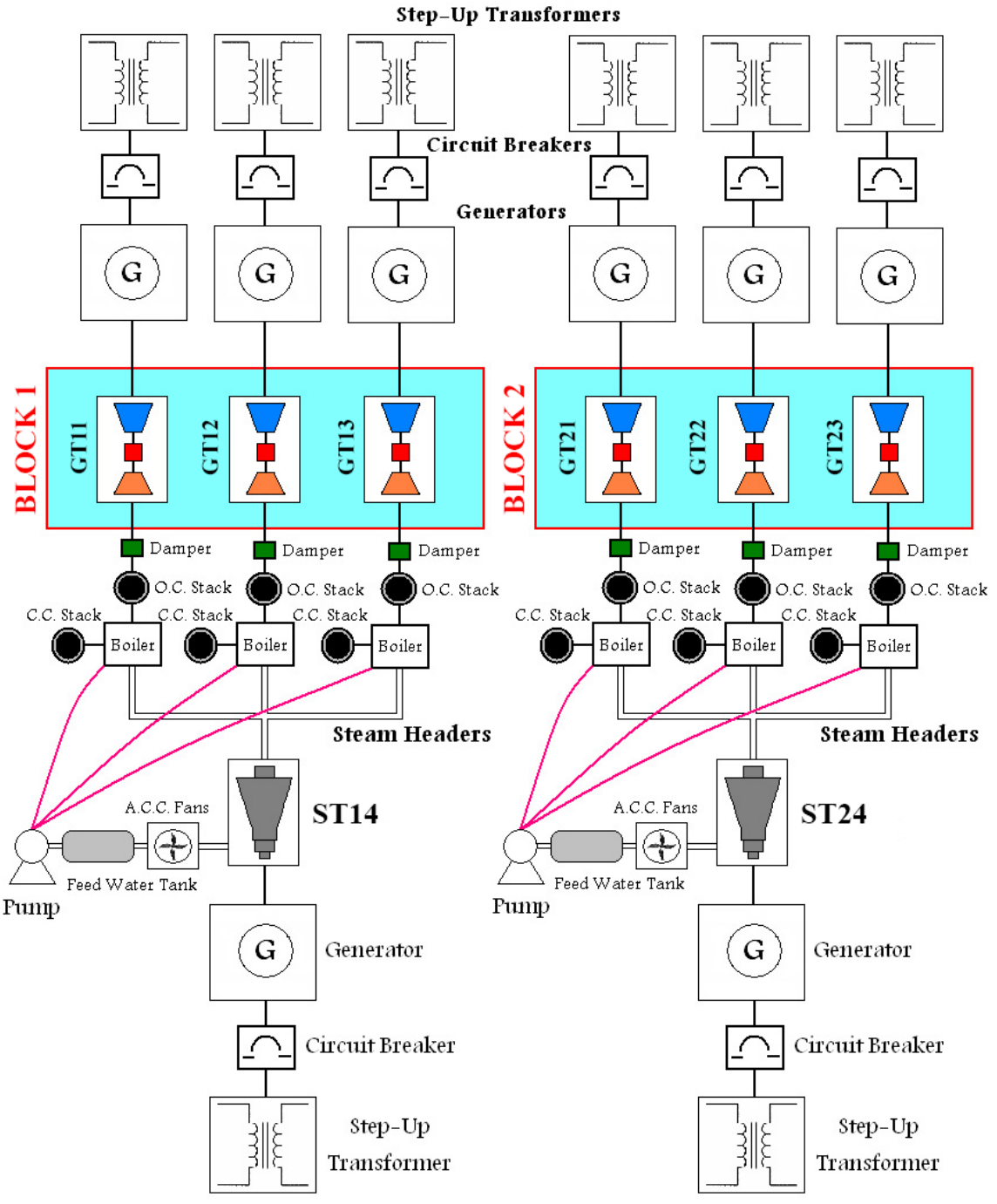


Figure 7.10: One of ALSTOM's CCPPs Designed Based on Two Blocks Containing 3× GTs and 1× ST Each

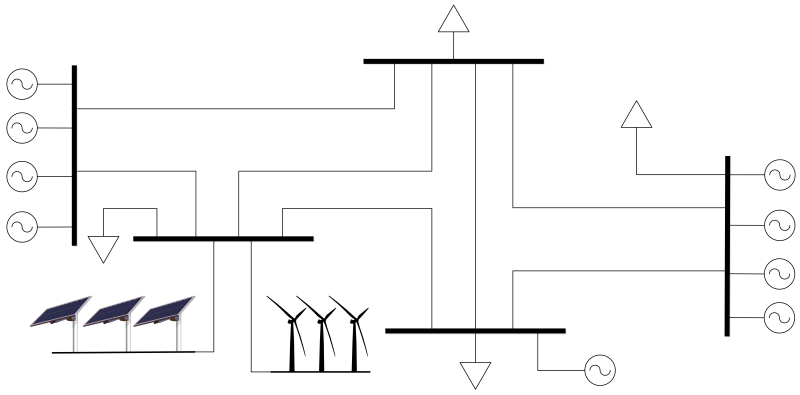


Figure 7.11: An Illustrated Real Electric Power System

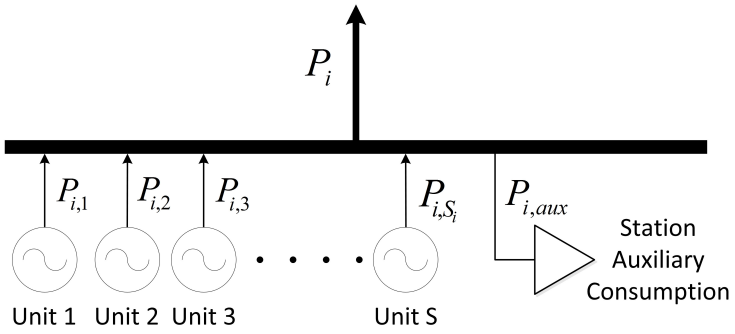


Figure 7.12: The Total Power Generated in the ith Power Station

station $P_{i,\text{aux}}$ as follows:

$$P_i = -P_{i,\text{aux}} + \sum_{k_i=1}^{S_i} P_{i,k_i} \tag{7.25}$$

Based on this new consideration, P_i could be just one unit connected to the ith busbar or a group of S_i units if the ith busbar is a power station output node. Thus, (7.24) becomes:

$$\text{OBJ} = \min \sum_{i=1}^{n} \sum_{k_i=1}^{S_i} C^*_{i,k_i}\left(P^*_{i,k_i}\right) \tag{7.26}$$

where n denotes the number of busbars energized by one or group of units, and S_i denotes the total number of units available in the ith power station. S_i equals 1 if only one unit is connected to the ith busbar. If all S_i of n busbars equal 1 (i.e., $S_1 = S_2 = \cdots = S_i = \cdots = S_n = 1$), then (7.26) is automatically converted to (7.24).

Consequently, if all S_i units have smooth quadratic fuel-cost functions and operated by the best fuel mixture, then the fuel-cost function can be expressed as follows:

$$\bar{C}_i^* \left(P_i^*\right) = \sum_{k_i=1}^{S_i} \bar{C}_{i,k_i}^* \left(P_{i,k_i}^*\right) = \sum_{k_i=1}^{S_i} \alpha_{i,k_i}^* + \beta_{i,k_i}^* P_{i,k_i}^* + \gamma_{i,k_i}^* P_{i,k_i}^{*^2} \tag{7.27}$$

Also, if the valve-point loading effect is considered for all the S_i units, then (7.18) can be modified to:

$$C_i^* \left(P_i^*\right) = \sum_{k_i=1}^{S_i} C_{i,k_i}^* \left(P_{i,k_i}^*\right) = \bar{C}_i^* \left(P_i^*\right) + \sum_{k_i=1}^{S_i} \left| e_{i,k_i}^* \sin\left[f_{i,k_i}^* \left(P_{i,k_i}^{*,\min} - P_{i,k_i}^*\right)\right]\right| \tag{7.28}$$

In addition, because S_i units are connected to the ith busbar, so (7.21) can be modified to:

$$\dot{V}_{i,k_i} = r_{i,k_i,1}\dot{v}_{i,k_i,1} + r_{i,k_i,2}\dot{v}_{i,k_i,2} + \cdots + r_{i,k_i,J}\dot{v}_{i,k_i,J} \ , \quad i : 1 \to n \ , \quad k_i : 1 \to S_i \tag{7.29}$$

From (7.28), both C_i^* and P_i^* are denoted by a star superscript because each variable can be affected by infinite sets of fuel ratios as seen before in (7.21) and (7.23). Thus, the best fuel ratio set, denoted by "$*$" is used to operate the k_ith unit. Based on this, the variable bounds given in (3.15) are modified as follows:

$$P_{i,k_i}^{*,\min} \leqslant P_{i,k_i}^* \leqslant P_{i,k_i}^{*,\max} \tag{7.30}$$

Substituting (7.25) in (3.17) at the optimal fuel mixture yields:

$$P_T = \sum_{i=1}^{n} \left[\left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) - P_{i,\text{aux}}^* \right] \tag{7.31}$$

Equation (7.31) can be simplified by adding $P_{i,\text{aux}}^*$ to the total demand P_D as shown in Figure 7.12. Thus, Kron's loss formula given in (3.18) is modified as follows:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) B_{ij} \left(\sum_{k_j=1}^{S_j} P_{j,k_j}^* \right) \right] + \sum_{i=1}^{n} \left[B_{0i} \left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) \right] + B_{00} \tag{7.32}$$

where S_i and S_j are the total units available in the ith and jth power stations, respectively.

7.3.1.1 Slack Generator vs. Slack Busbar

Based on the preceding equations and figures, the **slack generator** concept presented in the literature should be replaced with what can be called a "**slack busbar**[14]" because P_i could be a production of multiple units connected together to the ith busbar. Thus, two approaches can be used here to satisfy the equality constraint given in (3.16):

Slack Busbar

The net output shared by the ith power station is considered as one variable (i.e, $P_i^* = -P_{i,\text{aux}}^* + \sum_{k_i=1}^{S_i} P_{i,k_i}^*$). Thus, neglecting transmission losses P_L yields:

$$P_i^* = P_D + \sum_{j=1}^{n} P_{j,\text{aux}}^* - \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{k_j=1}^{S_j} P_{j,k_j}^* \tag{7.33}$$

If P_L is considered, then Kron's loss formula is used to modify (7.33) to:

$$\begin{aligned}
&\sum_{k_i=1}^{S_i} P_{i,k_i}^* - \sum_{j=1}^{n} \left[\left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) B_{ij} \left(\sum_{k_j=1}^{S_j} P_{j,k_j}^* \right) \right] - \sum_{\substack{j=1 \\ j\neq i}}^{n} \left[\left(\sum_{k_j=1}^{S_j} P_{j,k_j}^* \right) B_{ji} \left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) \right] \\
&- B_{0i} \left(\sum_{k_i=1}^{S_i} P_{i,k_i}^* \right) = P_D + \sum_{j=1}^{n} P_{j,\text{aux}}^* + \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{\substack{l=1 \\ l\neq i}}^{n} \left[\left(\sum_{k_j=1}^{S_j} P_{j,k_j}^* \right) B_{jl} \left(\sum_{k_l=1}^{S_l} P_{l,k_l}^* \right) \right] \\
&+ \sum_{\substack{j=1 \\ j\neq i}}^{n} \left[B_{0j} \left(\sum_{k_j=1}^{S_j} P_{j,k_j}^* \right) \right] + B_{00} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{k_j=1}^{S_j} P_{j,k_j}^*
\end{aligned} \tag{7.34}$$

Extracting P_i^* from the inner summations and grouping them to one side of the equation can convert (7.34) to a quadratic equation as follows:

$$a{P_i^*}^2 + bP_i^* + c = 0 \tag{7.35}$$

The analytical solution of (7.35) is:

$$P_i^* = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad b^2 - 4ac \leq 0 \tag{7.36}$$

[14] If the production of each power station is fed to the grid via one common busbar, then the slack busbar can also be called the "**slack power station**".

where $\{a, b, c\}$ can be respectively determined as follows:

$$a = -B_{ii} \tag{7.37}$$

$$b = 1 - \sum_{\substack{j=1 \\ j\neq i}}^{n} \left[B_{ij} \left(\sum_{k_j=1}^{S_j} P^*_{j,k_j} \right) \right] - \sum_{\substack{j=1 \\ j\neq i}}^{n} \left[\left(\sum_{k_j=1}^{S_j} P^*_{j,k_j} \right) B_{ji} \right] - B_{0i} \tag{7.38}$$

$$\begin{aligned} c = & \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{k_j=1}^{S_j} P^*_{j,k_j} - P_D - \sum_{j=1}^{n} P^*_{j,\text{aux}} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{\substack{l=1 \\ l\neq i}}^{n} \left[\left(\sum_{k_j=1}^{S_j} P^*_{j,k_j} \right) B_{jl} \left(\sum_{k_l=1}^{S_l} P^*_{l,k_l} \right) \right] \\ & - \sum_{\substack{j=1 \\ j\neq i}}^{n} \left[B_{0j} \left(\sum_{k_j=1}^{S_j} P^*_{j,k_j} \right) \right] - B_{00} \end{aligned} \tag{7.39}$$

It has to be noted that even if P^*_i satisfies the preceding design constraints[15], this P^*_i needs to be optimally defined because different schedules with many fuel mixtures can be selected for all the S_i units operated in the ith power station. In this case, the inner objective function could be used as follows:

$$P^*_i \;\rightarrow\; \text{OBJ2} = \min \sum_{k_i=1}^{S_i} C_{i,k_i} \left(P_{i,k_i} \right) \tag{7.40}$$

Note that the star superscripts are removed here because the best fuel mixture is considered during solving the optimal scheduling of the ith power station. It is similar to the concept used in (7.17) for the classical ELD problem.

Slack Generator

This approach is relatively easier than the previous one because the strategies used in the classical ELD method can be implemented here after applying some essential modifications. If P_L is neglected, then (3.16) becomes:

$$P^*_{i,\check{k}} = P_D + \sum_{j=1}^{n} P^*_{j,\text{aux}} - \sum_{j=1}^{n} \sum_{\substack{k_j=1 \\ (k_j\neq \check{k} \text{ if } j=i)}}^{S_j} P^*_{j,k_j} \tag{7.41}$$

where $P^*_{i,\check{k}}$ means the production of any unit selected among S_i units available in the ith power station.

[15] i.e., with/without considering P_L.

The fuel mixture assigned to the $\check{k}$th unit is denoted by $\dot{V}_{i,\check{k}}$; refer to (7.29). In this case, it is preferable to select the largest capacity unit of the ith power station as a slack generator to avoid facing infeasible solutions or adding external sub-algorithm(s) to deal with this condition [30].

Now, if P_L is considered, then Kron's loss formula can be used to modify (7.41) to the following expression:

$$\begin{aligned} P^*_{i,\check{k}} - \sum_{j=1}^{n}\left[P^*_{i,\check{k}}B_{ij}\left(\sum_{k_j=1}^{S_j}P^*_{j,k_j}\right)\right] - \sum_{j=1}^{n}\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)B_{ji}P^*_{i,\check{k}} - B_{0i}P^*_{i,\check{k}} = P_D \\ + \sum_{j=1}^{n}P^*_{j,\text{aux}} + \sum_{j=1}^{n}\sum_{l=1}^{n}\left[\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)B_{jl}\left(\sum_{\substack{k_l=1\\ k_l\neq\check{k}\text{ for }l=i}}^{S_l}P^*_{l,k_l}\right)\right] \\ + \sum_{j=1}^{n}\left[B_{0j}\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)\right] + B_{00} - \sum_{j=1}^{n}\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j} \end{aligned} \tag{7.42}$$

Similar to (7.35), a quadratic equation can be obtained by extracting $P^*_{i,\check{k}}$ from the summations of (7.42) as follows:

$$\bar{a}P^{*2}_{i,\check{k}} + \bar{b}P^*_{i,\check{k}} + \bar{c} = 0 \tag{7.43}$$

The analytical solution of (7.43) is:

$$P^*_{i,\check{k}} = \frac{-\bar{b} \pm \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}}}{2\bar{a}}\ , \quad \bar{b}^2 - 4\bar{a}\bar{c} \leq 0 \tag{7.44}$$

where $\{\bar{a}, \bar{b}, \bar{c}\}$ can be respectively determined as follows:

$$\bar{a} = -B_{ii} \tag{7.45}$$

$$\bar{b} = 1 - \sum_{j=1}^{n}\left[B_{ij}\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)\right] - \sum_{j=1}^{n}\left[\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)B_{ji}\right] - B_{0i} \tag{7.46}$$

$$\bar{c} = \sum_{j=1}^{n}\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j} - P_D - \sum_{j=1}^{n}P^*_{j,\text{aux}} - \sum_{j=1}^{n}\sum_{l=1}^{n}\left[\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j}P^*_{j,k_j}\right)\right.$$

$$B_{jl}\left(\sum_{\substack{k_l=1\\ k_l\neq\check{k}\text{ for }l=i}}^{S_l} P^{*}_{l,k_l}\right)\Bigg] - \sum_{j=1}^{n}\left[B_{0j}\left(\sum_{\substack{k_j=1\\ k_j\neq\check{k}\text{ for }j=i}}^{S_j} P^{*}_{j,k_j}\right)\right] - B_{00} \tag{7.47}$$

7.3.1.2 Generator Ramp Rate Limit Constraint

Considering the classical generator ramp rate limits given in (3.30)-(3.31). To make these inequality constraints valid in real-world applications, they should be modified as follows:

$$P^{*,\text{now}}_{i,k_i} - P^{*,\text{new}}_{i,k_i} \leqslant R^{*,\text{down}}_{i,k_i} \tag{7.48}$$

$$P^{*,\text{new}}_{i,k_i} - P^{*,\text{now}}_{i,k_i} \leqslant R^{*,\text{up}}_{i,k_i} \tag{7.49}$$

where $P^{*,\text{now}}_{i,k_i}$ and $P^{*,\text{new}}_{i,k_i}$ are respectively the existing and new power outputs of the k_ith unit operated in the ith power station using the optimal fuel mixture $\dot{V}_{i,k_i}$. The terms $R^{*,\text{down}}_{i,k_i}$ and $R^{*,\text{up}}_{i,k_i}$ are respectively the downward and upward ramp rate limits of the k_ith unit. These two equations can be embedded within (7.30) as follows:

$$\max\left(P^{*,\min}_{i,k_i}, P^{*,\text{now}}_{i,k_i} - R^{*,\text{down}}_{i,k_i}\right) \leqslant P^{*,\text{new}}_{i,k_i} \leqslant \min\left(P^{*,\max}_{i,k_i}, P^{*,\text{now}}_{i,k_i} + R^{*,\text{up}}_{i,k_i}\right) \tag{7.50}$$

7.3.1.3 Prohibited Operating Zone Constraint

Similarly, the classical prohibited operating zone constraint given in (3.33) cannot be applied in real ELD optimizers without considering the common busbar configuration of power stations. Therefore, the realized inequality constraint of this phenomenon can be mathematically expressed as follows:

$$\begin{aligned} P^{*,\min}_{i,k_i} &\leqslant P^{*}_{i,k_i} \leqslant P^{*,L}_{i,k_i,j} \\ P^{*,U}_{i,k_i,j} &\leqslant P^{*}_{i,k_i} \leqslant P^{*,L}_{i,k_i,j+1} \\ P^{*,U}_{i,k_i,\varkappa_{k_i}} &\leqslant P^{*}_{i,k_i} \leqslant P^{*,\max}_{i,k_i} \end{aligned} \tag{7.51}$$

where $P^{*,L}_{i,k_i,j}$ and $P^{*,U}_{i,k_i,j}$ are respectively the lower and upper limits of the jth prohibited operating zone on the fuel-cost curve of the k_ith unit of the ith power station when it operates at the optimal fuel mixture. The term $\varkappa_{k_i}$ stands for the total number of prohibited operating zones associated with the k_ith unit.

7.3.1.4 Emission Rates Constraint and Penalization Criterion

Based on rules followed in each jurisdiction, there is an environmental regulation that requires power stations to not exceed the maximum allowable limits of the emission rates (such as NOx, SOx, CO/CO_2, soot and UHC) [30,302]. The classical inequality constraint formulation of these emission rates, given in [4,243], can be used here, but after applying a minor modification as follows:

$$\sum_{k_i=1}^{S_i} \mathrm{GER}_{k_i,j}\left(P^*_{i,k_i}\right) \leqslant \mathrm{MAL}_{i,j}\ , \qquad \begin{matrix} i=1,2,\cdots,n \\ j=1,2,\cdots,m \\ k_i=1,2,\cdots,S_i \end{matrix} \tag{7.52}$$

where $\mathrm{GER}_{k_i,j}$ and $\mathrm{MAL}_{i,j}$ stand for the rate of the jth emission produced by the k_ith unit and the maximum allowable limit of that emission rate allowed to the ith power station, respectively. The notations m and n denote the total number of emissions and the total number of energized busbars, respectively.

Also, most of the ELD optimizers presented in the literature are designed based on one strict criterion that the solution is considered feasible only if there is no any violated constraint. However, in real-world applications, the ith power station could exceed $\mathrm{MAL}_{i,j}$ by paying some extra tax. Thus, the rejected infeasible solutions may contain very good results with an ignorable or very small violation of the emission rates constraint. Based on this, instead of using death penalty functions to kill these good individuals, it is better to use less aggressive penalty functions such as static exterior penalty functions (extinctive, binary, and distance-based) or other smarter penalty functions (exponential, adaptive, self-adaptive, superiority of feasible points, eclectic evolutionary algorithm, etc); refer to Figure 2.6. That is, to realize any ELD optimizer, this constraint should be carefully addressed and distinguished from other constraints. Therefore, the objective function given in (7.26) could be extended to be as follows:

$$\mathrm{OBJ} = \min \sum_{i=1}^{n} \left[\sum_{k_i=1}^{S_i} C^*_{i,k_i}\left(P^*_{i,k_i}\right) + \sum_{j=1}^{m} \Phi_{i,j} \right] \tag{7.53}$$

where $\Phi_{i,j}$ is the jth emission rate penalty function applied to the ith power station.

To determine the value of $\Phi_{i,j}$, the following equation can be implemented:

$$\Phi_{i,j} = \begin{cases} 0, & \text{if } \sum_{k_i=1}^{S_i} \mathrm{GER}_{k_i,j}\left(P^*_{i,k_i}\right) \leqslant \mathrm{MAL}_{i,j} \\ \left[\sum_{k_i=1}^{S_i} \mathrm{GER}_{k_i,j}\left(P^*_{i,k_i}\right) - \mathrm{MAL}_{i,j}\right] \times \sigma_{i,j}, & \text{if } \sum_{k_i=1}^{S_i} \mathrm{GER}_{k_i,j}\left(P^*_{i,k_i}\right) > \mathrm{MAL}_{i,j} \end{cases} \tag{7.54}$$

where $\sigma_{i,j}$ is the penalized cost coefficient (in \$/ppm) applied to the exceeded amount of the jth emission rate in the ith power station.

Although $\Phi_{i,j}$ is expressed in an additive form, the multiplicative form can also be used here with changing the first case of (7.54) from 0 to 1 if the maximum allowable limit of the jth emission rate is satisfied in the ith power station.

7.3.1.5 Planned Maintenance Constraint

Each power station has four principle departments; operation, maintenance, planning, and administration. The first one is responsible to operate machines and associated auxiliary equipment within some specific limits instructed by the head or the manager of the administrative department. If a fault or damage exists in an equipment, then the operation department issues some documents, like **Limit of Access** (**LOA**) and **Permit to Work** (**PTW**), to the respective section of the maintenance department (the laboratory, mechanical, electrical, C&I, or IT) to investigate or fix that problem and restoring it to the normal operating condition. This type of maintenance is called **reactive** or **run-to-failure** maintenance, which is **unplanned** maintenance and the jobs listed under this category are almost emergent and breakdown jobs. On the opposite side, there are four planned maintenance jobs; **predictive**, **preventive**, **proactive** (**improvement**), and **corrective** maintenance jobs [112]. The jobs listed under these four types of maintenance are issued from the operation to the maintenance section(s) after being initiated by the planning department and proved by the administrative department. Suppose that there are two units, one has been just returned from a major overhaul, while the other exceeds its allowable effective operating hours (EOH). Of course, the first unit is preferred if both have the same characteristics and supplying the same busbar. Even if they are from different manufacturers and supplying different busbars, a penalty function should be raised here to give weight when deciding to run this unhealthy unit. Therefore, the objective function given in (7.53) is further extended to:

$$\text{OBJ} = \min \sum_{i=1}^{n} \sum_{k_i=1}^{S_i} \left[C^*_{i,k_i}\left(P^*_{i,k_i}\right) \cdot \sum_{h=1}^{\mathfrak{w}} D^*_{i,k_i,h} + \sum_{j=1}^{m} \Phi_{i,j} \right] \tag{7.55}$$

where $D^*_{i,k_i,h}$ is a scaling factor. It becomes bigger than 1 if the hth planned maintenance on the k_ith unit (using an optimal fuel mixture $\rightarrow$ "$*$") of the ith power station

passes the due date. The term $D^*_{i,k_i,h}$ can be expressed in many ways, such as:

$$D^*_{i,k_i,h} = \begin{cases} 1, & \text{if the } k_i\text{th unit satisfies all the } h\text{th planned maintenance dates} \\ \xi_{i,k_i,h}, & \text{otherwise} \end{cases} \tag{7.56}$$

where $\xi_{i,k_i,h}$ is the severity level of the hth planned maintenance on the k_ith unit and its impact on the overall system when the maintenance due date is exceeded. The scaling value should be high with very sever maintenance jobs (such as generator winding impedance, excitation system and **static frequency converter** (**SFC**), significant vibration on the prime-mover, **electro-hydraulic governor** (**EHG**), **lube oil system**, etc), and vice versa for other minor maintenance jobs (such as periodic panel brush-cleaning, taking backup data from archiving servers, editing graphics of human-machine interface (HMI) screens, periodic oil quality analysis of power transformers, etc).

7.3.1.6 Degraded Efficiency Phenomenon

This is one of the unrevealed constraints and it still does not have any attention in the literature. All the preceding information is based on a fact that the power output of the k_ith unit of the ith power station at the optimal fuel mixture $\dot{V}^*_{i,k_i}$ has a fixed fuel-power curve. In fact, based on our experience with GTs and STs, the power output deviates with the time and it can be predicted and measured based on the recorded past and current status of the k_ith unit. For example, if some burners of GT are not working or if there are some disturbances in the **air mixing valve**, then the machine efficiency will decrease. A similar thing can be applied here when the k_ith machine is just returned back from its major overhaul or when it is operating for a long time without any proper maintenance. That is, as EOH increases or as the machine status degrades due to some worn-out components, the machine efficiency decreases. This means that the k_ith unit produces less power with the same fuel rate consumption. Therefore, this phenomenon directly affects the fuel-power curve of the k_ith unit as illustrated in Figure 7.13. Thus, the variable bounds given in (7.30) should be modified as follows:

$$\tilde{P}^{*,\min}_{i,k_i} \leqslant \tilde{P}^*_{i,k_i} \leqslant \tilde{P}^{*,\max}_{i,k_i} \tag{7.57}$$

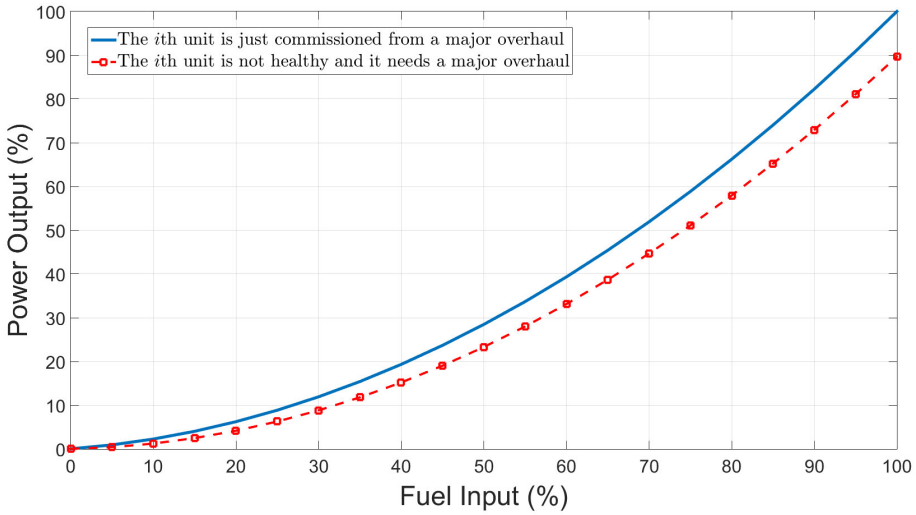


Figure 7.13: An Illustrated Degraded Efficiency Phenomenon Due to the Total EOH Accumulated for the k_ith Unit

where $\tilde{P}^{*,\min}_{i,k_i}$ and $\tilde{P}^{*,\max}_{i,k_i}$ are respectively the lower and upper limits that the k_ith unit can currently reach based on the total EOH accumulated for that unit.

Therefore, the final expression of the objective function can be modeled by modifying (7.55) to:

$$\text{OBJ} = \min \sum_{i=1}^{n} \sum_{k_i=1}^{S_i} \left[U^*_{i,k_i,\text{EOH}} \left(\tilde{P}^*_{i,k_i,\text{EOH}} \right) \cdot C^*_{i,k_i} \left(\tilde{P}^*_{i,k_i} \right) \cdot \sum_{h=1}^{\mathfrak{w}} D^*_{i,k_i,h} + \sum_{j=1}^{m} \Phi_{i,j} \right] \quad (7.58)$$

where the term $U^*_{i,k_i,\text{EOH}}$ represents the factor responsible to simulate the degradation rate of the k_ith unit of the ith power station based on the EOH accumulated for that unit. This term could be just a fixed coefficient for a linear degrading rate, or it could be a function of $\tilde{P}^*_{i,k_i,\text{EOH}}$ for a nonlinear degrading approach.

7.3.2 Practical Approach to Find Optimal Fuel Mixtures

From the last subsection, it is clear that finding the optimal fuel mixture for each unit needs some helps from OEMs to model the burning quality of each fuel and the behavior of the **combustion chamber** at different chemical compositions. Unfortunately, this is a very hard requirement and somewhat impractical. Also, it is subject to many dynamic changes, such as pressure, temperature, air/fuel ratio and flow-rate, condensate, number of working burners, and present load. Thus, a more practical alternative is required, which is the goal of this subsection.

7.3.2.1 New Multi-Fuel Cost Model

By referring to the classical single-fuel-based ELD cost functions, they are actually *one*-dimensional polynomial equations. Thus, the whole process can be simplified by finding a direct relation between "the cost spent on" and "the power generated from" the ith unit. This can be mathematically described as follows:

- **Single Fuel** $\rightarrow$ $M = 1$:
  If $M = 1$, then $r_i = 1$ and $\dot{V}_i^M = \dot{v}_i$. The heat-rate consumed by the ith unit is proportional to the fuel flow-rate $\dot{v}_i$. Thus, the active power P_i and the fuel cost C_i are functions of $\dot{v}_i$. Because both P_i and C_i are *one*-dimensional functions, so a direct relationship between them can be established by the mean of regression analysis. These steps can be mathematically explained as follows:

  - $P_i = f_i(\dot{v}_i) \quad \rightarrow \quad \text{dimension} = 1$
  - $C_i = u_i(\dot{v}_i) \quad \rightarrow \quad \text{dimension} = 1$
  - $C_i = g_i(P_i) \quad \rightarrow \quad \text{dimension} = 1$

  Thus, the plot of "C_i (\$/hr) vs. P_i (MW)" is a mirror image of the plot of "P_i (MW) vs $\dot{v}_i$ (Btu/hr)". That is, no need to know the exact $\dot{v}_i$ to calculate C_i. The price can be indirectly calculated if P_i is known [11, 326]. This process can be graphically described in Figure 7.14.

- **Multiple Fuels** $\rightarrow$ $M > 1$:
  Once $M > 1$, the traditional cost-function is no longer valid. This hidden problem comes from the preceding regression relationships. To clarify it more, P_i and C_i are functions of multiple fuels while the classical model forces C_i to be a function of one variable:

  - $P_i = f_i(\dot{v}_{i,1}, \dot{v}_{i,2}, \cdots, \dot{v}_{i,W}, \cdots, \dot{v}_{i,M}) \quad \rightarrow \quad \text{dimension} = M$
  - $C_i = u_i(\dot{v}_{i,1}, \dot{v}_{i,2}, \cdots, \dot{v}_{i,W}, \cdots, \dot{v}_{i,M}) \quad \rightarrow \quad \text{dimension} = M$
  - $C_i = g_i(P_i) \quad \rightarrow \quad \text{dimension} = 1$

  This process is depicted in Figure 7.15 for two fuel gases where the 3D surface plot is not shown in the box because its shape depends on the physical properties and the chemical compositions of the two fuels.

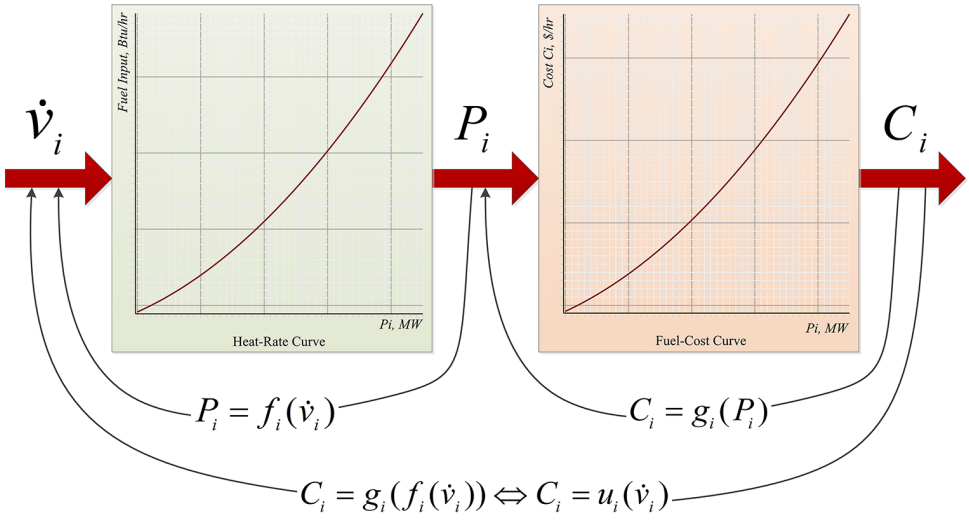


Figure 7.14: The Real Relation Between $\dot{V}_i$ and C_i $(J = 1)$

As can be clearly seen, the classical multi-fuel ELD model is not valid if $M > 1$ because P_i is regressed based on M variables while C_i is regressed based on only one variable. Thus, by varying r_i, the same power could be generated at different prices. One of the possible ways to solve this dilemma, directly without referring to OEMs, is by decomposing the fuel-cost function into two separate functions:

$$C_i = u_i\left(\dot{V}_i^M\right) \tag{7.59}$$

$$P_i = f_i\left(\dot{V}_i^M\right) \tag{7.60}$$

For C_i, the function u_i can be easily and directly expressed as follows:

$$C_i\left(\dot{V}_i^M\right) = \xi_1\dot{v}_{i,1} + \xi_2\dot{v}_{i,2} + \cdots + \xi_M\dot{v}_{i,M} \tag{7.61}$$

where $\{\xi_1, \xi_2, \cdots, \xi_M\}$ are the fuel price rates in \$/Btu or \$/Nm^3, which are fixed for all the units.

In our case study, $M = 2$. Thus, (7.61) becomes:

$$C_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \xi_1\dot{v}_{i,1} + \xi_2\dot{v}_{i,2} \tag{7.62}$$

where $\dot{v}_{i,1}$ and $\dot{v}_{i,2}$ are respectively the first and second gas flow-rates in Btu/hr.

Similar to (7.10), the last equation can be further modified to:

$$C_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \phi + \varphi\left[\xi_1\dot{v}_{i,1} + \xi_2\dot{v}_{i,2}\right] \tag{7.63}$$

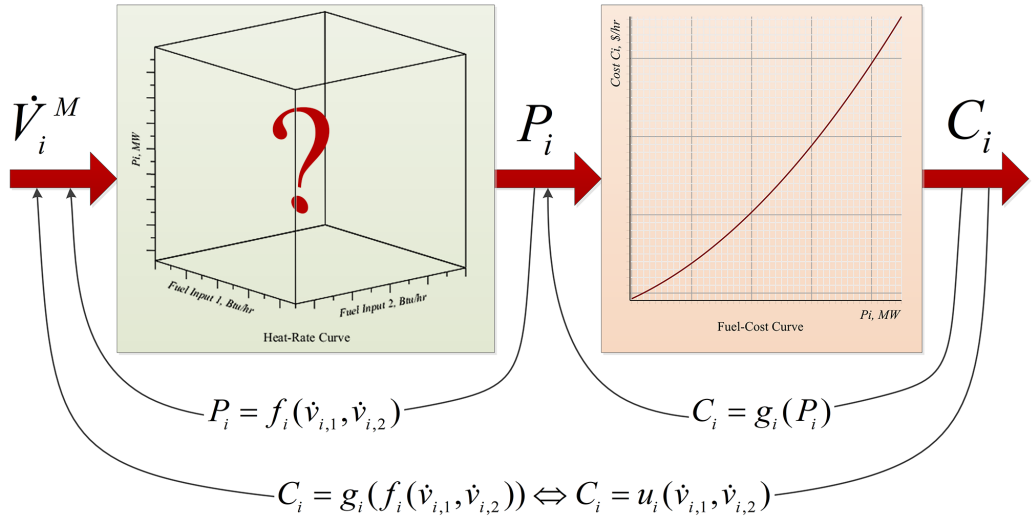


Figure 7.15: The Real Relation Between $\dot{V}_i$ and C_i $(J = 2)$

where ϕ is the initial cost in \$/hr and φ is the operational pricing gain "unitless".

Or, similar to (7.11) where the last formula can be even expanded to be in a quadratic form of $\dot{v}_{i,1}$ and $\dot{v}_{i,2}$ as follows:

$$C_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \phi + \varphi\left[\xi_1 \dot{v}_{i,1} + \xi_2 \dot{v}_{i,2}\right] + \psi\left[\xi_1 \dot{v}_{i,1} + \xi_2 \dot{v}_{i,2}\right]^2 \tag{7.64}$$

where ψ is the second-order pricing coefficient in hr/\$.

For P_i, many possible regression models can be used. In this dissertation, the following regression models are investigated:

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,1}^2 \tag{7.65}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,1}^2 + \zeta_{i,3}\dot{v}_{i,1}^3 \tag{7.66}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,2} + \zeta_{i,2}\dot{v}_{i,2}^2 \tag{7.67}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,2} + \zeta_{i,2}\dot{v}_{i,2}^2 + \zeta_{i,3}\dot{v}_{i,2}^3 \tag{7.68}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\left(\dot{v}_{i,1} + \dot{v}_{i,2}\right) + \zeta_{i,2}\left(\dot{v}_{i,1} + \dot{v}_{i,2}\right)^2 \tag{7.69}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\left(\dot{v}_{i,1} + \dot{v}_{i,2}\right) + \zeta_{i,2}\left(\dot{v}_{i,1} + \dot{v}_{i,2}\right)^2 + \zeta_{i,3}\left(\dot{v}_{i,1} + \dot{v}_{i,2}\right)^3 \tag{7.70}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} \tag{7.71}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} + \zeta_{i,3}\dot{v}_{i,1}\dot{v}_{i,2} \tag{7.72}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} + \zeta_{i,3}\dot{v}_{i,1}^2 + \zeta_{i,4}\dot{v}_{i,2}^2 \tag{7.73}$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} + \zeta_{i,3}\dot{v}_{i,1}^2 + \zeta_{i,4}\dot{v}_{i,2}^2 + \zeta_{i,5}\dot{v}_{i,1}\dot{v}_{i,2} \tag{7.74}$$

Table 7.3: Technical Specifications of the Real Multiple-Fuel-Based Gas Turbine Used in This Experiment

Manufacturer:	SIEMENS
Model:	Kraftwerk Union - KWU/V723/MeiBner
Base-Load:	43-45 MW
Fuel Type:	Natural Gas or Diesel

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} + \zeta_{i,3}\dot{v}_{i,1}^2 + \zeta_{i,4}\dot{v}_{i,2}^2 + \zeta_{i,5}\dot{v}_{i,1}^3 + \zeta_{i,6}\dot{v}_{i,2}^3 \quad (7.75)$$

$$P_i(\dot{v}_{i,1}, \dot{v}_{i,2}) = \zeta_{i,0} + \zeta_{i,1}\dot{v}_{i,1} + \zeta_{i,2}\dot{v}_{i,2} + \zeta_{i,3}\dot{v}_{i,1}^2 + \zeta_{i,4}\dot{v}_{i,2}^2 + \zeta_{i,5}\dot{v}_{i,1}^3 + \zeta_{i,6}\dot{v}_{i,2}^3 + \zeta_{i,7}\dot{v}_{i,1}\dot{v}_{i,2} \quad (7.76)$$

where $\{\zeta_{i,0}, \zeta_{i,1}, \cdots, \zeta_{i,7}\}$ are the polynomial regression coefficients of the ith unit.

Among these 12 regression-based models, the best model is the one that can compromise between different performance criteria, such as the dimension, degrees of freedom (DF), explanation levels (R^2 and R^2_{adj}), statistical F- and p-values, **variance inflation factors** (**VIF**), total number of **unusual observations**, etc.

7.3.2.2 Regression Experiments and Discussion

To find the best regression model and its zeta-coefficients, large data has been collected from the base-load of a real GT. The specifications of this turbo-generator shown in Figure 7.16 are tabulated in Table 7.3. The fuel mode of this GT was switched to the "Natural Gas" mode. Table 7.4 shows some specifications of the two fuel gases used with that GT. The measurements were recorded for 166 days and cover the first three months (January, February, and March) of 2012, 2013, and 2014. These measurements are shown in Figure 7.17. Although the top subplot is shown in MW, the regression analysis is carried out using kW to show more decimal places for higher-order polynomial equations. From Figure 7.17, it is clear that the relation between P_i and r_i is nonlinear. This means that the third scenario of r_i (i.e., variable ratio $\rightarrow$ $r_i \in [0, 1]$) is the case faced here with this GT, which is shown in Figure 7.18. These two facts prove our earlier claims, and thus it emphasizes the invalidity of (7.19).

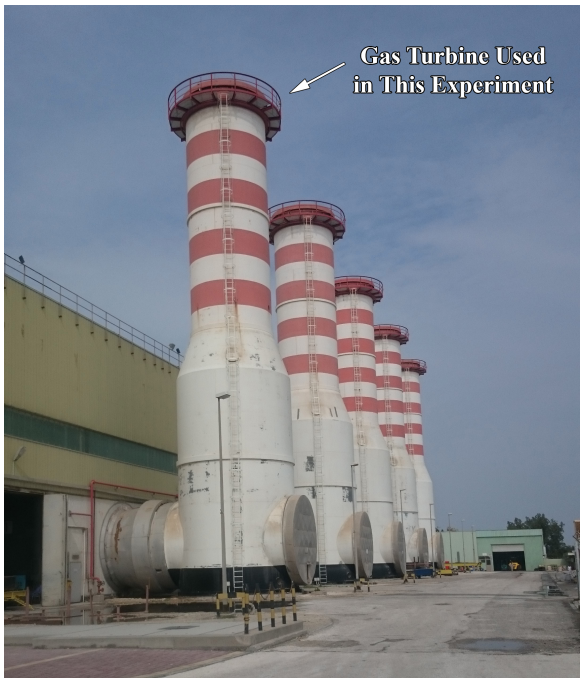


Figure 7.16: Real Photograph of the Gas Turbine Used in This Experiment

Table 7.4: Technical Specifications of the Two Fuel Gases

Fuel Type:	Residual Gas	Khuff Gas
Pressure[a]**:**	$\mathcal{P}_G \approx 18$ bar	$\mathcal{P}_G \approx 70$ bar
Reduction Plant:	No	Yes
Price:	more	less
Properties:	very clean	contains sludge

[a]The final pressure limits are adjusted via two gas stations.

Table 7.5 shows the coefficient values of the models given in (7.65)-(7.76) after being regressed by **MINITAB** using a **significance level** of 5%. From Table 7.6, it can be clearly seen that the explanation levels of (7.67) and (7.68) are less than 4% for R^2 and less than 2% for R^2_{adj}. A part of the reason is that the Khuff gas (i.e., $\dot{v}_2$) has a weak explanation of the variability, which is depicted in the second subplot of Figure 7.17. From the preceding subplot, it can be observed that the contribution of the Khuff gas is distinctly lower than that of the Residual gas (i.e., $\dot{v}_1$), which is clearly seen in Figure 7.18. However, even if someone wants to use only $\dot{v}_1$ to explain the variability of P, as a hope to stick with the classical fuel-cost functions given in

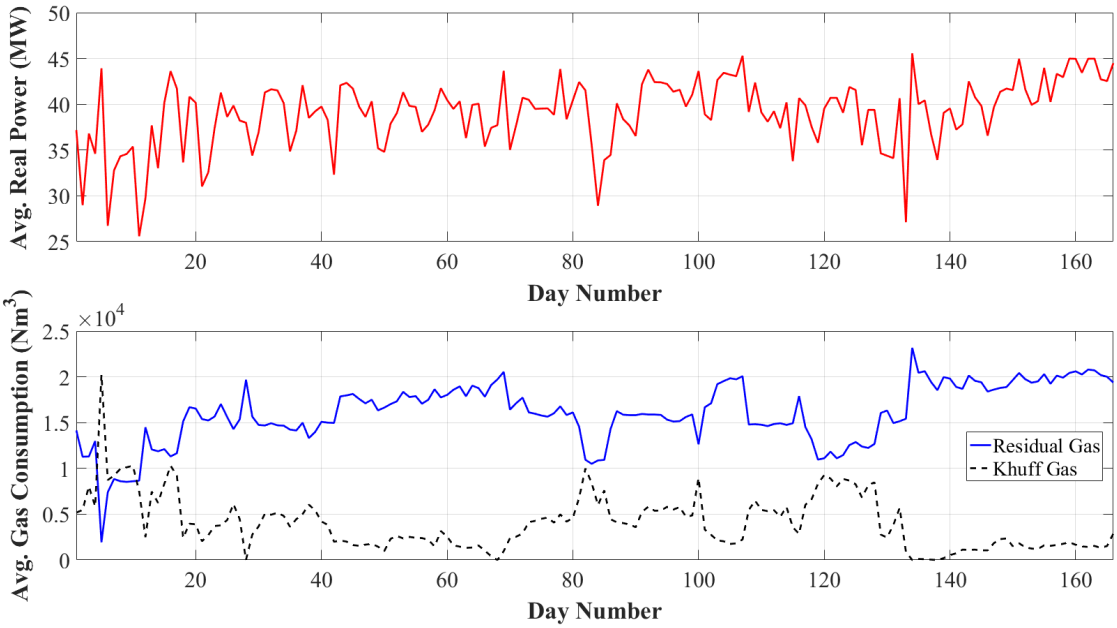


Figure 7.17: Average Daily Readings of GT: (a) Real Power Generation in MW. (b) Residual and Khuff Gas Consumption in Nm^3

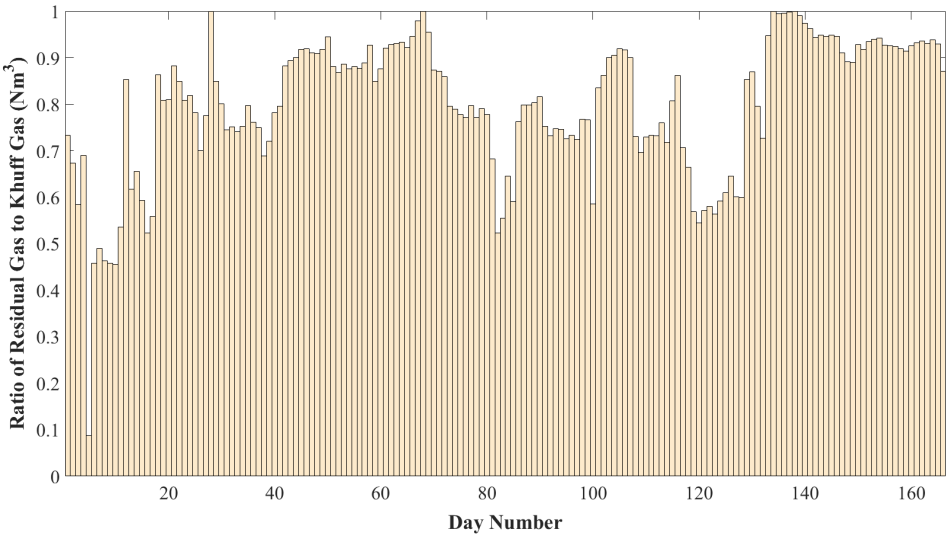


Figure 7.18: Variable Ratio Between the Two Fuel Gases

Table 7.5: Regression Model Coefficients of the Gas Turbine

Coefficient	Regression Model No.											
No.	(7.65)	(7.66)	(7.67)	(7.68)	(7.69)	(7.70)	(7.71)	(7.72)	(7.73)	(7.74)	(7.75)	(7.76)
ζ_0	39358	42325	40121	39391	-64943	-299631	-14475	-14303	-23606	-49017	-22791	-57737
ζ_1	-0.733	-1.61	-0.438	0.148	7.82	44.3	2.6818	2.6605	3.888	6.41	3.75	7.68
ζ_2	0.000043	0.000117	0.000027	-0.000068	-0.000131	-0.00201	2.6366	2.5237	2.478	5.40	2.713	6.43
ζ_3	—	≈ 0	—	≈ 0	—	≈ 0	—	0.000011	-0.000038	-0.000100	-0.000033	-0.000169
ζ_4	—	—	—	—	—	—	—	—	0.000023	-0.000057	-0.000014	-0.000135
ζ_5	—	—	—	—	—	—	—	—	—	-0.000147	≈ 0	≈ 0
ζ_6	—	—	—	—	—	—	—	—	—	—	≈ 0	≈ 0
ζ_7	—	—	—	—	—	—	—	—	—	—	—	-0.000182

Chapter 3, it will be realized that the explanation level of this major fuel gas will not help him/her that much. This can be proved by returning back to Table 7.6 where the explanation levels of (7.65) and (7.66) are also very low. This "weak explanation level" problem cannot be solved even if the **polynomial order** is increased from 2 to 3; as seen in (7.66) and (7.68). On the opposite side, once $\dot{v}_1$ and $\dot{v}_2$ are both considered in the regression model, the explanation level markedly jumps to near 90%. This can be obviously observed in (7.69)-(7.74). Choosing between these highly significant regression models is like compromising between different performance criteria. Let's focus on the variance inflation factor (VIF) as the first step to eliminate all unwanted regression models. Thus, (7.74)-(7.76) will lose the competition. Based on this, the recommended models, to explain the real power (in kW) generated by multi-fuel thermal units, are (7.71)-(7.73). The regression results of these models are:

$$P = -14475 + 2.6818\dot{v}_1 + 2.6366\dot{v}_2 \tag{7.77}$$

$$P = -14303 + 2.6605\dot{v}_1 + 2.5237\dot{v}_2 + 0.000011\dot{v}_1\dot{v}_2 \tag{7.78}$$

$$P = -23606 + 3.888\dot{v}_1 + 2.478\dot{v}_2 - 0.000038\dot{v}_1^2 + 0.000023\dot{v}_2^2 \tag{7.79}$$

The performances of (7.71) and (7.72) are almost same. Although the interaction between $\dot{v}_1$ and $\dot{v}_2$ in (7.72) can slightly help in reducing the variance s as well as improving R^2 and R^2_{adj} with less number of unusual observations, its F-Value is less than that of (7.71). A similar thing can be noticed with (7.73). Add to that, (7.73) has high VIF values, which is an indication of a high correlation between the predictors. Thus, to be kept away from the condition of difficult and unstable interpretation, (7.71) is preferred. However, before selecting (7.71) as the best regression model among these 12 models, many other alternatives between (7.71) and (7.74) could

Table 7.6: Some Performance Criteria of the Regression Models

Performance	Regression Model No.											
Criteria[a]	(7.65)	(7.66)	(7.67)	(7.68)	(7.69)	(7.70)	(7.71)	(7.72)	(7.73)	(7.74)	(7.75)	(7.76)
DF	2	3	2	3	2	3	2	3	4	5	6	7
F-Value	26.68	17.95	1.97	1.80	781.34	527.74	724.79	492.42	397.55	322.40	265.00	232.07
p-value	0.000	0.000	0.143	0.148	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Variance "s"	3283.21	3286.97	3737.60	3732.35	1162.51	1155.98	1202.58	1192.77	1154.00	1147.22	1154.73	1143.83
$\mathbf{R^2}$	24.66%	24.95%	2.36%	3.23%	90.55%	90.72%	89.89%	90.12%	90.81%	90.97%	90.91%	91.14%
$\mathbf{R^2_{adj}}$	23.73%	23.56%	1.16%	1.44%	90.44%	90.55%	89.77%	89.93%	90.58%	90.69%	90.57%	90.74%
Minimum VIF	32.14	236.47	5.03	29.19	418.00	101611.24	6.16	3.28	16.60	459.27	53.13	174.51
Maximum VIF	32.14	1321.29	5.03	122.64	418.00	415051.36	6.16	9.62	137.25	3204.62	11723.17	17215.00
Large Residual	10	11	7	7	15	15	16	16	15	14	15	13
Unusual Observations	3	3	1	3	8	6	4	2	4	5	4	6

[a] $\mathbf{R^2}$: the explanation level of the response variable. $\mathbf{R^2_{adj}}$: the adjusted version that takes into account the number of predictors used in the model.

provide better regression results. Using a significance level of 5% with the "**backward elimination**", "**forward selection**" and "**stepwise**" algorithms, the best model is:

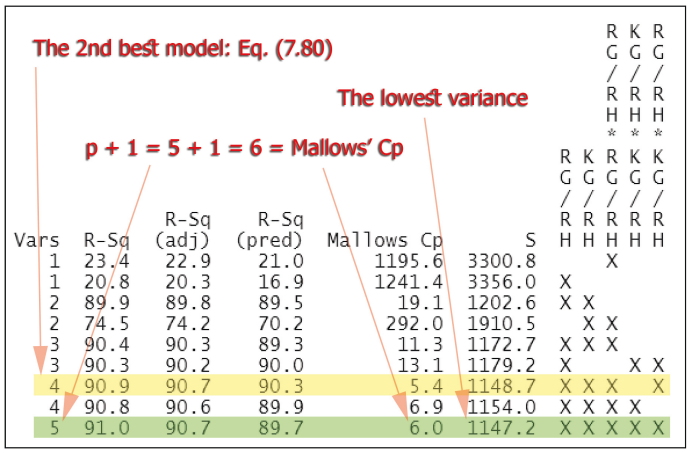
$$P = -32033 + 4.744\dot{v}_1 + 3.381\dot{v}_2 - 0.000059\dot{v}_1^2 - 0.000046\dot{v}_1\dot{v}_2 \tag{7.80}$$

where the degrees of freedom DF = 4, F-Value = 401.61, p-value = 0.000, s = 1148.69, $R^2 = 90.89\%$, and $R^2_{\text{adj}} = 90.66\%$.

If the built-in "**best subsets regression**" function is activated, then the result shown in Figure 7.19 can be attained. From that result, it can be observed that one of the best regression models is (7.80). Also, regardless of high VIF, (7.74) can be selected as a competitive model where the variance s and the explanation levels (i.e., R^2 and R^2_{adj}) are at the lowest and highest values (s = 1147.2, $R^2 = 90.97\%$, and $R^2_{\text{adj}} = 90.69\%$), respectively. The other evidence that leads MINITAB to select (7.74), as one of the best regression models, is the gap between (p = DF + 1) and the **Mallows' $\mathbf{C_p}$ factor**. This model has a zero difference between C_p and p where $p = C_p = 6$. Therefore, one of the possible regression models that can be used with multi-fuel thermal units is:

$$P = -49017 + 6.41\dot{v}_1 + 5.40\dot{v}_2 - 0.000100\dot{v}_1^2 - 0.000057\dot{v}_2^2 - 0.000147\dot{v}_1\dot{v}_2 \tag{7.81}$$

This model is more general because the model obtained by the three elimination/selection algorithms, i.e. (7.80), neglects the effect of the fifth term (i.e., $\zeta_4\dot{v}_2^2$) due to the majority of $\dot{v}_1$.



Vars	R-Sq	R-Sq (adj)	R-Sq (pred)	Mallows Cp	S	RGRH	KGRH	RGRH*	KGRH*	RGRH*
1	23.4	22.9	21.0	1195.6	3300.8			X		
1	20.8	20.3	16.9	1241.4	3356.0	X				
2	89.9	89.8	89.5	19.1	1202.6	X	X			
2	74.5	74.2	70.2	292.0	1910.5		X	X		
3	90.4	90.3	89.3	11.3	1172.7	X	X	X		
3	90.3	90.2	90.0	13.1	1179.2	X			X	X
4	90.9	90.7	90.3	5.4	1148.7	X	X	X		X
4	90.8	90.6	89.9	6.9	1154.0	X	X	X	X	
5	91.0	90.7	89.7	6.0	1147.2	X	X	X	X	X

Figure 7.19: Result Obtained by the MINITAB's Best Subsets Regression Function

How to Solve Multi-Fuel-Based ELD Problems

It has been seen that the fuel-cost functions of multi-fuel thermal generating units cannot be represented as functions of real-power anymore. Rather, they are functions of fuel flow-rates. Thus, there are two stages to optimize this realistic ELD problem. The first stage is to minimize C_i using any one of (7.62)-(7.64), while the other is to satisfy the equality constraint of P using any one of (7.71)-(7.74) with the lowest possible fuel price rates.

7.4 Incorporating Other Abandoned Facts and Phenomena

The preceding sections summarize many facts and phenomena faced in real electric power systems. However, the realized ELD model needs to be further modified with some essential corrections. This section gives a quick overview of these new facts and phenomena and how to be incorporated into the proposed realized ELD model.

7.4.1 Equality Constraints of Power Stations

Suppose that there are Þ power stations, i.e. $\mathrm{PS}_x \in [\mathrm{PS}_1, \mathrm{PS}_2, \cdots, \mathrm{PS}_{\text{Þ}}]$, where each one is configured like the one shown in Figure 7.12. This common busbar configuration is mathematically explained by (7.25), which reveals a very important issue that is frequently faced in real power stations but completely ignored in all the studies presented in the literature. It says that, in addition to the power balance

equality constraint given in (3.16), there is at least one equality constraint that should be satisfied for each power station[16]. This new equality constraint can be called the "**station power balance equality constraint**", the "**shared power equality constraint**", or the "**common busbar equality constraint**". Thus, for Þ power stations with common busbars, there are at least Þ new equality constraints that must be added to any ELD optimization problem. If all the power stations are operated using the optimal fuel mixtures, then these Þ station power balance equality constraints can be mathematically expressed as follows:

$$P^*_{\mathrm{PS}_1} = P^*_{1,1} + P^*_{1,2} + \cdots + P^*_{1,j} + \cdots + P^*_{1,S_1} - P_{1,\mathrm{aux}} \tag{7.82}$$

$$P^*_{\mathrm{PS}_2} = P^*_{2,1} + P^*_{2,2} + \cdots + P^*_{2,j} + \cdots + P^*_{2,S_2} - P_{2,\mathrm{aux}} \tag{7.83}$$

$$\vdots$$

$$P^*_{\mathrm{PS}_x} = P^*_{i,1} + P^*_{i,2} + \cdots + P^*_{x,j} + \cdots + P^*_{i,S_x} - P_{x,\mathrm{aux}} \tag{7.84}$$

$$\vdots$$

$$P^*_{\mathrm{PS}_{\text{Þ}}} = P^*_{\text{Þ},1} + P^*_{\text{Þ},2} + \cdots + P^*_{\text{Þ},j} + \cdots + P^*_{\text{Þ},S_{\text{Þ}}} - P_{\text{Þ},\mathrm{aux}} \tag{7.85}$$

Based on this, if this real phenomenon is modeled, then all the existing optimization algorithms will fail. Previously, it has been seen how the **active power balance equality constraint** given in (3.16) makes lots of headache technical problems. Now, imagine that Þ new equality constraints are also added to the optimization model. Therefore, the current dispatchers will require some special sub-algorithms and a significant amount of CPU time to deal with this realized ELD model. This is one of the reasons why there is a gap between the mathematical-based ELD solvers and the actual optimal solutions that are supposed to be detected.

7.4.2 Steam Turbine Fuel-Cost Functions in CCPPs

In CCPPs, STs are operated by superheated steams generated by the heat exhausted from GTs[17]. This fatal fact has a fateful decision and a direct impact on the feasibility of solutions obtained from ELD, UC and OPF problems. It tells us that some STs, which have a massive power production, depend on their corresponding GTs. Thus,

[16]It depends on the number of common busbars used to deliver the power generated from each power station. Please, refer to Figure 7.10.

[17]This is done via some **heat recovery steam generators** (**HRSGs**) connected on the same header as illustrated in Figure 7.10.

for each CCPP block, as the power output of GTs increases the amount of exhausted heat goes up, so the power output of ST increases too; and vice versa when the power output of GTs decreases. Mathematically speaking, the power output of the jth ST is a function of the heat exhausted from all GTs connected in the same CCPP block. Thus, for Λ GTs, the relationship can be expressed as follows:

$$P_{\mathrm{ST}_j} = f\left(P_{\mathrm{GT}_1}, P_{\mathrm{GT}_2}, \cdots, P_{\mathrm{GT}_\Lambda}\right) \tag{7.86}$$

Thus, if the ith power station is operated by multiple GTs and one ST, then (7.25) should be modified to:

$$P_i = -P_{i,\mathrm{aux}} + \sum_{k_i=1}^{S_i-1} P_{i,k_i} + f\left(P_{i,1}, P_{i,2}, \cdots, P_{i,S_i-1}\right) \tag{7.87}$$

Using **multiple regression**, the power output from the S_ith ST could be approximated as follows[18]:

$$P_{i,S_i} = f\left(P_{i,1}, P_{i,2}, \cdots, P_{i,S_i-1}\right) = \eta_i + \sum_{k_i=1}^{S_i-1} \left(\mu_{i,k_i} P_{i,k_i} + \nu_{i,k_i} P_{i,k_i}^2\right) \tag{7.88}$$

where η_i is the intercept, and μ_{i,k_i} and ν_{i,k_i} are respectively the first- and second-order coefficients associated with the active power of the k_ith GT.

The same thing can be applied to calculate the operating cost of the S_ith ST. Thus, for the preceding ith power station, the total operating cost can be calculated as follows:

$$C_i = \sum_{k_i=1}^{S_i} C_{i,k_i} = \sum_{k_i=1}^{S_i-1} C_{i,k_i} + u\left(C_{i,1}, C_{i,2}, \cdots, C_{i,S_i-1}\right) \tag{7.89}$$

where the last term could be approximated as follows:

$$C_{i,S_i} = u\left(C_{i,1}, C_{i,2}, \cdots, C_{i,S_i-1}\right) = \varsigma_i + \sum_{k_i=1}^{S_i-1} \left(\rho_{i,k_i} C_{i,k_i} + \varphi_{i,k_i} C_{i,k_i}^2\right) \tag{7.90}$$

where ς_i is the intercept, and ρ_{i,k_i} and φ_{i,k_i} are respectively the first- and second-order coefficients associated with the operating cost of the k_ith GT.

[18] It has to be said that there are many other possible multiple regression models.

7.4.3 The Inaccuracy of Kron's Loss Formula

It has been seen that the active power loss P_L of the classical ELD models are calculated based on Kron's loss formula. The literature contains many other alternatives, such as George's formula, the Depezo loss formula, A-loss coefficients technique, sensitivity factors technique, loss formula as a function of generations and loads, loss formulas as functions of active and reactive power, network admittance matrix technique, etc [114, 210, 270]. All these techniques can estimate P_L under some circumstances, and thus each one of them has some inherent weaknesses. For instance, Kron's loss formula is not accurate because it is built based on many assumptions, such as constant bus voltages, constant bus angles, fixed load, and steady-state network topology [123, 210]. In Chapter 9/Subsection 9.1.4, the performance of Kron's loss formula is evaluated using a simple test system[19] subjected to different dynamic changes. The results show that the estimation error significantly increases if one of the branches is taken out of service or if one of the generator or load settings is far away from its initial value where the B-coefficients are calculated based on.

Since the CPU time is not an issue anymore, it is preferable to use NR or any other accurate PF solver to calculate P_L; and Q_L if it is needed. This can be done by applying (6.105) and (6.106) for both the active and reactive power losses. The other option is to involve AI, which is covered in Chapter 9.

7.4.4 Unaccounted Losses in Power Stations

By referring to all the preceding realized ELD models, the power losses that need to be subtracted from the total generated power are $[P_L, P_{1,\text{aux}}, P_{2,\text{aux}}, \cdots, P_{\text{Þ,aux}}]$. The same thing can be applied to the reactive power losses. Realistically speaking, these losses reflect just a small part of the real losses across the entire power system.

To explain these losses, first, let's look into the problem from the energy[20] side. To have electrical energy from thermal power stations, multiple energy conversions are needed. For CCPPs, the natural gas[21] is sent from its well to **gas stations**. That dirty fuel gas is cleaned through **scrubbers**. Also, by other **separation plants**, the

[19] It is the WSCC 9-bus test system. Please, refer to Figure 6.26.
[20] Power = Energy / Time.
[21] It can also be diesel fuel.

condensate and sludge can be filtered out. Further, part of the fuel gas could be burned and discharged to the atmosphere by a **flare stack** to prevent the process components from being pressurized. Moreover, a small part of the fuel gas is utilized in some **heating plants** to heat the gas before being sent to GTs. Add to that, a very small part of unburned fuel gas is exhausted out[22].

Once the fuel gas is burnt, a chemical to thermal energy conversion will take a place. This happens in the turbine stage of each GT. As a result, the prime-mover will start rotating, which is a thermal to mechanical energy conversion. Because the prime-mover is connected with a generator, so a mechanical to electrical energy will take a place. Also, the total heat exhausted from all the GTs, installed in the same block[23], is utilized to generate superheated steam via HRSGs, which is a **heat exchange process**. Then, the superheated steam is used to rotate the prime-mover of ST, which is a thermal to mechanical energy conversion. Again, that prime-mover will rotate a generator, which is a mechanical to electrical energy conversion. A part of electrical energy is used in the excitation system of each generator. After that, step-up transformers are used to increase the voltage and reduce the current so that the power losses in the network can be minimized. Also, **inter-bus transformers** (**IBTs**) are used to connect branches of different voltage levels. Add to that, a significant part of the total power is consumed by the station auxiliary plants[24]. Finally, a small part of electricity is used in the station buildings, air conditioning systems, **local control rooms** (**LCRs**), main control rooms (MCRs), lightings, etc.

As can be seen, many losses need to be considered in our realized ELD model. The operating cost is sensitive to many of these factors. The **Sankey diagram** shown in Figure 7.20 depicts most of these losses. Including them means a major improvement in the realized ELD model.

[22]The amount depends on many factors, such as the combustor type, its status, the fuel gas temperature, the fuel/air ratio, etc.

[23]Please, refer to Figure 7.10.

[24]Such as boiler **feed water pumps**, **air compressors**, gas stations - including **gas compressors** and **gas heaters** -, water and lube oil cooling systems, **utility transformers**, **air intake systems**, **reverse osmosis** (**RO**) plants, etc.

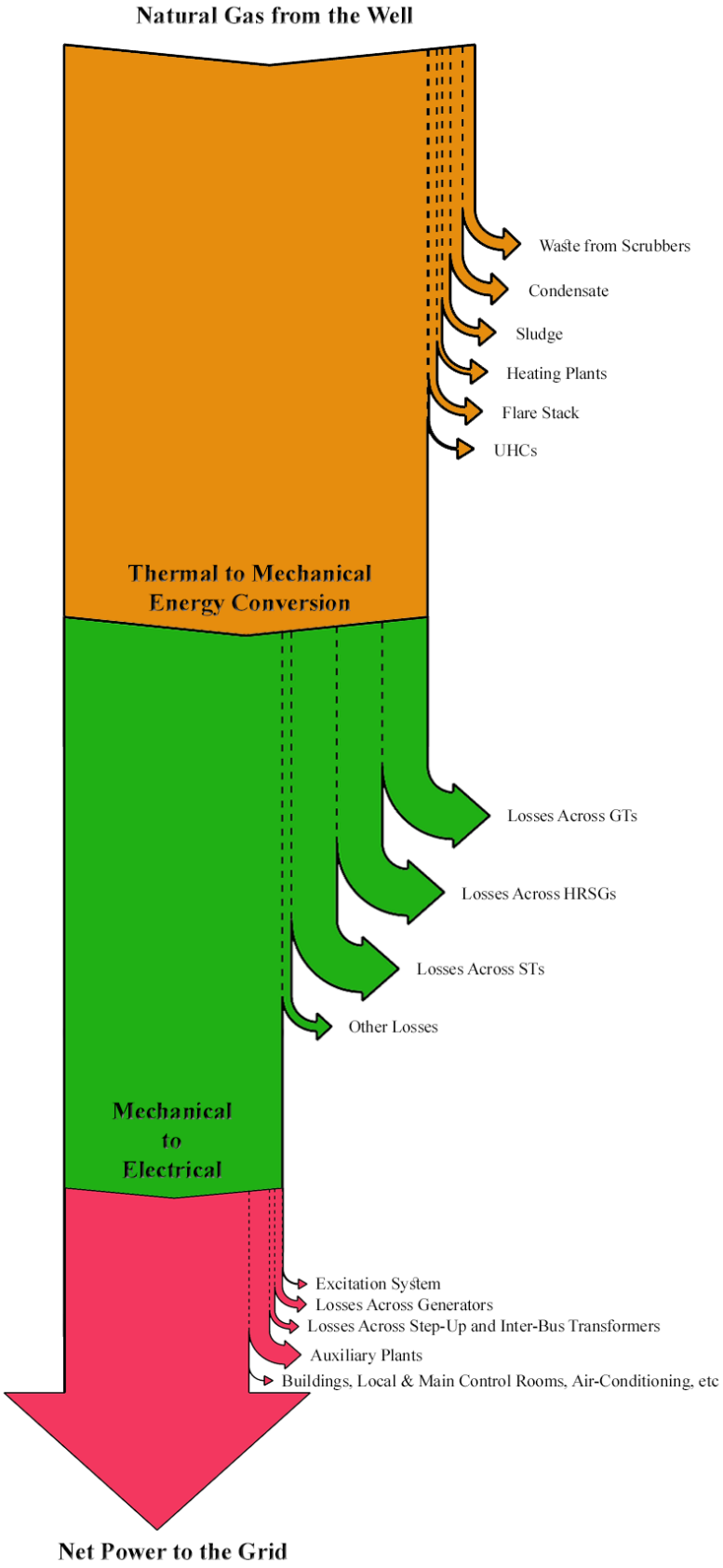


Figure 7.20: Sankey Diagram of Many Energy Losses That Need to Be Considered in the Realized ELD Model

7.4.5 Dynamic Weather Conditions

Due to the effects of weather conditions, degraded efficiency, and fuel type, the fuel-cost model coefficients will not be constant anymore[25]! To clarify this point, let's consider the conventional quadratic equation given in (3.7). If only the dynamic change of the ambient temperature $\breve{T}_a$ is taken into account, then the following piece-wise equation can be used if $\breve{T}_a \in [10°\text{C}, 40°\text{C}]$ and $\Delta\breve{T}_a = \pm 5°\text{C}$:

$$C_i^a = \begin{cases} \alpha_{1,0} + \alpha_{1,1}P_i + \alpha_{1,2}P_i^2, & \text{if } \breve{T}_a = 10°\text{C}; \\ \alpha_{2,0} + \alpha_{2,1}P_i + \alpha_{2,2}P_i^2, & \text{if } \breve{T}_a = 15°\text{C}; \\ \alpha_{3,0} + \alpha_{3,1}P_i + \alpha_{3,2}P_i^2, & \text{if } \breve{T}_a = 20°\text{C}; \\ \alpha_{4,0} + \alpha_{4,1}P_i + \alpha_{4,2}P_i^2, & \text{if } \breve{T}_a = 25°\text{C}; \\ \alpha_{5,0} + \alpha_{5,1}P_i + \alpha_{5,2}P_i^2, & \text{if } \breve{T}_a = 30°\text{C}; \\ \alpha_{6,0} + \alpha_{6,1}P_i + \alpha_{6,2}P_i^2, & \text{if } \breve{T}_a = 35°\text{C}; \\ \alpha_{7,0} + \alpha_{7,1}P_i + \alpha_{7,2}P_i^2, & \text{if } \breve{T}_a = 40°\text{C}. \end{cases} \tag{7.91}$$

As can be clearly seen from (7.91), the model requires many conditions to cover all the ambient temperature readings. The number of conditions increases as the span of $\breve{T}_a$ increases[26] and/or the step-size of the temperature change decreases[27]. To solve this issue, the preceding coefficients should be treated as dependent variables as follows:

$$C_i^a = \alpha_0\left(\breve{T}_a\right) + \alpha_1\left(\breve{T}_a\right)P_i + \alpha_2\left(\breve{T}_a\right)P_i^2 \tag{7.92}$$

which is now a very concise mathematical equation and can be used to represent any value of $\breve{T}_a$.

However, this model does not take into account many other weather variables, such as relative humidity RH, atmospheric pressure $\mathcal{P}_a$, wind speed ω_s, wind direction ω_d, etc. Also, nobody knows the behavior of such a realized model. Thus, to be on the safe side, the following general expression can be used:

$$C_i^a = f_i^a\left(P_i, \breve{T}_a, \text{RH}, \mathcal{P}_a, \omega_s, \omega_d, \cdots\right) \tag{7.93}$$

[25] Further explanation will be given later in Chapter 9/Sections 9.2-9.3.

[26] i.e., $\breve{T}_a^{\min} \downarrow$ and/or $\breve{T}_a^{\max} \uparrow \rightarrow \text{SPAN} = \breve{T}_a^{\max} - \breve{T}_a^{\min}$.

[27] i.e., the change in $\breve{T}_a \rightarrow \Delta\breve{T}_a = \left|\breve{T}_a^{\text{new}} - \breve{T}_a^{\text{old}}\right|$.

For spinning reserve units, the mathematical model given in (7.3) can be re-expressed as follows:

$$C_i^a = \begin{cases} f_i^a \left(\varpi_i, \breve{T}_a, \text{RH}, \mathcal{P}_a, \omega_s, \omega_d, \cdots \right), & \varpi_i \neq \text{FSNL or } P_i < 0 \\ g_i^a \left(P_i, \breve{T}_a, \text{RH}, \mathcal{P}_a, \omega_s, \omega_d, \cdots \right), & \text{otherwise} \end{cases} \tag{7.94}$$

and the same concept can be applied (7.4) as well.

For multi-fuel thermal units, the mathematical model given in (7.59) can be re-expressed as follows:

$$C_i^a = u_i^a \left(\dot{v}_{i,1}, \dot{v}_{i,2}, \cdots, \dot{v}_{i,M}, \breve{T}_a, \text{RH}, \mathcal{P}_a, \omega_s, \omega_d, \cdots \right) \tag{7.95}$$

As can be clearly seen in (7.91)-(7.95), the functions f_i^a, g_i^a, and u_i^a are unknown. Thus, there is a need to approximate many phenomena to realize existing classical ELD models. Because of the nonlinearity issue of real data and measurements, unfortunately, LR is not always a good choice. Adopting NLR creates many technical problems, such as selecting proper analytical function(s), placing parameters, guessing good initial starts, defining problem search space or side constraints, etc. Modern ML computing systems, such as ANNs and SVMs, could provide precise/accurate models. However, because they are black-boxes[28], so they are not good at this engineering application. All these facts motivate us to develop our own ML computing systems, which will be presented later in Chapter 10.

[28]They do not have solid mathematical equations.

Chapter 8

Realizing Optimal Relay Coordination Problems

The classical formulation of optimal relay coordination (ORC) problems are covered in Chapter 4 for directional overcurrent relays (DOCRs). The relay coordination stage is very crucial in any protection design that involves sets of primary and backup protective devices. Based on our comprehensive review on this topic, it has been found that all the ORC solvers presented in the literature are designed based on many assumptions and simplifications. For example, a common coordination time interval (CTI) is taken for all ϱ DOCRs, which is incorrect. Also, real electric power networks could have different types and models[1] of DOCRs. This means that the variable bounds of time multiplier settings (TMS) and plug settings (PS) and their **step-size resolutions** are not identical, and thus the variable bounds of relay operating times (T) are also not same. Further, Some primary relays have two backup relays installed on the same current flow direction; one is a **local backup relay** and the other is a **remote backup relay**. In the literature, this phenomenon still has not been realized yet. Add to that, the dynamic changes of the surrounding weather conditions (such as temperature, humidity, wind speed, and wind direction) and system conditions[2] (such as network topology, frequency, and current) are not accounted for. This chapter offers some corrections and realizations toward this protection problem. Although only inverse-time DOCRs are used in the following optimization models, the other types of DOCRs can be used. Moreover, this realistic model can be modified to integrate other overcurrent protective devices (such as OCRs and fuses) and other protective relays (such as distance and differential relays).

[1] We are talking here about relays having the same time-current characteristic curve (TCCC), but they are from different manufacturers. Even the same manufacturer's relays, they could have different models and hence different specifications and capabilities.

[2] Please, refer to Chapter 6.

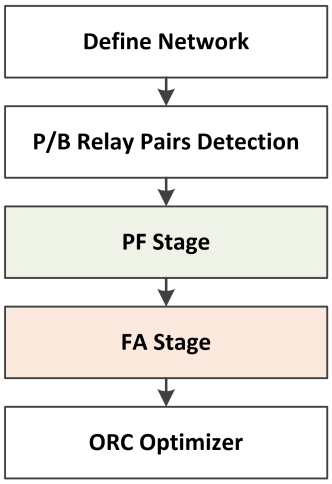


Figure 8.1: Flowchart of the Basic Optimal Relay Coordination Process

8.1 Optimal Relay Coordination Using the TFB Model

The ORC stage depends on two essential stages; the power flow (PF) analysis and the fault analysis (FA). The dependency on these two stages can be illustrated in Figure 8.1. Because electric power systems are subject to many **operational**[3] and **topological**[4] changes, so some stages of Figure 8.1 are repeated. The detailed flowchart of this dynamic ORC problem is shown in Figure 8.2.

In Chapter 6, it has been seen how the PF stage is directly affected by the model used in representing powerlines (i.e., transmission, sub-transmission, and distribution lines). Also, it has been seen how the FA stage is directly affected by the PF stage. Thus, $\breve{T}$ and f have multiple effects on the solution quality of the ORC stage. By referring to Figure 8.2, the variation of $\breve{T}$ and f can be considered as operational changes. Based on this, the dependability criterion of protection systems could be violated if these effects are not considered during solving their ORC problems. This means that the relay settings at one case of $\breve{T}$ and f could not be optimal or even feasible for other cases. Thus, $\breve{T}$ and f could jeopardize the whole electric system.

In this section, the simple IEEE 8-bus test system shown in Figure 6.27 is used to solve the optimal coordination problem of DOCRs. Again, the goal of this optimization problem is to achieve two things: 1. satisfying the correct coordination

[3]Such as changing the power settings of generators and loads.
[4]Such as taking busbars and lines out of service or returning them back into service.

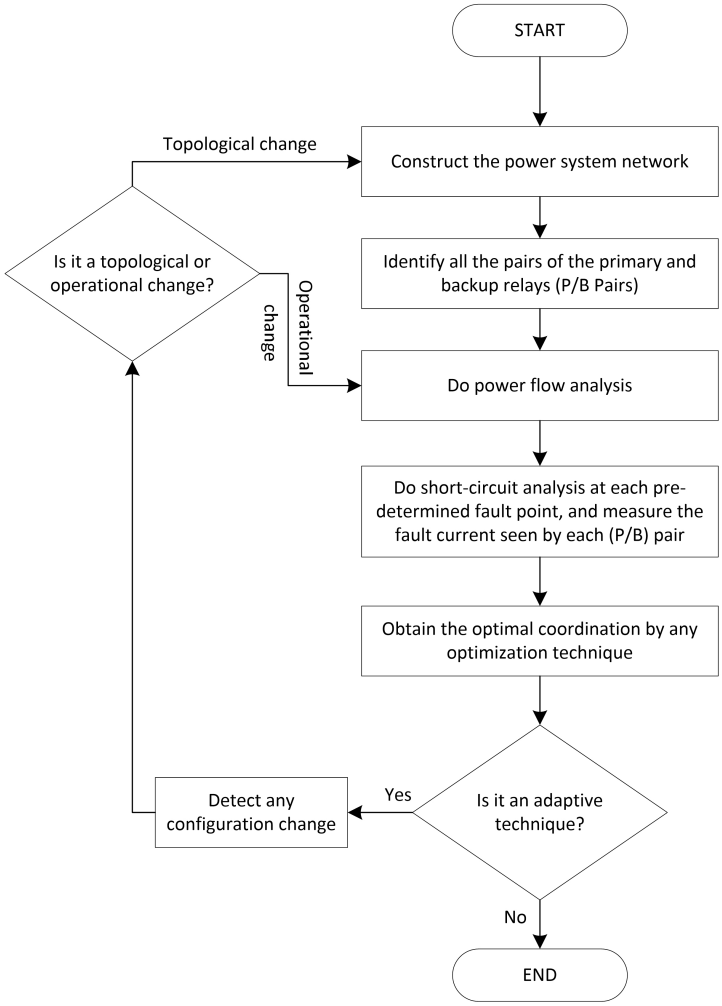


Figure 8.2: Detailed Flowchart of the Dynamic Optimal Relay Coordination Process

between P/B relay pairs, and 2. minimizing the total operating times of DOCRs when they act as primary protective devices [280]. Both can be done by optimally selecting TMS and PS of the 14 DOCRs shown in Figure 6.27. This semi-realistic ORC problem is solved using the same parameters given in [40, 280], but with high enough iterations to ensure that all the sensitivity effects are eliminated. That is, similar to the realistic ELD problem presented in Chapter 7/Section 7.1, the reason for using high iterations is to study only the effects of $\breve{T}$ and f; and not the optimization algorithm parameters. The hybrid BBO-LP algorithm covered in Chapter 2 and Chapter 4 is used for this mission. Table 8.1 shows the fitness under the four cases covered in Chapter 6 and Chapter 7 for $\breve{T}$ and f.

It has to be said that Figure 6.27 represents a very simple network. The effects

Table 8.1: $\breve{T}$ and f Effects on the Settings of the ORC Problem

Primary	**Case No.1**		**Case No.2**		**Case No.3**		**Case No.4**	
DOCR	*TMS*	*PS* (A)	*TMS*	*PS* (A)	*TMS*	*PS* (A)	*TMS*	*PS* (A)
R_1	0.1040	0.8	0.1042	0.8	0.1044	0.8	0.1043	0.8
R_2	0.1089	1.5	0.1104	1.5	0.1096	1.5	0.1105	1.5
R_3	0.1121	1.5	0.1137	1.5	0.1131	1.5	0.1138	1.5
R_4	0.1000	2.5	0.1000	2.5	0.1236	2.0	0.1000	2.5
R_5	0.1072	1.5	0.1064	1.5	0.1072	1.5	0.1124	1.5
R_6	0.1099	1.5	0.1099	1.5	0.1102	1.5	0.1000	2.0
R_7	0.1000	2.5	0.1000	2.5	0.1000	2.5	0.1000	2.5
R_8	0.1242	1.0	0.1479	0.8	0.1262	1.0	0.1250	1.0
R_9	0.1086	2.0	0.1117	2.0	0.1086	2.0	0.1078	2.0
R_{10}	0.1208	2.0	0.1000	2.5	0.1000	2.5	0.1200	2.0
R_{11}	0.1211	1.0	0.1222	1.0	0.1502	0.8	0.1222	1.0
R_{12}	0.1231	1.5	0.1246	1.5	0.1281	1.5	0.1247	1.5
R_{13}	0.1003	0.6	0.1014	0.6	0.1032	0.6	0.1015	0.6
R_{14}	0.1000	2.0	0.1281	1.5	0.1000	2.0	0.1000	2.0
Fitness	7.652695 s		7.750774 s		7.750901 s		7.763599 s	

of $\breve{T}$ and f increase as the line length increases. Also, these effects become more significant in countries that have very harsh weather. Furthermore, the parameters of lines and transformers and the settings of generators and loads play an important role in deciding whether the effects of $\breve{T}$ and f are significant or not.

8.2 Considering the Actual Settings of Different Relay Technologies in the Same Network

In the literature, many amazing optimization algorithms are proposed to solve ORC problems effectively and quickly. Such algorithms are reported in [14, 21, 40, 51, 70, 73, 127, 262, 280, 314]. A good literature review is given in [40, 51, 314]. All the existing studies are based on a general unrealistic assumption that all DOCRs have the same model[5]. If someone refers to modern electric power networks, he/she will find that different relay technologies are used in the same network. That is, electromechanical, static, digital "hardware-based" and numerical "software-based" DOCRs all could be installed in the same network. Thus, the existing ORC optimizers will not work unless

[5]Be careful, we are talking here about the relay model and not TCCC.

Table 8.2: Dimensions of ELD, Classical ORC and Realistic ORC Problems

Test System	ELD Problem	Classical DOCRs Coordination		Mixture of Different Technologies[a] ✓	
		One TCCC[b]	Multiple TCCCs	One TCCC ✓	Multiple TCCCs
3-Bus [367]	1-3	12	30	18	36
6-Bus [280]	2-4	28	70	42 ✓	84
15-Bus [51]	6-7	84	210	126 ✓	252
30-Bus [296]	2-5	60	150	90	180
42-Bus [40]	3	194	485	291 ✓	582

[a] ✓: means this case is simulated in this study.
[b] Any type of time-current characteristic curves (TCCCs); please refer to Table 4.2.

adjusting their classical models to accept different settings of TMS, PS, T and CTI for the same TCCC. Taking this consideration will increase the problem dimension by 1.5× if only one **unified TCCC** is used for all DOCRs, and by 1.2× if multiple TCCCs are used[6]. Table 8.2 gives some examples of five popular test systems.

This section tries to realize the existing ORC model to deal with different relay technologies. To validate its correctness, three test systems are solved by using BBO.

8.2.1 Mathematical Formulation

The new mathematical formulation of this semi-realistic ORC problem can be modeled in any n-dimensional optimization algorithm as follows:

8.2.1.1 Objective Function

Suppose that a mesh network has ß branches. If each branch is equipped with two DOCRs mounted on both ends, then there are $\varrho = 2ß$ DOCRs in that network. The operating time of the ith DOCR for a short-circuit fault occurred at the kth location can be mathematically expressed as follows [40]:

$$T_{i,k}^{y_i} = f(TMS_i^{y_i}, PS_i^{y_i}, I_{R_i^{y_i},k}) \quad , \quad i = 1, 2, \cdots, \varrho \tag{8.1}$$

where the superscript y_i denotes the technology of the ith relay R_i installed on one end or terminal of a faulty line, and $I_{R_i^{y_i},k}$ is a short-circuit current seen by the ith relay for a fault occurring at the kth location.

[6] Regardless of the relay technology used, remember that the ORC problem dimensions using multiple TCCCs are bigger than those when only one unified TCCC is used for all the relays. Thus, 1.2× is the gain applied to the dimensions of ORC problems having multiple TCCCs. This point is clarified in Table 8.2.

Because the current is independent of relay technology, so the superscript y_i can be dropped from the current term to have just $I_{R_i,k}$. Based on (8.1), the operating time of a relay installed on the ith terminal depends on the technology of that relay itself. If that relay is replaced with a relay having different technology, then $T_{i,k}^{y_i}$ will have a different curve. This creates many problems during optimizing the operating times of these ϱ DOCRs by existing mathematical models. The operating time $T_{i,k}^{y_i}$ can be numerically determined by using the IEC/BS and ANSI/IEEE standard models given in [177, 179] as follows:

$$T_{i,k}^{y_i} = TMS_i^{y_i} \left[\frac{\beta}{\left(\frac{I_{R_i,k}}{PS_i^{y_i}} \right)^{\alpha} - 1} + \gamma \right], \quad \begin{matrix} i = 1, \cdots, \varrho \\ k = 1, \cdots, l \end{matrix} \tag{8.2}$$

If 3ϕ short-circuit analysis is carried out for l fault locations on each line, then the objective function can be derived by minimizing the sum of operating times of all ϱ DOCRs when they act as primary relays:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} \sum_{k=1}^{l} w_{i,k} T_{i,k}^{y_i,pr} \tag{8.3}$$

where $w_{i,k}$ is the weight given to the fault occurred at the kth location. For the sake of simplicity, fa weight of 1 is used for all the fault locations [367]. Also, most of the studies presented in the literature consider just the fault located at the near-end point of each relay. Thus, (8.3) can be simplified to:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} T_i^{y_i,pr} \tag{8.4}$$

To ensure that the solutions obtained from minimizing (8.4) are feasible and useful, the following design constraints should be satisfied:

8.2.1.2 Selectivity Constraint Among Primary and Backup Relay Pairs

The goal of satisfying this constraint is to ensure that each primary protective relay has enough chance to isolate the fault occurred in its zone. If it fails to operate within that period, then one or more backup relays located in the upstream should act. This inequality constraint can be mathematically expressed as follows:

$$T_{j,k}^{u_j} \geqslant T_{i,k}^{y_i} + CTI_i^{y_i} \tag{8.5}$$

where $T_{j,k}^{u_j}$ is the operating time of the jth backup relay R_j for a fault occurred at the kth out-zone location of the ith primary relay, and the superscript u_j denotes the technology of the jth backup relay R_j. $CTI_i^{y_i}$ is the coordination time interval (i.e., the allowable chance) given to the ith primary relay R_i to clear that fault [52, 74].

From (8.5), it can be clearly seen that CTI is not constant anymore. Similar to (8.2), $T_{j,k}^{u_j}$ can be calculated for the fault located at the kth point as follows:

$$T_{j,k}^{u_j} = TMS_j^{u_j} \left[\frac{\beta}{\left(\frac{I_{R_j,k}}{PS_j^{u_j}} \right)^{\alpha} - 1} + \gamma \right], \begin{matrix} j = 1, \cdots, \varrho \\ k = 1, \cdots, l \end{matrix} \tag{8.6}$$

where $I_{R_j,k}$ is a short-circuit current seen by the jth backup relay R_j for a fault occurring at the kth location (i.e., belonging to the ith primary relay R_i).

8.2.1.3 Inequality Constraints on Relay Operating Times

From (8.2), the minimum and maximum values of $T_{i,k}^{y_i}$ depend on the technology of that ith relay R_i. Thus, by mentioning the technology type y_i, the operating time $T_{i,k}^{y_i}$ can be bounded between two bounds as follows:

$$T_{i,k}^{y_i,\min} \leqslant T_{i,k}^{y_i} \leqslant T_{i,k}^{y_i,\max} \tag{8.7}$$

where $T_{i,k}^{y_i,\min}$ and $T_{i,k}^{y_i,\max}$ are respectively the minimum and maximum operating times of R_i for a fault occurring at the kth location. The lower bound $T_{i,k}^{y_i,\min}$ depends on the technology of the ith relay y_i, and the applicable upper bound $T_{i,k}^{y_i,\max}$ is determined based on the critical clearing time $T_{i,k}^{y_i,\text{cr.}}$ required to preserve system stability [40, 213, 291].

Because $T_{i,k}^{y_i}$ is a dependent variable, as can be obviously observed in (8.1)-(8.2), so (8.7) is a functional constraint. Based on this, the constraint must be re-expressed to have the following two inequality constraints [40]:

$$T_{ik}^{y_i,\min} - T_{i,k}^{y_i} \leqslant 0 \tag{8.8}$$

$$T_{i,k}^{y_i} - T_{i,k}^{y_i,\max} \leqslant 0 \tag{8.9}$$

8.2.1.4 Side Constraints on Relay Time Multiplier Settings

For TMS, each R_i is manufactured with a specific domain and step-size resolution predefined based on y_i. This side constraint is formulated for R_i as follows:

$$TMS_i^{y_i,\min} \leqslant TMS_i^{y_i} \leqslant TMS_i^{y_i,\max} \tag{8.10}$$

where $TMS_i^{y_i,\min}$ and $TMS_i^{y_i,\max}$ are respectively the minimum and maximum allowable limits of TMS of the ith relay manufactured based on the technology[7] y_i.

8.2.1.5 Side Constraints on Relay Plug Settings

Similar to $TMS_i^{y_i}$, the domain and step-size resolution of $PS_i^{y_i}$ depend on the relay technology. However, the same guidance used in Chapter 4 is applied here as well. Thus, this side constraint can be expressed for R_i as follows [40, 51, 280]:

$$PS_i^{y_i,\min} \leqslant PS_i^{y_i} \leqslant PS_i^{y_i,\max} \tag{8.11}$$

where $PS_i^{y_i,\min}$ and $PS_i^{y_i,\max}$ are respectively the minimum and maximum allowable limits of PS of the ith relay manufactured based on the technology y_i.

8.2.2 Biogeography-Based Optimization Algorithm

The detailed information about BBO is given in Chapter 2, while Chapter 4 shows how to employ that algorithm in solving ORC problems. This subsection covers some essential modifications to make BBO applicable.

8.2.2.1 Clear Duplication Stage

This optional stage can be activated to increase the diversity of islands by avoiding the features of one island being duplicated on other islands. However, the clear duplication stage is completely disabled in this study because the realistic settings of TMS and PS of all the relay technologies are discrete [21]. Thus, the collateral damage of this clear duplication stage is more than the expected benefit.

[7] This is the difference here compared with other similar studies. In this realization, even with the same TCCC, the variable bounds and step-size resolution are not constant anymore.

8.2.2.2 Avoiding Facing Infeasible Selectivity Constraints

Based on the work done in Chapter 4, the selectivity constraint given in (8.5) for each P/B relay pair is satisfied by initiating a *while*-loop to generate random y_i, $PS_i^{y_i}$, and $TMS_i^{y_i}$ until getting feasible values. This pre-processing unit is known in Chapter 4 as the feasibility checker (FC), which can accelerate finding feasible solutions to this highly constrained nonlinear non-convex mixed-integer ORC problem.

8.2.2.3 Linear Programming Stage

Because both TMS and PS are discrete variables and each relay has its own realized settings, so the linear programming (LP) sub-algorithm proposed in Chapter 4 is deactivated[8].

8.2.2.4 Linking $PS_i^{y_i}$ and $TMS_i^{y_i}$ with y_i

It is important to note that the settings $PS_i^{y_i}$ and $TMS_i^{y_i}$ are randomly generated based on their technology y_i. This means that the first variable that needs to be randomly generated is y_i, which is a discrete value that lies between 1 and 3 (1 for electromechanical or electromagnetic, 2 for static or solid-state, and 3 for digital or numerical). Based on this value, the step-size resolutions and the lower and upper bounds of $PS_i^{y_i}$ and $TMS_i^{y_i}$ are updated.

By integrating all the above modifications and taking into account the considerations recommended in [43, 44], the final design of the BBO algorithm used to solve this ORC problem is shown in Figure 8.3.

8.2.3 Numerical Experiments

The performance of the proposed BBO algorithm with this realistic ORC model is evaluated using the IEEE 6-bus, 15-bus, and 42-bus test systems. The programs are coded in MATLAB R2018b using the following computing machine: Alienware X51, i7-6700 CPU @ 3.4 GHz, 8 GB RAM with 64 bit Windows 10 operating system. The simulation parameters used in each relay technology are tabulated in Table 8.3. The BBO parameters are: population size of 50, 4 elite solutions, non-death penalty

[8] If this fine-tuning stage must be activated, then LP should be replaced by ILP. Please, refer to Chapter 4/Subsection 4.3.2.

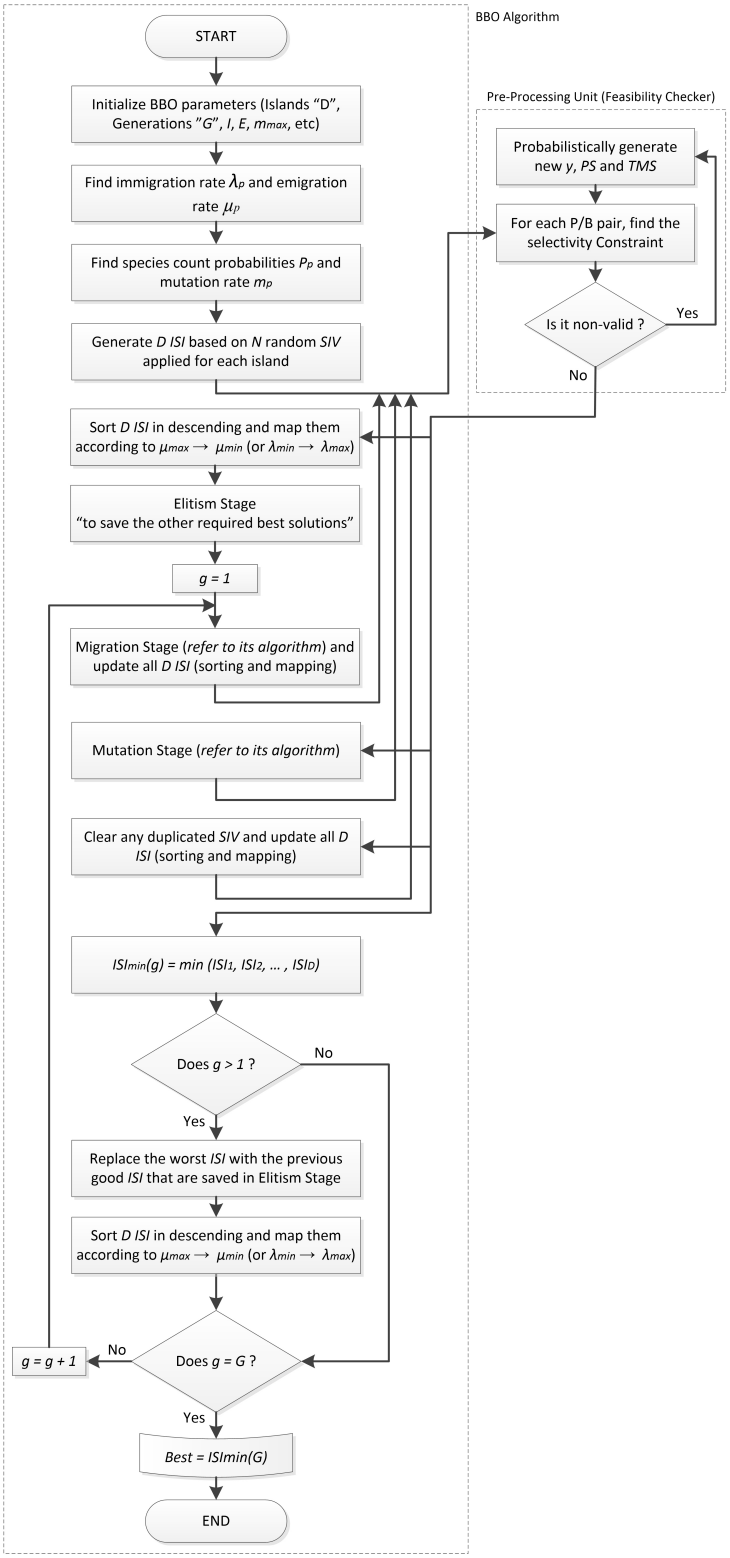


Figure 8.3: Flowchart of the Modified BBO Algorithm

Table 8.3: Simulation Parameters Used in Each Relay Technology [46,51,56,366,395]

Technology	TMS			PS (A)			T (s)		CTI
Used	$TMS^{\min}$	$TMS^{\max}$	Step-size	$PS^{\min}$	$PS^{\max}$	Step-size	$T^{\min}$	$T^{\max}$	(s)
Numerical[a]	0.025	1.5	0.001	0.5	2.5	0.01	0.01	4.0[b]	0.3
Static	0.05	1.3	0.025	0.5	2.5	0.1	0.03	4.0	0.35
Electromechanical	0.1	1.1	0.05	{0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 2.5}[c]			0.05	4.0	0.4[d]

[a]The settings of directional and non-directional OCRs of AREVA Micom/P12xy series are used.
[b]Although it can go beyond this limit, it is restricted to this value to avoid exceeding the stability critical clearing time.
[c]The original settings of the 15-bus test system are given as a vector of steps of 0.5 A between $PS^{\min}$ and $PS^{\max}$.
[d]The original CTI of the 15-bus test system is 0.2 s, which is impractical.

function with a penalty factor of $r = 30$, $m_{\max} = 0.1$, 30 trials, and each trial uses 200 generations for the 6-bus test system and 1000 generations for the 15-bus and 42-bus test systems. The dimensions of these three test systems are given in Table 8.2. Finally, the IEC's standard inverse TCCC is used to model all ϱ DOCRs. Thus, the coefficients $\{\alpha, \beta, \gamma\}$ of (8.2) and (8.6) are respectively set to $\{0.02, 0.14, 0\}$.

8.2.3.1 Test Case I: 6-Bus System

This test system is shown in Figure 8.4 where all the necessary information is available in [25]. The full result obtained by BBO for this test system is listed in Table 8.4. As can be clearly seen, the optimal settings of each ith relay depend on the technology of that relay itself.

8.2.3.2 Test Case II: 15-Bus System

This test system is shown in Figure 4.9 where all the necessary information is given in [25]. The full result obtained by BBO for this test system is listed in Table 8.5. Again, the optimal solution shows that the relay technology can decide whether the settings and decimal places are feasible or not; even if only the IDMT curve is used.

8.2.3.3 Test Case III: 42-Bus System (IEEE Std. 399-1997)

This test system is known as the IEEE Std 399-1997 network[9]. It is one of the largest ORC test systems available in the literature [40]. The short-circuit analysis is carried

[9]It is available in the IEEE Standards - Brown Book.

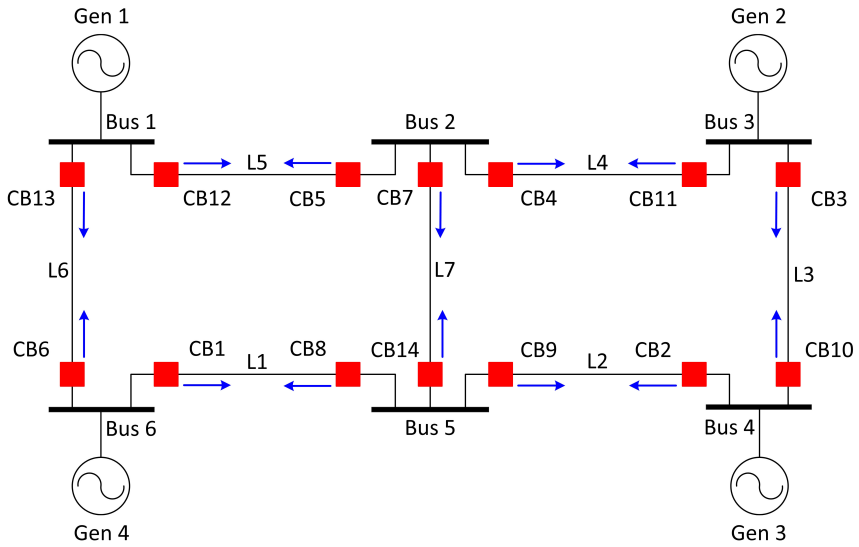


Figure 8.4: Single-Line Diagram of the IEEE 6-Bus Test System

Table 8.4: The Full Result Obtained for Test Case I (IEEE 6-Bus System)

Primary	**Optimal Settings**			
Relay	PS (A)	TMS	T (s)	Type
R_1	2.42	0.555	1.0899	Numerical
R_2	1.2	0.363	0.7641	Numerical
R_3	0.8	0.475	0.5747	Static
R_4	1.7	0.15	0.3458	Static
R_5	2.5	0.125	0.4379	Static
R_6	2	0.2	0.3705	Electromechanical
R_7	1	0.5	1.0146	Electromechanical
R_8	0.6	0.275	0.5828	Static
R_9	1.5	0.25	0.5243	Electromechanical
R_{10}	2	0.2	0.5363	Electromechanical
R_{11}	0.7	0.625	0.7354	Static
R_{12}	1.5	0.85	1.3248	Electromechanical
R_{13}	1.8	0.1	0.1813	Static
R_{14}	1.74	0.492	1.1233	Numerical
Best (s)			9.6056	
Worst (s)			13.0547	
Mean (s)			11.5629	
StDev (s)			0.7404	

Table 8.5: The Full Result Obtained for Test Case II (IEEE 15-Bus System)

Primary	Optimal Settings				Primary	Optimal Settings			
Relay	PS (A)	TMS	T (s)	Type	**Relay**	PS (A)	TMS	T (s)	Type
R_1	1.1	0.85	1.9087	Static	R_{22}	0.5	0.9	1.5587	Electromechanical
R_2	2.4	0.225	0.7426	Static	R_{23}	1.5	0.75	1.9572	Static
R_3	1.83	0.974	2.5441	Numerical	R_{24}	0.5	0.95	1.7589	Electromechanical
R_4	1.5	0.825	2.2535	Static	R_{25}	1.3	0.85	2.1562	Static
R_5	2	0.8	2.3385	Electromechanical	R_{26}	1.9	0.525	1.5535	Static
R_6	2.3	0.6	1.8161	Static	R_{27}	2.13	0.561	1.8647	Numerical
R_7	1.23	1.144	2.7520	Numerical	R_{28}	2.23	0.607	1.8509	Numerical
R_8	1.8	0.575	1.6471	Static	R_{29}	2.5	0.375	1.0934	Static
R_9	1.25	0.794	1.8122	Numerical	R_{30}	2.1	0.475	1.3908	Static
R_{10}	1.6	0.7	1.8113	Static	R_{31}	0.78	1.192	2.3559	Numerical
R_{11}	2.31	0.368	1.2260	Numerical	R_{32}	0.64	1.264	2.5989	Numerical
R_{12}	1.48	0.539	1.4907	Numerical	R_{33}	1.54	0.981	2.6531	Numerical
R_{13}	1.83	0.808	2.2500	Numerical	R_{34}	1	0.8	1.7716	Electromechanical
R_{14}	1.2	0.7	1.7190	Static	R_{35}	1.3	0.9	2.3630	Static
R_{15}	1.55	0.979	2.6312	Numerical	R_{36}	0.7	1.125	2.2534	Static
R_{16}	0.6	1.05	2.0697	Electromechanical	R_{37}	1.62	1.04	2.7890	Numerical
R_{17}	1.2	0.875	2.0002	Static	R_{38}	2	0.925	2.9179	Static
R_{18}	2.13	0.364	0.9880	Numerical	R_{39}	0.7	1	2.0893	Static
R_{19}	1.14	1.045	2.2970	Numerical	R_{40}	1.2	1.125	2.7400	Static
R_{20}	1.6	0.675	1.6995	Static	R_{41}	2.16	0.849	2.3826	Numerical
R_{21}	0.5	0.95	1.6141	Electromechanical	R_{42}	1.94	0.496	1.4353	Numerical
Best (s)					83.1461				
Worst (s)					94.2674				
Mean (s)					87.8263				
StDev (s)					3.0954				

out using DIgSILENT PowerFactory software. The single-line diagram of this test system is shown in Figure 8.5 [178]. More details about this test system are given in Appendix Q. With the same simulation parameters used in the last test system, the best solution obtained by the BBO algorithm is tabulated in Table 8.6. It has to be said that all the simulations done in this section are feasible with realized settings.

8.2.4 Further Discussion

It is known that there are many manufacturers available in the market for each relay technology. These relays come with different models and versions. Thus, to make the results more realistic, each relay setting could have a slight difference based on its manufacturer, model, and version. The model developed in this section can deal with this extended realistic situation.

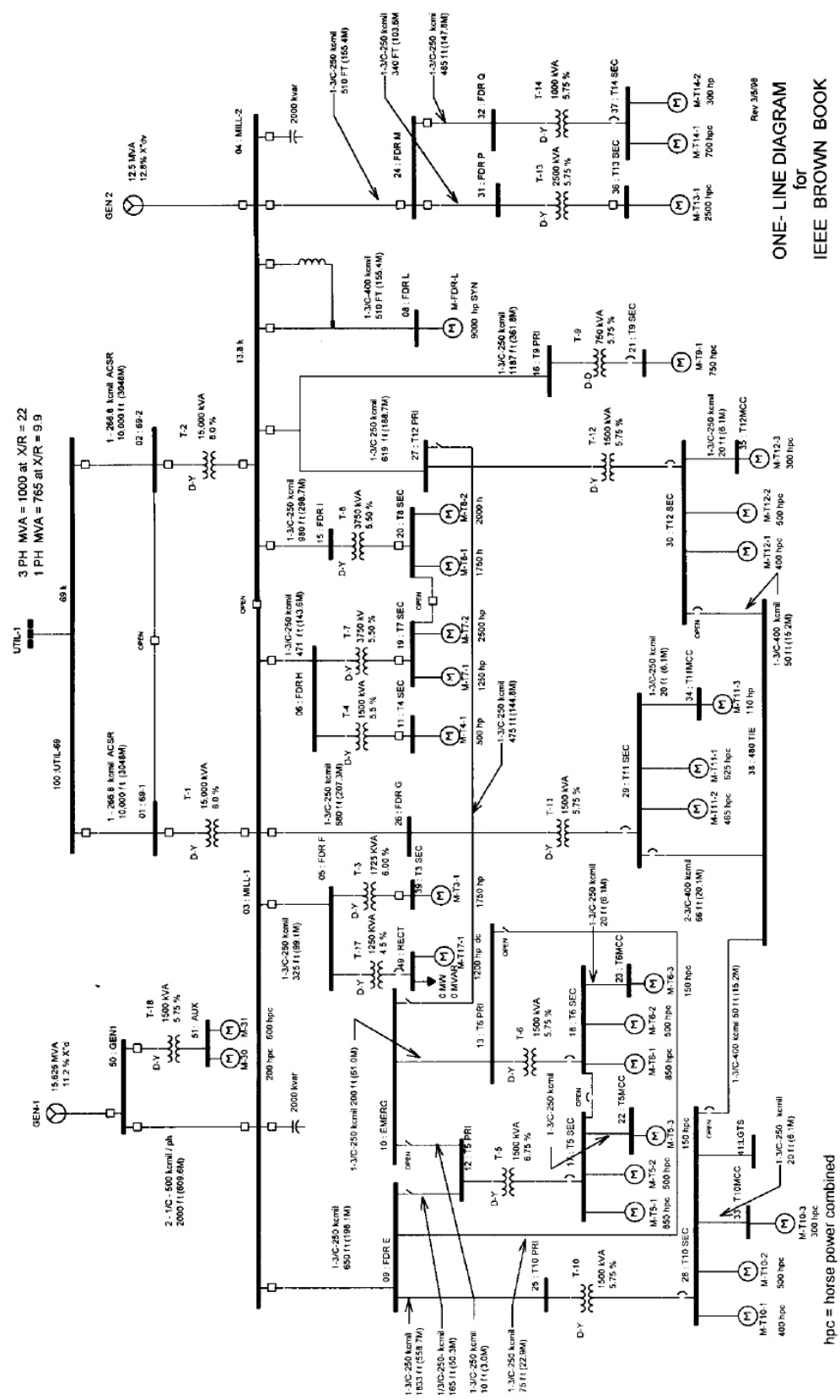


Figure 8.5: Single-Line Diagram of the IEEE 42-Bus Test System (IEEE Std. 399-1997)

Table 8.6: The Full Result Obtained for Test Case III (IEEE Std. 399-1997)

Primary	**Optimal Settings**				**Primary**	**Optimal Settings**			
Relay	PS (A)	TMS	T (s)	Type	**Relay**	PS (A)	TMS	T (s)	Type
R_1	2.24	1.363	2.9341	Numerical	R_{50}	1	0.9	1.0787	Static
R_2	1.84	1.435	3.7200	Numerical	R_{51}	0.8	1.1	1.2591	Electromechanical
R_3	1.46	1.291	2.7188	Numerical	R_{52}	1.34	1.424	1.5065	Numerical
R_4	2.3	0.822	1.9418	Numerical	R_{53}	2	0.65	1.3652	Electromechanical
R_5	0.84	1.142	1.9782	Numerical	R_{54}	0.5	0.9	1.3400	Static
R_6	1.88	1.006	1.0710	Numerical	R_{55}	1.5	0.45	0.8978	Electromechanical
R_7	2.23	1.039	1.7771	Numerical	R_{56}	1.92	1.049	1.2776	Numerical
R_8	1.4	1.2	1.8268	Static	R_{57}	2.5	0.9	1.0092	Electromechanical
R_9	0.9	0.775	2.2393	Static	R_{58}	0.5	1.1	1.6292	Electromechanical
R_{10}	1.48	0.611	2.2450	Numerical	R_{59}	1.13	0.773	1.2288	Numerical
R_{11}	2.37	0.975	1.5289	Numerical	R_{60}	2.46	0.802	0.9036	Numerical
R_{12}	1.4	1.2	2.4689	Static	R_{61}	1.29	0.539	0.6475	Numerical
R_{13}	1.1	1.25	2.3934	Static	R_{62}	1	0.7	1.2074	Electromechanical
R_{14}	2.5	0.95	1.5092	Electromechanical	R_{63}	2.5	0.275	0.6430	Static
R_{15}	2	0.85	1.8407	Static	R_{64}	1.85	0.551	0.5524	Numerical
R_{16}	2.5	0.75	1.7527	Static	R_{65}	1.3	0.35	0.6465	Static
R_{17}	2.4	0.975	1.6306	Numerical	R_{66}	0.61	0.606	0.8458	Numerical
R_{18}	1.5	0.45	0.7739	Electromechanical	R_{67}	1	0.3	0.3689	Electromechanical
R_{19}	1	0.8	1.0812	Electromechanical	R_{68}	0.84	0.155	0.1843	Numerical
R_{20}	2.5	1.3	2.1897	Static	R_{69}	0.99	0.408	0.4595	Numerical
R_{21}	2.5	1	1.6904	Static	R_{70}	0.7	0.85	0.9064	Static
R_{22}	2.44	1.101	1.8582	Numerical	R_{71}	1.3	0.125	0.1446	Static
R_{23}	2.02	1.125	1.7899	Numerical	R_{72}	1.73	0.123	0.1187	Numerical
R_{24}	1.7	1.175	1.8014	Static	R_{73}	2.5	0.075	0.1686	Static
R_{25}	1.13	1.481	2.0651	Numerical	R_{74}	1.5	0.05	0.0990	Static
R_{26}	1.5	0.225	0.3710	Static	R_{75}	1	0.65	0.6659	Electromechanical
R_{27}	0.5	1.025	1.2176	Static	R_{76}	1.36	0.238	0.3736	Numerical
R_{28}	1.57	0.346	0.4234	Numerical	R_{77}	1	0.7	0.6270	Electromechanical
R_{29}	1.9	0.975	1.5447	Static	R_{78}	0.69	0.451	0.4228	Numerical
R_{30}	2.5	0.35	0.4328	Electromechanical	R_{79}	1.56	0.153	0.1437	Numerical
R_{31}	1	0.55	0.6250	Electromechanical	R_{80}	0.5	0.5	0.4383	Electromechanical
R_{32}	1.15	0.574	0.8326	Numerical	R_{81}	1.5	0.325	0.5588	Static
R_{33}	2	0.925	1.4946	Static	R_{82}	0.5	0.1	0.1134	Static
R_{34}	1.58	0.609	0.7636	Numerical	R_{83}	1.7	0.2	0.3311	Static
R_{35}	1.7	0.2	0.1999	Static	R_{84}	1	0.175	0.2420	Static
R_{36}	1.5	0.6	1.1695	Electromechanical	R_{85}	1.9	0.225	0.2479	Static
R_{37}	0.8	1.025	1.0060	Static	R_{86}	2.17	0.094	0.0947	Numerical
R_{38}	1	1.075	1.0887	Static	R_{87}	1.67	0.111	0.1054	Numerical
R_{39}	2.21	0.499	1.1558	Numerical	R_{88}	1.19	0.073	0.0736	Numerical
R_{40}	2.2	0.725	0.8604	Static	R_{89}	2.48	0.09	0.0867	Numerical
R_{41}	1.32	1.192	1.2629	Numerical	R_{90}	2.43	0.194	0.1974	Numerical
R_{42}	1.94	1.155	1.8374	Numerical	R_{91}	2.19	0.121	0.1354	Numerical
R_{43}	1.6	0.1	0.1010	Static	R_{92}	1.5	0.2	0.2182	Electromechanical
R_{44}	2.49	1.156	1.3708	Numerical	R_{93}	1.7	0.075	0.1095	Static
R_{45}	1	1.091	1.8941	Numerical	R_{94}	1	0.35	0.3100	Electromechanical
R_{46}	2	0.8	1.7036	Electromechanical	R_{95}	1.5	0.225	0.2102	Static
R_{47}	0.8	0.7	1.1496	Static	R_{96}	2.3	0.05	0.0499	Static
R_{48}	2.3	0.675	1.1229	Static	R_{97}	1.27	0.065	0.1224	Numerical
R_{49}	2.4	0.75	1.2553	Static	-	-	-	-	-
Best (s)					100.0754				
Worst (s)					116.0101				
Mean (s)					107.3341				
StDev (s)					4.0406				

8.3 Considering Double Primary Relay Strategy

DOCRs, which are special types of OCRs, can be categorized as follows:

- Hardware-Based Protective Relays:
  1. Electromechanical (or Electromagnetic) Relays
  2. Solid-State (or Static) Relays
  3. Digital Relays
- Numerical (or Processor-Based) Protective Relays:
  4. Microprocessor-Based Relays
  5. DSP-Based Relays

From the above classification, the **electromechanical** or **electromagnetic relays** (such as **moving-coil**, **attracted armature**, **induction**, and **motor-operated** devices) are the first-generation OCRs, having appeared early in the last century (in the 1900s). They are limited by the need for **periodic maintenance** (**PM**) and calibration because of their mechanical moving parts. Also, they have few discrete values of PS and TMS, which means that the feasible search space is very confined and thus it is hard to be feasibly optimized. In addition, they have a slower response due to their over-shoot delay T_{OS} caused by **inertia**. Moreover, their simple technology makes them blind to each other, and the designer is forced to use three to four relays to protect all the lines; the three phases and ground. However, these devices have some advantages, such as their stability and insensitivity to the network conditions, and they are still in service because of their life-span. Furthermore, because of their longevity, there are many skilled experts to coordinate this kind of relays [65, 248, 251, 292].

The second-generation OCRs were made based on **analog electronics** technology to mimic the first-generation OCRs. These relays are called **solid-state** or **static relays**. The main technical problems faced with these relays are their stability and precision. For example, the ambient temperature can affect the device stability. Also, precise **passive components** (such as resistors, capacitors, and inductors) are required to reduce the total error [147].

In less than one decade, the third-generation **digital relays** appeared. Many manufacturers successfully designed new techniques to establish **two-way communications** between relays through some standard protocols and media [46, 126, 147].

Although all the inherent weaknesses of the electromechanical and solid-state relays can be permanently resolved by the third-generation relays, they are still hardware-based. Thus, to have **programmable relays**, the fourth-generation relays were invented. Some of these relays are designed to act as **general-purpose relays**. These innovative relays are called **numerical relays**; and because of their capabilities, some researchers considered them as **intelligent electronic devices** (**IEDs**). They can be manufactured based on **micro-controllers**, **micro-processors**, or even **digital signal processors** (**DSPs**) for high computational applications. All the relay settings, instructions, and operations can be updated, modified, or even upgraded through some special software provided by their original equipment manufacturers (OEMs) [147, 193, 319, 387].

This historical account is essential since it emphasizes that real electric power systems may contain many types and models of protective relays at the same time. During **upgrading/retrofit/rehabilitation** phases, there are two possible alternatives that can be selected by protection engineers, either:

- Completely replacing all electromechanical, static, and digital "hardware-based" relays with numerical relays, or;
- Integrating these old relays with numerical relays as a second line of defense.

If the first approach is followed, then the upgrading phase is straightforward without dealing with "**out of stock**" or "**obsolete**" spare parts. Besides, simplicity is preserved, and the classical ORC model can be used. However, this protection design requires removing all outdated infrastructures and replacing gained skill-sets. In contrast, while the second approach requires more maintenance it allows upgrading existing protection systems and providing additional backup protection. Both approaches can be seen in many realistic electric power networks; where electromechanical, static, and/or digital "hardware-based" relays are either completely replaced or partially retained to work in parallel with numerical relays.

This section focuses on the second approach where both old and new DOCRs are

assumed to protect electric networks from the same fault. These electromechanical, static, and/or digital "hardware-based" relays (i.e., old relays) mounted at the same points (the left and right ends of powerlines) of numerical relays are called "local-backup" or "**secondary**" relays. Some protection engineers called these new and old relays as "**main-1**" and "**main-2**" relays, respectively, instead of primary and local-backup/secondary relays. The logical reason behind that comes from the fact that when there is a fault the corresponding local-backup/secondary relays can act without waiting the chance given to numerical relays to operate plus their time delays explained by (4.1) [267, 292, 307]. Therefore, these old DOCRs work independently and they could act before new DOCRs. Thus, the terms (main-1 and main-2 DOCRs) are more suitable than the terms (primary and local-backup/secondary DOCRs). If the protection design based on main-1 and main-2 DOCRs is adopted, then it is called a "**double primary relay strategy** (**DPRS**)"; and this special case of ORC is the core of this section.

In real-world applications, this DPRS is mostly applied to just a few, or even many (but not all), busbars. However, we try to solve the **extreme case** of DPRS. That is, each terminal of powerlines has both main-1 and main-2 DOCRs where main-1 DOCRs are always taken as numerical relays and main-2 DOCRs could be electromechanical or static[10]. This extreme case has two possible solutions:

- If there is no violation, then DPRS can be applied at any P/B relay pair. This phenomenon can be seen in small test systems.

- Because the test systems used with DPRS are feasibly solved in the literature, so the violations of this optimization problem have a different meaning. They give important information about the spots where DPRS is not applicable[11]. This phenomenon can be seen in large test systems.

Solving this extreme case means that the problem dimension is increased by 100% (i.e., the problem is duplicated). Therefore, if the IEEE 3-bus test system shown in Figure 8.6 is used, then the number of variables, i.e. PS and TMS, increases from

[10]The digital relays can also be considered as main-2 relays, but they are not included because their settings are almost identical to those of numerical relays. The objective here is to analyze the feasibility when different relay settings and step-size resolutions are involved.

[11]In other words, these violations can tell which busbars are incompatible with DPRS.

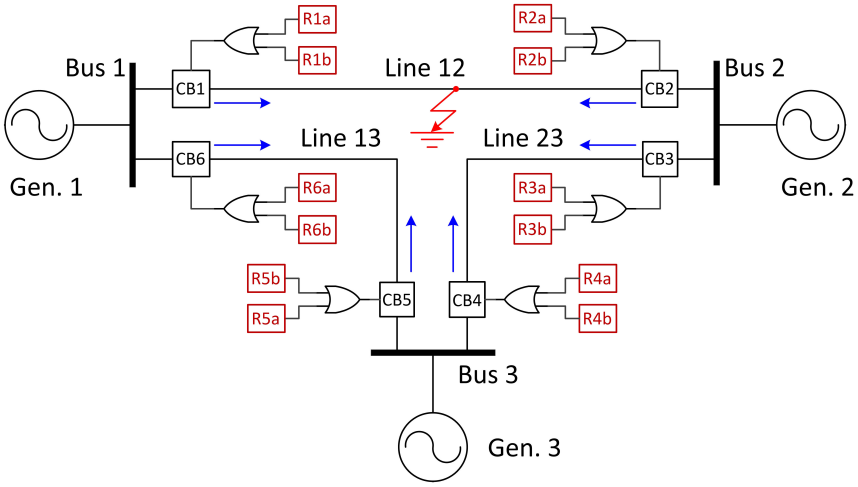


Figure 8.6: Single-Line Diagram of the IEEE 3-Bus Test System Equipped with Main-1 "a" and Main-2 "b" DOCRs

being 12 variables (for 6 DOCRs) to 24 variables (for 12 DOCRs). The complexity of this problem increases considerably if the TCCCs of these DOCRs are not identical (i.e., using multiple IEEE and/or IEC standards, such as inverse, very inverse, extremely inverse, and even - in some cases - user-defined TCCCs). Thus, the coefficients $\{\alpha, \beta, \gamma\}$ are not constants anymore, which means that each DOCR has 5 variables instead of 2. Based on this, if these DOCRs (i.e., the relays installed in the IEEE 3-bus test system) are equipped with asymmetrical TCCCs, then the dimension of that ORC problem further increases from being 24 variables (for one unified TCCC) to 60 variables (for multiple TCCCs). Table 8.7 compares this special ORC problem to the classical one. If the IEEE 42-bus test system (IEEE Std. 399-1997) is selected, then the dimension of its ELD problem is just 3 (i.e., three units), while it has 194 TMS variables and 194 PS variables (total is 388) for this special ORC problem. If multiple TCCCs are used to coordinate these DOCRs, then the dimension increases to 970 variables. This section uses one unified TCCC for all DOCRs. The other highly complicated scenario, which is based on multiple TCCCs, could be covered in future work.

To solve this special ORC problem with one unified TCCC for all DOCRs, the hybrid BBO/DE optimization algorithm presented in Chapter 2 is applied here, but after modifying it to act as a combinatorial algorithm. Again, this algorithm is reinforced with a modified FC as a sub-algorithm to avoid any infeasible settings of

Table 8.7: Dimensions of ELD, Classical ORC and DPRS-Based ORC Problems

Test System	ELD Problem	Classical DOCRs Coordination		Double Relays Strategy[a] ✓	
		One TCCC[b]	Multiple TCCCs	One TCCC ✓	Multiple TCCCs
3-Bus [367]	1-3	12	30	24	60
6-Bus [280]	2-4	28	70	56 ✓	140
15-Bus [51]	6-7	84	210	168 ✓	420
30-Bus [296]	2-5	60	150	120	300
42-Bus [40]	3	194	485	388 ✓	970

[a] ✓: means this case is simulated in this study.
[b] Any type of time-current characteristic curves (TCCCs); for example IDMT standard.

P/B relay pairs. Solving the extreme case of this ORC problem ensures detecting which P/B relay pairs accept DPRS when this strategy is not adopted for all circuit breakers. The main achievements of this section are:

1. This is the first time to address this realistic ORC problem through a new mathematical model.

2. This highly constrained constrained nonlinear non-convex mixed-integer ORC problem is solved by using a new fully discretized hybrid BBO/DE optimization algorithm.

3. It modifies the existing FC to work with this new realistic ORC model, so infeasible P/B relay pairs can be rejected, and thus. This effective sub-algorithm can accelerate detecting feasible solutions.

4. The violations of P/B relay pairs detected by this technique do not disqualify the result. Instead, these violations mean that there are some pairs do not accept DPRS[12].

The proposed combinatorial BBO/DE algorithm is evaluated using numerical/-electromechanical, numerical/static, and numerical/electromechanical-static sets as main-1/main-2 DOCRs. Digital DOCRs are not considered because their hardware features are almost similar to numerical DOCRs.

[12] In this case, only a group of relays accepts DPRS, which means that the extreme case is infeasible.

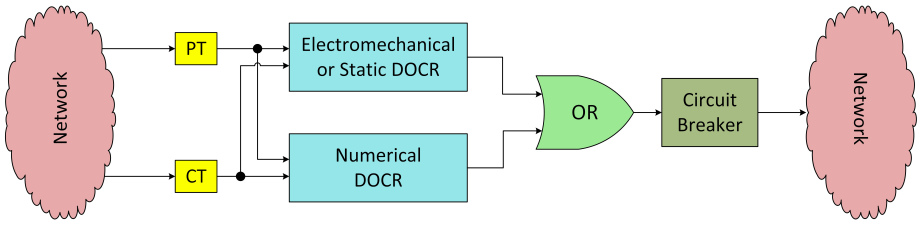


Figure 8.7: A Simple Protection Logic of the Double Primary Relay Strategy (PT: Potential Transformer — CT: Current Transformer)

8.3.1 Mathematical Formulation of the Double Primary Relay Coordination Problem

Figure 8.6 shows how the extreme case of DPRS can be illustrated on the IEEE 3-bus test system, given in [367], while assuming that a fault takes place on branch 1-2. In the classical ORC problem, R_1 and R_2 are allowed to act as primary relays to isolate that fault. If R_1 fails to operate, then R_5 operates as a backup relay. Similarly, R_4 acts as a backup relay for R_2. With DPRS, the operational philosophy is completely different because each circuit breaker is initiated by a trip signal that might come from its main-1 (a numerical) or main-2 (an electromechanical or a static) DOCR. If both main-1 and main-2 DOCRs fail to operate, then there are also, at least, two DOCRs (i.e., "**backup-1**" and "**backup-2**" relays) for each primary set. Therefore, ORC problems with DPRS are more difficult than ever; as seen in Table 8.7. To achieve this, a simple protection logic, shown in Figure 8.7, is considered. The same fault signal goes to both primary relays. The fastest device is the one that sends the first trip signal to the circuit breaker. If both main-1 and main-2 relays and/or their corresponding circuit breaker fail to operate, then again the same logic is applied to their backup-1 and backup-2 relays of each backup set. By referring to (4.2) and Table 8.7, the dimension n of such DPRS-based ORC problems can be determined by using the following updated formula[13]:

$$n = 4\varrho = 8ß \tag{8.12}$$

This special ORC problem can be mathematically modeled in any n-dimensional optimization algorithm as follows:

[13]It is used with the extreme case of DPRS.

8.3.1.1 Objective Function

Assume that a network contains ϱ main-1 and $\mathfrak{g}$ main-2 DOCRs, then the general objective function can be defined as follows:

$$\text{OBJ} = \min \sum_{k=1}^{l} \left[\sum_{i=1}^{\varrho} \left(w_{ia,k} T_{ia,k}^{\text{pr}} \right) + \sum_{j=1}^{\mathfrak{g}} \left(w_{jb,k} T_{jb,k}^{\text{pr}} \right) \right] \tag{8.13}$$

where $T_{ia,k}^{\text{pr}}$ and $T_{jb,k}^{\text{pr}}$ are respectively the operating times of the ith main-1 "a" and the jth main-2 "b" DOCRs when they act as primary protective relays for the kth fault location. The coefficients $w_{ia,k}$ and $w_{jb,k}$ are the kth fault location probability weights assigned to each relay, respectively.

If $\mathfrak{g} \neq \varrho$, then it means that some terminals of branches are not equipped with DPRS[14]. Based on this, the problem dimension is reduced, and hence becomes easier to solve. For the extreme case, both ϱ and $\mathfrak{g}$ are equal, so (8.13) becomes:

$$\text{OBJ} = \min \sum_{k=1}^{l} \left[\sum_{i=1}^{\varrho} \left(w_{ia,k} T_{ia,k}^{\text{pr}} + w_{ib,k} T_{ib,k}^{\text{pr}} \right) \right] \tag{8.14}$$

For simplicity, all the weights given in (8.14) are set equal to one [367]. Also, if only one common TCCC is used for all 2ϱ DOCRs, then the ORC problem can be solved using only 3ϕ near-end faults (i.e., $l = 1$) where the other less severe fault types and locations can also be achieved within that solution [40, 73]. Therefore, (8.14) is further simplified to:

$$\text{OBJ} = \min \sum_{i=1}^{\varrho} \left(T_{ia}^{\text{pr}} + T_{ib}^{\text{pr}} \right) \tag{8.15}$$

The operating time of the xth DOCR of the ith primary protective set can be computed using either the IEC/BS or the ANSI/IEEE standard model, given in [177, 179] and tabulated in Table 4.2, as follows:

$$T_{ix}^{\text{pr}} = TMS_{ix} \left[\frac{\beta}{\left(\frac{I_{R_{ix}}}{PS_{ix}} \right)^{\alpha} - 1} + \gamma \right] \quad , \begin{matrix} i = 1, \cdots, \varrho \\ x = a \text{ or } b \end{matrix} \tag{8.16}$$

where TMS_{ix} and PS_{ix} are the two independent variables of the xth DOCR of the ith primary protective set (x is either a or b), and $I_{R_{ix}}$ is the fault current seen by the

[14] i.e., the extreme case is not applicable here.

xth relay. Based on the most studies conducted in the literature, the IEC/BS IDMT-TCCC is used, so the coefficients $\{\alpha, \beta, \gamma\}$ are respectively set equal to $\{0.02, 0.14, 0\}$ for all 2ϱ DOCRs. By referring to Figure 8.7, the same current is seen by both main-1 and main-2 DOCRs through one common CT, so $I_{R_{ix}}$ is replaced by I_{R_i}. Therefore, (8.16) is modified for IDMT-based DOCRs as follows:

$$T_{ix}^{\text{pr}} = \frac{0.14\ TMS_{ix}}{\left(\frac{I_{R_i}}{PS_{ix}}\right)^{0.02} - 1}\ , \quad \begin{matrix} i = 1, \cdots, \varrho \\ x = a \text{ or } b \end{matrix} \tag{8.17}$$

Now, after defining the terms T_{ia}^{pr} and T_{ib}^{pr} of (8.15) by (8.17), this objective function should satisfy the following design constraints:

8.3.1.2 Selectivity Constraint

If T_{ja}^{bc} and T_{jb}^{bc} are the jth new (i.e., numerical) and old (i.e., electromechanical or static) backup relays assigned to T_{ia}^{pr} and T_{ib}^{pr} of the ith primary protective set, then the operating time of the fastest backup relay should be equal or longer than the slowest sum of the operating time and the coordination time interval of the xth primary relay:

$$\min\left[T_{ja}^{\text{bc}}, T_{jb}^{\text{bc}}\right] \geq \max\left[T_{ia}^{\text{pr}} + CTI_{ia}, T_{ib}^{\text{pr}} + CTI_{ib}\right] \tag{8.18}$$

where CTI_{ia} and CTI_{ib} are the coordination time intervals defined for the ath and bth DOCRs of the ith primary protective set to isolate their in-zone faults. Because different technologies are involved, so the typical value of CTI is between 0.2 and 0.5 s [52, 74]. Table 8.3 shows the practical range of different relay technologies.

The operating time of the xth backup relay of the jth protective set[15] can be calculated for an out-zone fault as follows:

$$T_{jx}^{\text{bc}} = \frac{0.14\ TMS_{jx}}{\left(\frac{I_{R_j}}{PS_{jx}}\right)^{0.02} - 1}\ , \quad \begin{matrix} j = 1, \cdots, \varrho \\ x = a \text{ or } b \end{matrix} \tag{8.19}$$

where I_{R_j} is the fault current that is supposed to be cleared by either the ith main-1 or main-2 relay and seen by both the jth backup-1 and backup-2 relays.

[15] Each ith protective set contains main-1 and main-2 relays, and each jth protective set contains backup-1 and backup-2 relays.

8.3.1.3 Inequality Constraints on Relay Operating Times

The operating time of each ixth relay is practically bounded between two limits:

$$T_{ix}^{\min} \leqslant T_{ix} \leqslant T_{ix}^{\max} \tag{8.20}$$

where $T_{ix}^{\min}$ is the minimum operating time that the ixth DOCR can achieve, and $T_{ix}^{\max}$ is the maximum operating time allowed to reach by the ixth DOCR without losing the system stability. From (8.16) and (8.17), T_{ix} is a dependent variable, so (8.20) must be divided into two functional inequality constraints as follows:

$$T_{ix}^{\min} - T_{ix} \leqslant 0 \tag{8.21}$$

$$T_{ix} - T_{ix}^{\max} \leqslant 0 \tag{8.22}$$

8.3.1.4 Side Constraints on Relay Time Multiplier Settings

Based on the technology of each relay, the side constraints of TMS can be mathematically expressed as follows:

$$TMS_{ix}^{\min} \leqslant TMS_{ix} \leqslant TMS_{ix}^{\max} \tag{8.23}$$

where $TMS_{ix}^{\min}$ and $TMS_{ix}^{\max}$ are respectively the minimum and maximum allowable limits of TMS of the ixth DOCR.

8.3.1.5 Side Constraints on Relay Plug Settings

The trip signal of the ixth DOCR is initiated once the current angle is correct and the current magnitude exceeds the given predetermined value. This value is defined as the severity level, which can be measured as follows:

$$PSM_{ix} = \frac{I_{R_i}}{PS_{ix}} \tag{8.24}$$

where PSM_{ix} is the plug setting multiplier of the ixth DOCR. If $PSM_{ix} < 1$, it means that no fault exists in the system.

This equation requires that the value of PS should be carefully selected to sense the lowest possible short-circuit current and, at the same time, the overload current is not isolated. Therefore, the proper PS of the ixth DOCR should be equal to or

less than two-thirds of the minimum fault current and equal to or higher than the overload current. This can be expressed as:

$$PS_{ix}^{\min} \leqslant PS_{ix} \leqslant PS_{ix}^{\max} \tag{8.25}$$

where $PS_{ix}^{\min}$ and $PS_{ix}^{\max}$ are respectively the minimum and maximum allowable limits of PS of the ixth DOCR.

To have a more realistic representation, these bounds are determined through some specific equations for each ixth DOCR; as described before in Subsection 4.2.4. However, most of the studies presented in the literature take PS as a predefined vector of discrete settings for all DOCRs.

8.3.2 Possible Configurations of Double Primary ORC Problems

In the preceding subsection, a full mathematical optimization model has been formulated. However, this special ORC problem could also be solved analytically if all main-2 relays are considered as local backup relays with dropping remote backup relays. This protection scheme is not practical and highly risky, but it is one of the following three possible protection schemes:

1. **Only remote backup relays are considered:**

   This is the classical ORC problem, which is also the one presented in the literature [40, 51, 70, 73, 245, 280, 296, 347, 360, 367, 368, 394, 395].

2. **Both local/remote backup relays are considered:**

   It is the core of this section. The protection design based on DPRS could be seen in some practical applications. It is introduced here with the extreme case where all ϱ circuit breakers are initiated either by main-1 numerical relays or main-2 old relays. But, most of (or even all) the time this strategy is partially applied to some selected circuit breakers.

3. **Only local backup relays are considered:**

   If all remote backup relays are dropped from the protection design, which is highly not recommended, then all main-2 (electromechanical, static, and digital

"hardware-based") relays will act as local backup relays, because the corresponding state-of-the-art numerical relays are always faster than the preceding relays.

If someone selects this protection design, then the exact optimal solution can be analytically obtained by setting both relays to their highest possible speed as follows:

$$T_i^{\text{primary}} = T_{ia}^{\min} \tag{8.26}$$

$$T_i^{\text{local backup}} = T_{ib}^{\min} \tag{8.27}$$

That is, there is no need to add CTI to discriminate between the primary and local backup relays. The reason behind this is that each one can operate without awaiting the other. This means that the selectivity constraint described in (8.18) is also dropped.

However, there are many practical problems associated with this odd protection design, such as:

- It is not practical to leave all the relays installed on the other branches without effectively utilizing them as a second line of defense. Even if they are all completely unutilized to clear out-zone faults, there is also some doubt about whether the selectivity criterion is satisfied or not. This claim can be clarified as follows:
- Short distribution lines have low impedance, so there is a chance that the other out-zone numerical relays act before the in-zone old relays. This could also happen if the in-zone local backup relays are electromechanical and the other out-zone remote backup relays are static or digital "hardware-based" because the last two old relay types are faster and thus they could isolate the faulty branch before the respective in-zone local backup relays; especially if $I_{R_j} \approx I_{R_i}$. This means that the protection design could be unselective, which is one of the major weaknesses of this protection scheme.
- For isolating faults, it is risky to depend on the same CTs and PTs for both protective relays.

- Having double relays on each circuit breaker is a very expensive design. Practically, this strategy is partially applied to some selected circuit breakers. It has been said before that the extreme case is investigated here because when it is solved the other less complicated problems, where DPRS is partially applied, can be easily solved as well.

Therefore, going with the third protection scheme (i.e., only local-backup relays) is not feasible.

8.3.3 Experimental Results and Discussion

To examine this realistic ORC model, the new hybrid BBO/DE algorithm is evaluated using the IEEE 6-bus, 15-bus, and 42-bus test systems. These three test systems are coded in MATLAB environment. For the first simple test system, the simulations are carried out using the following computing machine: ALIENWARE M14x Laptop, 64-bit Windows 10 OS, Intel Core i7-4700MQ CPU @ 2.4 GHz, and 16 GB RAM. For the larger test systems, a faster computing machine is used with the following specifications: ALIENWARE X51 Desktop, 64-bit Windows 10 OS, Intel Core i7-6700 CPU @ 3.4 GHz, and 8 GB RAM.

To conform with other studies reported in the literature, the following points are not considered[16] [40, 51, 70, 245, 280]:

1. Line thermal limit constraints.
2. Decaying DC, harmonics, and/or CTs saturation.
3. Transient network topology due to unequal operating times of both end relays of faulty lines.

Table 8.3 shows the predefined settings of TMS, PS and CTI used in this study. The maximum operating time $T^{\max}$ is set equal to 1 instead of 4. It is clear that the technology determines the lower and upper limits of relay variables as well as their step-size resolutions. Changing these resolutions can affect the feasible search space and making it more difficult to find, but the mechanism of the technique remains without any change. This means that the realistic model presented in this study can be applied to any DOCR-based ORC problem by simply updating the preceding

[16] They can be included in the model if someone wants to go further in the realization.

settings. For example, the numerical relay of this study is AREVA Micom/P12xy. However, the SIEMENS relay model 7SJ80 suggested in [100] can also be used by just updating the corresponding step-size resolutions.

8.3.3.1 Test Case I: 6-Bus System

Referring to the network shown in Figure 8.4, each circuit breaker can be initiated by a trip signal that might come from either main-1 or main-2 DOCR. If both relays fail to operate or if their circuit breaker does not open, then there is at least one set of backup-1 and backup-2 DOCRs located in the upstream ready to act after awaiting the assigned time delay. There are 28 DOCRs in the network as an extreme case of DPRS. If each relay has 2 variables (i.e., PS and TMS if only the IEC/BS IDMT-TCCC is used for all 2ϱ DOCRs), then the dimension of this ORC problem is 56. Besides, this problem has the following constraints:

- 20 inequality constraints for (8.18)
- 28 inequality constraints for (8.21)
- 28 inequality constraints for (8.22)
- 28 side constraints for (8.23)
- 28 side constraints for (8.25)

From the last row of Table 8.3, both TMS and PS are discrete where electromechanical DOCRs have very limited settings of PS. As stated in Table 8.7, the ELD problem has only 4 variables for this test system. The classical ORC problem has 28 variables with many constraints that need to be satisfied. Using our DPRS means dealing with 56 variables and the search space has different layers assigned to each type of DOCRs. This is the reason why the ORC problems are considered highly constrained nonlinear non-convex mixed-integer optimization problems; especially if the extreme case of DPRS is applied.

For this test system, the initialization parameters are: 200 generations, population size of 50, and 30 independent simulation runs. The clear duplication stage (of both BBO and BBO/DE) is completely disabled to avoid infinite loops due to searching within discrete variables of PS and TMS. The elitism stage is activated by recycling the best four individuals of one iteration into the other. The **step-size** (F) and

crossover rate (C_r) of DE are both equal to 0.5. Also, the binary static-exterior penalty function (BS-EPF) described in [122] is used here to handle all the preceding constraints. During modeling main-2 DOCRs, two scenarios are considered as follows:

Scenario 1: All Main-2 DOCRs Are Either Static or Electromechanical

The results are tabulated in Table 8.8. As can be seen from the table, two simulations are covered where the ϱ main-1 relays are always selected as numerical DOCRs. Each one of these two simulations is carried out by two optimization algorithms: the conventional BBO and the hybrid BBO/DE; both with FC. The constraints are handled by BS-EPF with a penalty factor of $r = 30$. The proposed hybrid BBO/DE with FC gives better results in both simulations. Disabling FC can save a remarkable CPU time, but the probability to get feasible solutions decreases steeply. For example, if electromechanical relays are used as main-2 DOCRs, then BBO without FC can save around 75% of the total CPU time spent when FC is enabled (BBO consumes 12.82 seconds and BBO+FC consumes 52.93 seconds). However, disabling FC generates two violations. Thus, to get feasible solutions without FC, it is required to increase the number of generations and/or population size plus modifying the penalty function and other settings, which leads to higher CPU time; especially with larger test systems. Returning back to Table 8.8, it is obvious that when static DOCRs are selected as main-2 protective devices the optimizer can find better solutions, while electromechanical DOCRs can provide better average and standard deviations. The reason is that the static DOCRs have a wider search space than that of the electromechanical DOCRs; as can be seen in the step-size resolutions of PS and TMS listed in Table 8.3. Therefore, the static DOCRs need more generations to converge to better settings. The fitness curves of the BBO and BBO/DE algorithms are shown in Figure 8.8 for both simulations (i.e., all main-2 relays are either static or electromechanical DOCRs).

Scenario 2: Main-2 DOCRs Are Both Static and Electromechanical

In this scenario, a mixture of static and electromechanical DOCRs are used as main-2 relays. They are selected based on a vector of uniformly distributed pseudorandom integers between 1s and 2s, which are generated by the MATLAB command

Table 8.8: Simulation Results of the IEEE 6-Bus Test System - Scenario 1

DOCR No.	Main-1: Numerical & Main-2: Electromechanical								Main-1: Numerical & Main-2: Static							
	BBO + FC				Hybrid BBO/DE + FC				BBO + FC				Hybrid BBO/DE + FC			
	Numerical		Electromechanical		Numerical		Electromechanical		Numerical		Static		Numerical		Static	
	PS	*TMS*	*PS*	*TMS*	*PS*	*TMS*	*PS*	*TMS*	*PS*	*TMS*	*PS*	*TMS*	*PS*	*TMS*	*PS*	*TMS*
R_1	1.47	0.522	2.0	0.50	0.63	0.605	0.5	0.70	1.34	0.520	2.3	0.425	1.54	0.540	0.9	0.550
R_2	2.49	0.190	2.5	0.15	1.03	0.383	0.6	0.40	0.80	0.518	1.9	0.325	0.97	0.431	1.0	0.375
R_3	1.43	0.277	1.0	0.40	1.71	0.399	0.8	0.55	1.24	0.447	2.2	0.675	0.73	0.589	1.0	0.475
R_4	1.00	0.357	0.8	0.30	2.27	0.167	0.6	0.35	1.96	0.128	1.0	0.250	2.21	0.091	1.2	0.225
R_5	2.18	0.216	1.0	0.25	0.72	0.317	1.0	0.25	1.12	0.283	1.1	0.350	0.63	0.361	1.7	0.275
R_6	1.91	0.486	2.0	0.25	0.96	0.299	2.0	0.25	1.29	0.239	2.5	0.125	2.25	0.148	2.1	0.150
R_7	1.48	0.570	2.5	0.35	2.17	0.272	1.5	0.40	2.41	0.396	1.7	0.375	1.81	0.292	1.4	0.450
R_8	1.38	0.261	0.8	0.40	0.69	0.251	2.0	0.15	2.00	0.233	1.1	0.200	0.81	0.299	0.6	0.225
R_9	1.22	0.346	0.8	0.40	0.70	0.527	1.5	0.25	1.41	0.377	0.7	0.525	2.31	0.231	1.1	0.375
R_{10}	0.89	0.440	1.0	0.45	0.83	0.542	0.5	0.55	1.58	0.370	2.2	0.250	1.02	0.644	1.8	0.325
R_{11}	2.10	0.385	1.5	0.55	1.02	0.498	2.0	0.30	0.76	0.668	1.8	0.300	2.06	0.290	1.4	0.325
R_{12}	0.87	0.783	2.5	0.60	1.10	0.693	0.6	0.75	0.52	0.985	1.6	0.550	1.78	0.601	2.5	0.500
R_{13}	1.09	0.186	2.5	0.10	2.32	0.187	1.0	0.25	1.16	0.248	1.3	0.150	1.04	0.469	2.1	0.050
R_{14}	2.14	0.343	2.5	0.25	1.40	0.316	0.6	0.45	2.40	0.252	1.0	0.425	0.96	0.423	1.1	0.375
OBJ (s)	20.1840				**19.0082**				20.0758				**18.9457**			
Mean (s)	22.4254				**21.6983**				22.6473				**22.4227**			
StDev (s)	1.4291				**1.3229**				1.4808				**1.4679**			

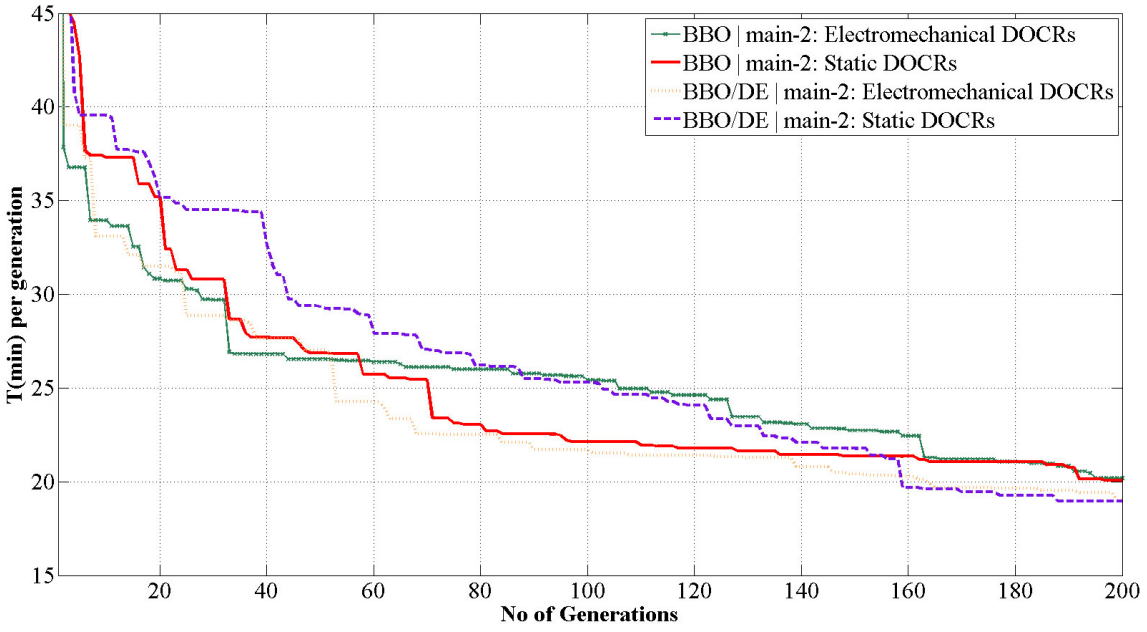


Figure 8.8: Curves of Fitness Functions of BBO and BBO/DE (Scenario 1 of the IEEE 6-Bus Test System: Main-2 DOCRs Are Either Static or Electromechanical)

`randi(`u`,`v`,`ϱ`)`; where u means the maximum discrete value that the vector elements can reach starting from 1, v means the required number of vectors to be generated, and ϱ is the total number of DOCRs assigned as main-2. These three parameters are set as: $u = 2$, $v = 4$, and $\varrho = 14$. Also, the preceding penalty factor is updated to $r = 35$ for this scenario. Table 8.9 shows the simulation results obtained for these 4 randomly generated vectors. All the solutions obtained for these 4 groups are feasible. The total number of installed relays, their types, models, TCCCs, and locations as well as the algorithm initialization parameters all play an important role in converging to better solutions. From Table 8.9, it can be clearly seen that the first group has the lowest standard deviation, while the second group has the lowest fitness and mean. Because the ORC problem is solved with the extreme case of DPRS where the total number of static relays are almost equal to that of electromechanical relays, so all the means are almost equal. The fitness curves obtained by the BBO/DE algorithm for these 4 randomly generated groups of main-2 relays are shown in Figure 8.9.

Table 8.9: Simulation Results of the IEEE 6-Bus Test System - Scenario 2

Primary	**Simulation of Group No.1**					**Simulation of Group No.2**					**Simulation of Group No.3**					**Simulation of Group No.4**				
DOCR	**Main-1**		**Main-2**			**Main-1**		**Main-2**			**Main-1**		**Main-2**			**Main-1**		**Main-2**		
No	*PS*	*TMS*	*PS*	*TMS*	**Type**	*PS*	*TMS*	*PS*	*TMS*	**Type**	*PS*	*TMS*	*PS*	*TMS*	**Type**	*PS*	*TMS*	*PS*	*TMS*	**Type**
R_{1x}	1.01	0.603	0.8	0.8	Elec[a]	1.65	0.528	1	0.8	Elec	0.96	0.638	0.6	0.725	Stat	0.83	0.589	0.5	0.75	Elec
R_{2x}	0.91	0.457	1	0.6	Elec	1.55	0.322	0.7	0.425	Stat	0.56	0.512	2.5	0.15	Stat	0.55	0.468	2.3	0.3	Stat
R_{3x}	2.35	0.479	2.3	0.375	Stat[b]	1.21	0.449	2.5	0.35	Elec	1.58	0.302	0.5	0.7	Elec	0.71	0.494	1	0.45	Elec
R_{4x}	2.43	0.154	1	0.2	Elec	0.94	0.189	0.8	0.325	Stat	2.02	0.152	1.5	0.15	Elec	2.21	0.239	2	0.125	Stat
R_{5x}	2.08	0.218	0.8	0.35	Elec	1.59	0.338	2.4	0.175	Stat	1.23	0.284	0.7	0.35	Stat	2.2	0.318	2	0.3	Elec
R_{6x}	2.36	0.235	1.6	0.325	Stat	2.03	0.123	2	0.2	Elec	1.62	0.379	1.5	0.275	Elec	2.08	0.179	0.9	0.4	Stat
R_{7x}	1.63	0.422	0.6	0.475	Stat	0.55	0.676	0.6	0.6	Stat	0.55	0.585	1.5	0.375	Stat	1.76	0.337	0.7	0.5	Stat
R_{8x}	2.12	0.136	0.6	0.35	Elec	0.97	0.223	0.6	0.25	Elec	1.52	0.229	1.8	0.25	Stat	2.22	0.165	0.7	0.35	Stat
R_{9x}	1.17	0.298	1.5	0.3	Elec	1.23	0.263	1	0.25	Elec	1.81	0.269	0.5	0.55	Elec	1.87	0.192	1	0.4	Elec
R_{10x}	0.83	0.459	2.5	0.25	Elec	1.59	0.253	1.7	0.325	Stat	0.54	0.707	0.9	0.45	Stat	1.52	0.251	0.5	0.45	Elec
R_{11x}	2.06	0.265	1.4	0.325	Stat	0.7	0.733	0.5	0.6	Elec	1.4	0.354	1	0.7	Elec	1.47	0.35	0.6	0.6	Elec
R_{12x}	1.13	0.851	1.5	0.55	Elec	2.39	0.601	0.6	0.85	Stat	0.77	0.675	0.8	0.675	Stat	2.28	0.509	0.5	0.8	Stat
R_{13x}	1.18	0.369	2	0.1	Elec	2.02	0.148	0.8	0.325	Stat	1.99	0.058	2.5	0.35	Elec	0.66	0.763	1.5	0.2	Elec
R_{14x}	1.72	0.237	0.6	0.475	Stat	0.55	0.461	1	0.35	Elec	1.04	0.365	0.9	0.5	Stat	1.67	0.294	2.5	0.325	Stat
OBJ (s)	19.7223					**19.4669**					20.3409					20.0743				
Mean (s)	22.4577					**22.0182**					22.6952					22.2022				
StDev (s)	**1.1870**					1.2929					1.7982					1.4053				

[a]Electromechanical (or Electromagnetic) Relay.
[b]Static (or Solid-State) Relay.

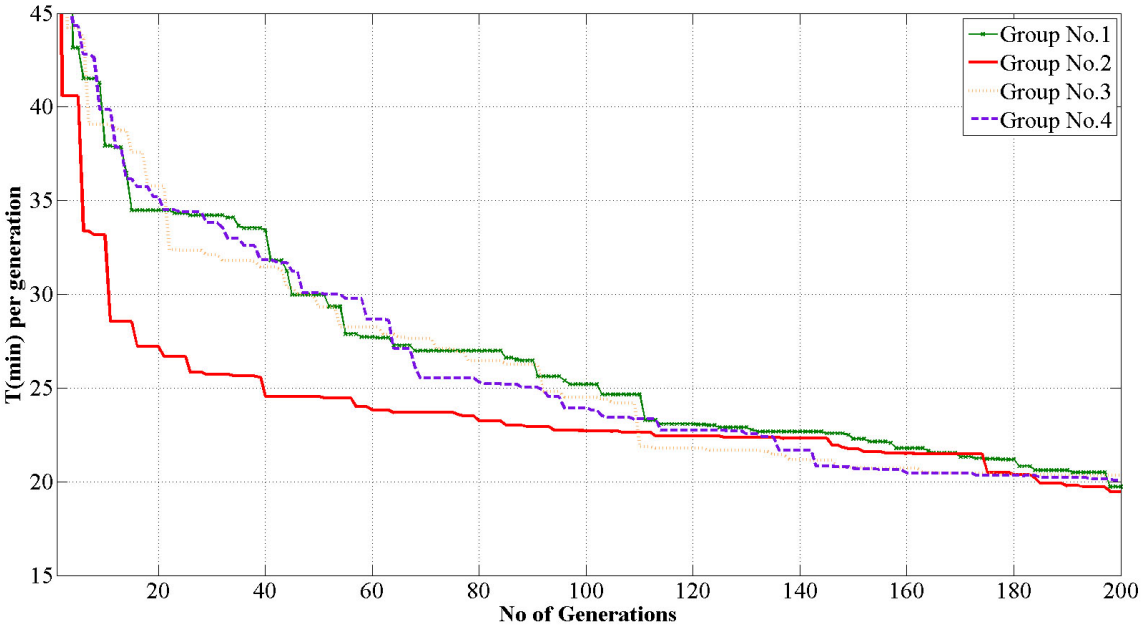


Figure 8.9: Curves of Fitness Functions of BBO/DE (Scenario 2 of the IEEE 6-Bus Test System: Main-2 DOCRs Are a Mixture of Static and Electromechanical)

8.3.3.2 Test Case II: 15-Bus Test System

This test system is shown in Figure 4.9 and all the necessary information is given in [25]. The classical ORC problem has 42 DOCRs. Thus, by applying the extreme case of DPRS, the dimension increases from being 84 to 168 variables with the following constraints:

- 82 inequality constraints for (8.18)
- 84 inequality constraints for (8.21)
- 84 inequality constraints for (8.22)
- 84 side constraints for (8.23)
- 84 side constraints for (8.25)

The same preceding initialization parameters are used with this ORC problem, except that the number of generations is set equal to 1000. The best solution obtained by the BBO/DE algorithm is tabulated in Table 8.10. For the selectivity constraints, 16 out of 82 are violated. However, as stated earlier, the classical model of this ORC

problem has been successfully solved in many papers with feasible solutions [40, 51, 100]. Thus, these violations have a different meaning. They could be called "**DPRS Incompatibility**" instead of violations, because the preceding 16 violations simply say: some P/B relay pairs do not accept DPRS. Therefore, the extreme case of DPRS can give a quick scan to the whole network to check whether all the P/B relay pairs are compatible with DPRS or not. If not, then: *Which pairs do not accept the DPRS strategy?* For the network shown in Figure 4.9, all the P/B relay pairs incompatible with DPRS are tabulated in Table 8.11.

8.3.3.3 Test Case III: 42-Bus Test System (IEEE Std. 399-1997)

As said in the last section, the test system shown in Figure 8.5 is one of the largest ORC test systems available in the literature [40]. The classical model of this test system has 97 DOCRs. Thus, by applying the extreme case of DPRS, the dimension increases from being 194 to 388 variables with:

- 114 inequality constraints for (8.18)
- 194 inequality constraints for (8.21)
- 194 inequality constraints for (8.22)
- 194 side constraints for (8.23)
- 194 side constraints for (8.25)

Using the same simulation parameters of the last test system, the best solution obtained by the BBO/DE algorithm is shown in Table 8.12. Based on (8.18), 48 out of 114 P/B relay pairs are incompatible with DPRS. It has to be said that these 48 incompatible pairs could be reduced if the population size and/or generations are increased because 388 variables definitely require more exploration and exploitation.

8.4 Is It Enough to Just Rely on Near-End, Middle, and Far-End Points to get Feasible Relay Coordination?

Since the end of the eighties of the last century, ORC becomes one of the hot topics covered in the literature [40, 51, 367]. Many analytical and numerical techniques have been presented as effective tools to solve this highly constrained nonlinear non-convex

Table 8.10: Simulation Results of the IEEE 15-Bus Test System

Primary	**Main-1**		**Main-2**			**Primary**	**Main-1**		**Main-2**			**Primary**	**Main-1**		**Main-2**		
DOCR	PS	TMS	PS	TMS	**Type**	**DOCR**	PS	TMS	PS	TMS	**Type**	**DOCR**	PS	TMS	PS	TMS	**Type**
R_{1x}	1.27	1.414	0.7	1.25	Stat	R_{15x}	1.97	1.112	1.6	0.85	Stat	R_{29x}	2.11	0.847	1.4	1.05	Stat
R_{2x}	1.32	1.076	1.3	1.05	Stat	R_{16x}	1.17	0.451	0.6	0.2	Stat	R_{30x}	1.35	1.362	1	1.025	Stat
R_{3x}	1.99	0.99	1.6	1.225	Stat	R_{17x}	1.41	0.997	2.3	0.575	Stat	R_{31x}	1.61	0.889	2.3	0.75	Stat
R_{4x}	2.23	0.911	2	0.7	Elec	R_{18x}	0.77	1.013	1.5	1.05	Elec	R_{32x}	1.85	0.375	1.2	0.425	Stat
R_{5x}	1.53	1.421	2.5	0.8	Elec	R_{19x}	0.81	1.138	2	0.75	Stat	R_{33x}	1.21	1.072	1	0.95	Stat
R_{6x}	1.59	0.677	0.8	0.35	Elec	R_{20x}	1.32	1.076	1	1.05	Elec	R_{34x}	0.92	1.415	2.5	1.075	Stat
R_{7x}	1.96	1.169	1.4	1.175	Stat	R_{21x}	1.74	0.832	2	1.05	Stat	R_{35x}	2.38	1.01	2	0.95	Elec
R_{8x}	0.58	1.101	1.3	0.375	Stat	R_{22x}	0.63	1.26	1	1.1	Elec	R_{36x}	1.01	0.279	2	0.2	Elec
R_{9x}	1.2	1.41	2.5	0.8	Elec	R_{23x}	0.59	1.426	1.1	1.075	Stat	R_{37x}	1.07	1.385	2.5	0.75	Elec
R_{10x}	1.16	1.366	1.3	1.075	Stat	R_{24x}	1.6	1.051	2	1	Elec	R_{38x}	2.09	0.32	0.6	1	Elec
R_{11x}	1.7	1.399	1.8	0.825	Stat	R_{25x}	2.43	0.813	2.4	1.075	Stat	R_{39x}	2.28	0.888	2	0.9	Elec
R_{12x}	0.78	1.128	1.5	0.55	Elec	R_{26x}	2.23	0.096	1.7	0.375	Stat	R_{40x}	0.86	0.558	0.8	1.1	Elec
R_{13x}	1.73	1.23	0.8	1.125	Stat	R_{27x}	0.91	1.335	1.4	1.025	Stat	R_{41x}	1.75	0.771	1.4	0.075	Stat
R_{14x}	2.35	1.06	2.5	0.7	Stat	R_{28x}	2.29	1.279	2	1	Elec	R_{42x}	1.27	1.257	1.6	0.95	Stat
OBJ (s)	200.9725828																

Table 8.11: Incompatible P/B Pairs with DPRS in the IEEE 15-Bus System

P/B Relay Pair		**P/B Relay Pair**		**P/B Relay Pair**		**P/B Relay Pair**	
R_{pr}	R_{bc}	R_{pr}	R_{bc}	R_{pr}	R_{bc}	R_{pr}	R_{bc}
1	6	6	8	16	26	28	32
2	16	9	8	17	26	30	32
3	16	11	7	26	36	38	40
4	12	11	20	27	36	40	41

mixed-integer optimization problem. However, these optimizers are built based on a hypothesis that feasible optimal solutions can be guaranteed if the discrimination margin between the operating times of each primary and backup (P/B) relay pair is satisfied at some 3ϕ fault points specified on each branch. This section tries to study the standard design criteria, used during solving ORC problems, to answer the main question raised in the title of this section.

8.4.1 Infeasibility of Selectivity Constraints

In Chapter 4, it has been seen that the selectivity constraint can be satisfied by giving enough time to each primary relay before initiating its backup relay(s). This checking process is done at some user-defined 3ϕ faults. The standard fault locations are shown in Figure 4.2 and listed in Table 4.1. It seems that everything works smoothly without any practical problem. That is, if the discrimination margin between each P/B relay pair is satisfied at the highest short-circuit current (i.e., 3ϕ fault in most cases [123]), then that margin will definitely be larger for lesser currents as depicted in Figure 8.10.

However, it has been found that some backup relays see non-monotonic changes in short-circuit currents when the bolted point is gradually shifted from the near-end point to the far-end point. This phenomenon reveals a very important fact that the existing techniques used in solving ORC problems do not assure the feasibility of their solutions. This section covers this phenomenon with some numerical results.

8.4.2 Numerical Experiment and Discussion

To prove our claim, let's create different faults between bus 1 and bus 6 of the IEEE 8-bus test system shown in Figure 9.68. The detailed information about this test system is given in [25]. For that faulty branch, the primary DOCR mounted on bus 1

Table 8.12: Simulation Results of the 42-Bus Test System (IEEE Std. 399-1997)

Primary	**Main-1**		**Main-2**			**Primary**	**Main-1**		**Main-2**			**Primary**	**Main-1**		**Main-2**			**Primary**	**Main-1**		**Main-2**		
DOCR	*PS*	*TMS*	*PS*	*TMS*	**Type**	**DOCR**	*PS*	*TMS*	*PS*	*TMS*	**Type**	**DOCR**	*PS*	*TMS*	*PS*	*TMS*	**Type**	**DOCR**	*PS*	*TMS*	*PS*	*TMS*	**Type**
R_{1x}	2	1.19	1.5	0.95	Elec	R_{26x}	1.31	0.272	1.4	0.225	Stat	R_{51x}	1.71	0.878	2.1	0.85	Stat	R_{76x}	2.27	0.495	2	0.3	Elec
R_{2x}	2.46	1.283	2.5	0.85	Elec	R_{27x}	1.09	1.215	1	1.05	Elec	R_{52x}	1.95	1.265	1.4	1.275	Stat	R_{77x}	1.65	0.257	0.8	0.925	Stat
R_{3x}	0.94	0.95	1	1	Elec	R_{28x}	1.41	0.807	1	0.75	Elec	R_{53x}	2.28	0.792	1.5	0.9	Stat	R_{78x}	2.25	0.176	2.5	0.1	Elec
R_{4x}	2.27	0.875	2.4	0.7	Stat	R_{29x}	2.46	1.207	2.5	1.225	Stat	R_{54x}	2.25	1.248	1.7	1.1	Stat	R_{79x}	0.98	0.26	1	0.3	Elec
R_{5x}	1.59	1.134	0.8	1.15	Stat	R_{30x}	0.81	1.167	1.4	0.725	Stat	R_{55x}	1.3	1.076	2.4	1.225	Stat	R_{80x}	2.31	0.598	1.7	0.35	Stat
R_{6x}	2.23	1.185	2.4	0.4	Stat	R_{31x}	1.35	1.035	2.2	0.875	Stat	R_{56x}	1.79	0.424	1.5	0.5	Elec	R_{81x}	1.53	0.873	2.1	0.825	Stat
R_{7x}	1.47	1.358	2	0.7	Elec	R_{32x}	2.16	0.242	2.5	0.7	Elec	R_{57x}	1.94	0.917	2	0.9	Elec	R_{82x}	1.41	0.123	1.4	0.575	Stat
R_{8x}	2.02	1.462	1.5	1	Elec	R_{33x}	1.39	1.258	1.7	1.2	Stat	R_{58x}	0.54	1.082	1.5	1.125	Stat	R_{83x}	1.45	0.821	0.5	0.75	Elec
R_{9x}	1.6	0.197	1	0.15	Elec	R_{34x}	1.6	1.041	1.5	0.9	Elec	R_{59x}	0.62	1.452	2.5	0.85	Elec	R_{84x}	1.61	0.7	2	0.7	Elec
R_{10x}	0.52	0.234	2	0.125	Stat	R_{35x}	1.61	1.124	1.5	0.95	Elec	R_{60x}	1.8	0.836	0.7	0.95	Stat	R_{85x}	0.77	0.383	0.9	1	Stat
R_{11x}	1.94	1.474	2	1.175	Stat	R_{36x}	1.38	0.885	2.4	1.275	Stat	R_{61x}	0.52	0.735	2.3	1.075	Stat	R_{86x}	2.2	0.627	2	0.6	Elec
R_{12x}	1.44	0.868	1.5	0.2	Elec	R_{37x}	2.04	0.655	1.5	0.725	Stat	R_{62x}	2.34	0.229	1	0.5	Elec	R_{87x}	2.07	0.072	1.5	0.25	Stat
R_{13x}	2.3	0.853	0.5	1	Elec	R_{38x}	0.94	1.038	1.3	0.95	Stat	R_{63x}	1.7	1.252	1.4	0.975	Stat	R_{88x}	1.99	0.268	1	0.225	Stat
R_{14x}	1.17	1.442	2	1.1	Elec	R_{39x}	0.79	0.878	1.5	0.7	Elec	R_{64x}	2.42	1.068	1	1.1	Elec	R_{89x}	1.79	0.409	0.5	0.075	Stat
R_{15x}	1.85	0.299	0.5	0.15	Elec	R_{40x}	2.19	1.325	1	1.1	Elec	R_{65x}	1.97	0.847	1.9	0.65	Stat	R_{90x}	0.97	0.121	1.5	0.2	Elec
R_{16x}	1.01	1.263	2.5	0.9	Elec	R_{41x}	1.87	0.495	2	0.7	Elec	R_{66x}	1.75	1.17	1.9	1.05	Stat	R_{91x}	0.84	0.309	1.1	0.075	Stat
R_{17x}	1.37	1.217	2.5	1	Elec	R_{42x}	1.57	0.665	1.5	0.2	Elec	R_{67x}	0.91	0.388	2.1	0.325	Stat	R_{92x}	0.56	0.735	2.5	0.6	Elec
R_{18x}	0.59	1.212	2.5	0.85	Stat	R_{43x}	1.16	0.299	0.8	0.225	Stat	R_{68x}	1.59	0.678	2.5	0.6	Elec	R_{93x}	2.24	0.039	2	0.5	Elec
R_{19x}	0.89	1.144	1.1	0.9	Stat	R_{44x}	2.04	0.691	1.5	0.65	Elec	R_{69x}	0.63	0.855	1	0.8	Elec	R_{94x}	1.41	0.05	2.5	0.4	Elec
R_{20x}	1.55	1.15	1.2	1.175	Stat	R_{45x}	1.36	1.45	2	0.9	Elec	R_{70x}	1.88	0.624	1.1	1.25	Stat	R_{95x}	1.67	1.004	2.5	0.6	Elec
R_{21x}	1.59	1.242	2.5	1.1	Elec	R_{46x}	2.47	0.867	2	0.65	Elec	R_{71x}	0.83	0.627	2	0.75	Elec	R_{96x}	1.54	0.911	2	0.3	Stat
R_{22x}	2.32	1.303	1	1	Elec	R_{47x}	2.48	0.445	2	0.55	Elec	R_{72x}	1.42	1.03	0.9	1.275	Stat	R_{97x}	0.93	0.157	2.3	0.25	Stat
R_{23x}	2.21	0.635	2.5	0.75	Elec	R_{48x}	0.82	1.061	1.5	0.85	Elec	R_{73x}	0.73	0.78	1	0.225	Stat	—	-	-	-	-	-
R_{24x}	1.55	1.29	2	1.075	Stat	R_{49x}	2.26	0.778	0.8	0.975	Stat	R_{74x}	1.2	0.048	0.5	0.2	Elec	—	-	-	-	-	-
R_{25x}	2.3	1.13	0.9	0.925	Stat	R_{50x}	0.95	1.186	1.5	0.55	Elec	R_{75x}	0.61	0.904	1.8	0.15	Stat	—	-	-	-	-	-
OBJ (s)	226.7734744																						

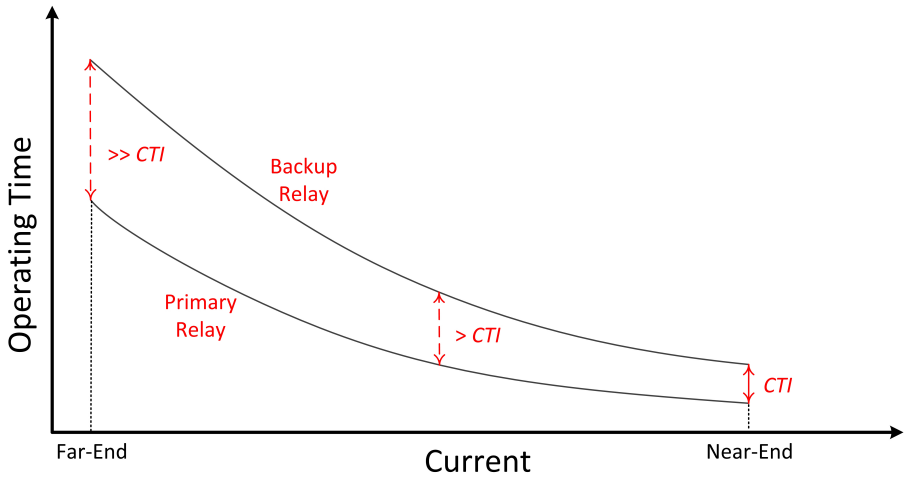


Figure 8.10: Discrimination Margin Between Primary and Backup Relays

Table 8.13: Optimal Settings of Some Relays of the IEEE 8-Bus Test System - Near-End 3ϕ Faults with the IEC Standard Inverse TCCC

setting	R_1	R_5	R_7	R_9	R_{13}	R_{14}
TMS	0.1040	0.1072	0.1	0.1086	0.1003	0.1
PS	0.8	1.5	2.5	2.0	0.6	2.0

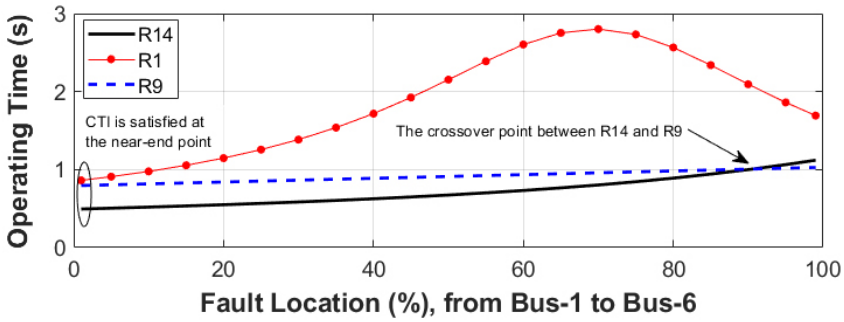
is R_{14} and its backup DOCRs are R_1 and R_9. Similarly, R_7 is the primary DOCR of bus 6 and its backup DOCRs are R_5 and R_{13}. Now, let's use the BBO-LP algorithm, described in Chapter 2 and Chapter 4, to find the optimal settings of these two P/B relay pairs. If the IEC standard inverse[17] TCCC is selected, then (4.4) should be used with ($\alpha = 0.02$) and ($\beta = 0.14$). For this test system, $CTI = 0.3$ s [40]. Table 8.13 shows the optimal TMS and PS of these six relays. Table 8.14 shows the 3ϕ short-circuit currents fed to these six DOCRs and the operating times measured from them.

The operating times of both P/B relay pairs, at different fault locations, are depicted in Figure 8.11. From both plots, it is very clear that the selectivity constraint is satisfied at the near-end fault of each primary relay (i.e., 1% for R_{14} and 99% for R_7). It happens because this test system is built based on the minimum design criterion; covered in Table 4.1 [34, 40, 51, 280]. If some faults are created on different

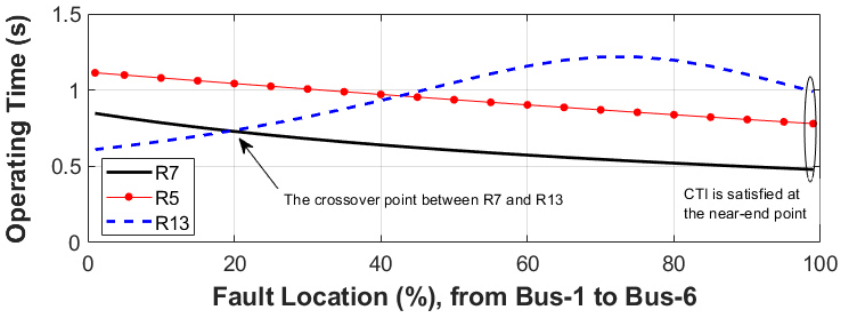
[17] i.e., the IDMT curve.

Table 8.14: The Near-End 3ϕ Fault Currents and Operating Times of the P/B Relay Pairs of Branch 1-6 of the IEEE 8-Bus Test System

Fault	**Primary Relays**				**Backup Relays**							
Location	R_7		R_{14}		R_1		R_5		R_9		R_{13}	
k (%)	$I_{7,k}$ (A)	$T_{7,k}$ (s)	$I_{14,k}$ (A)	$T_{14,k}$ (s)	$I_{1,k}$ (A)	$T_{1,k}$ (s)	$I_{5,k}$ (A)	$T_{5,k}$ (s)	$I_{9,k}$ (A)	$T_{9,k}$ (s)	$I_{13,k}$ (A)	$T_{13,k}$ (s)
1	907.51	0.8475	1291.33	0.4948	443.60	0.8621	703.08	1.1136	824.74	0.7954	449.43	0.6099
5	933.78	0.8187	1255.12	0.5052	424.89	0.9092	709.50	1.0986	815.64	0.8049	431.67	0.6325
10	967.10	0.7859	1210.68	0.5191	402.41	0.9766	717.80	1.0799	804.68	0.8168	410.28	0.6636
15	1001.01	0.7561	1167.45	0.5339	381.00	1.0550	726.42	1.0614	794.19	0.8287	389.85	0.6980
20	1035.53	0.7289	1125.37	0.5497	360.72	1.1472	735.39	1.0431	784.14	0.8406	370.41	0.7361
25	1070.66	0.7040	1084.38	0.5666	341.65	1.2560	744.70	1.0249	774.51	0.8525	352.04	0.7784
30	1106.43	0.6810	1044.45	0.5848	323.89	1.3849	754.37	1.0069	765.26	0.8643	334.81	0.8251
35	1142.86	0.6598	1005.51	0.6044	307.57	1.5377	764.41	0.9891	756.37	0.8762	318.84	0.8763
40	1179.99	0.6401	967.56	0.6257	292.85	1.7172	774.84	0.9714	747.84	0.8880	304.26	0.9315
45	1217.84	0.6217	930.55	0.6488	279.91	1.9239	785.68	0.9540	739.62	0.8998	291.21	0.9900
50	1256.47	0.6046	894.47	0.6740	268.94	2.1530	796.93	0.9368	731.71	0.9116	279.88	1.0495
55	1295.90	0.5885	859.33	0.7016	260.16	2.3890	808.62	0.9198	724.09	0.9234	270.45	1.1070
60	1336.20	0.5734	825.11	0.7320	253.76	2.6030	820.77	0.9030	716.74	0.9351	263.13	1.1577
65	1377.41	0.5591	791.84	0.7656	249.94	2.7532	833.39	0.8865	709.65	0.9469	258.09	1.1963
70	1419.59	0.5457	759.56	0.8028	248.80	2.8019	846.51	0.8702	702.79	0.9587	255.50	1.2174
75	1462.80	0.5329	728.30	0.8442	250.43	2.7328	860.15	0.8541	696.17	0.9705	255.47	1.2177
80	1507.12	0.5207	698.14	0.8903	254.80	2.5653	874.35	0.8382	689.75	0.9823	258.07	1.1964
85	1552.64	0.5092	669.15	0.9419	261.85	2.3390	889.13	0.8225	683.54	0.9940	263.28	1.1566
90	1599.28	0.4981	641.52	0.9995	271.48	2.0944	904.45	0.8071	677.49	1.0059	271.10	1.1027
95	1647.32	0.4876	615.30	1.0637	283.47	1.8613	920.44	0.7919	671.64	1.0178	281.36	1.0412
99	1686.75	0.4795	595.48	1.1201	294.66	1.6924	933.72	0.7799	667.07	1.0273	291.25	0.9898



(a) Operating Times of the Primary Relay R_{14} and the Backup Relays R_1 and R_9



(b) Operating Times of the Primary Relay R_7 and the Backup Relays R_5 and R_{13}

Figure 8.11: Infeasibility of Discrimination Margins of the P/B Relay Pairs at Some Fault Locations Between Bus 1 and Bus 6

locations of that line, then there is no guarantee that the selectivity constraint will be satisfied. This can be observed in the preceding plots. For instance, the backup relay R_9 operates before the primary relay R_{14} at the far-end point where the fault is 99% away from bus 1 and 1% away from bus 6. Similarly, the backup relay R_{13} operates before R_7 at the far-end point where the fault is 1% away from bus 1 and 99% away from bus 6.

This numerical example can be generalized for other coordination criteria listed in Table 4.1. Even if all the three standard points (i.e., near-end, middle, and far-end points) are considered, the suspicion still exists. This can be easily proved by looking at the crossover between each P/B relay pair shown in Figure 8.11. For R_{14}, the crossover happens with R_9 at about 10% away from bus 6. For R_7, the crossover happens with R_{13} at about 20% away from bus 1. It is obvious that there are other

doubtful points that need to be checked in addition to the preceding three standard points. Thus, satisfying any one, or even multiple, of coordination criteria listed in Table 4.1 could be insufficient to ensure the feasibility of ORC problems.

The more complex, but safer, option is to use a gradient of fault points along each line to satisfy the selectivity constraint. That is, $(k = 1, 2, \cdots, l)$, where $(l >> 2)$ and covers all the spots of each faulty line. This means that the simplification applied to (8.3), regarding l, is not valid anymore. Although this addition will make ORC problems harder than ever, the solutions obtained by this approach are more practical and thus the system reliability can be enhanced. The problem dimension and complexity will dramatically increase if multiple TCCCs are involved [40]. Also, if the double primary relay strategy (DPRS), presented in the last section, is applied, or/and if other protective relays and fuses are involved, then the new optimization model of ORC problems will need a very powerful algorithm to deal with this dilemma. Thus, artificial intelligence (AI) could be a must.

Chapter 9

Innovative Ideas for Power System Studies

This chapter presents part of our innovative ideas that can be implemented in modern electric power systems. Each one of these studies, or sections, is independent and not related to others.

9.1 Power Flow Quantities Estimation Using ANNs

A brief introduction to artificial neural networks (ANNs) has been covered in Chapter 5. It is a vast field where many techniques and applications are frequently presented from time to time. Modern electric power systems highly depend on many **artificial computation** techniques, including ANNs, SVMs, **fuzzy systems** (**FS**), and optimization algorithms. Some of ANN-/SVM-based applications have been covered in Chapter 5, which occupy just a very small portion of AI capabilities. For example, some recent studies suggest to use ANNs to solve power flow (PF) problems directly without referring to any mathematical equations. Such studies are reported in [183, 212, 242, 265, 274]. However, there is one missing piece to make them very practical! Solving PF problems by ANNs does not help that much since modern highly efficient software[1] integrated with highly advanced computing machines can effectively solve PF problems even with tens of thousands of buses. What makes a difference here is something that can solve PF problems from other perspectives! For example, estimating the active and reactive power losses (P_L and Q_L) of a network without knowing the power supplied from the slack bus. This has a significant impact on the economic load dispatch (ELD) and unit commitment (UC) studies where the stiff active power balance constraint given in (3.16) can be directly solved. Also, Kron's loss formula given in (3.18) is not accurate, and it is built based on many hard assumptions [123]; as will be seen later. Thus, building an AI-based technique

[1]Such as SIEMENS PSS/E, ETAP, SKM, EasyPower, DigSILENT PowerFactory, EUROSTAG, ASPEN, CYME, Neplan, PowerWorld Simulator, SIMPOW and Paladin DesignBase.

to find the power losses, directly and accurately without referring to the slack bus, is a great add-on. This section reveals the mechanism of a classical AI-based PF estimator, and then it dives into our proposed innovative techniques that can accurately and precisely estimate the power losses, magnitudes, and directions. Some numerical experiments are conducted on the WSCC 9-bus test system shown in Figure 6.26 to validate their performance.

9.1.1 Active and Reactive Power Flow Estimation

The term "power flow (PF)" as in some references [59, 149, 155, 184, 185, 192, 310, 326, 349, 357, 378, 386], or "load flow (LF)" as in others [57, 58, 108, 109, 123, 184, 186, 211, 231, 352, 378], is frequently used as a subject of one of the most important tools in electric power systems engineering. Here are some nice sentences taken from textbook authors; Saadat in page 189 of [326]: "*Power flow studies are the backbone of power system analysis and design*", Bergen in page 150 of [59]: "*It is an integral part of studies in system planning and operation and is, in fact, the most common of power system computer calculations*", and finally by Grainger-Stevenson in page 329 of [155]: "*Power flow studies are of great importance in planning and designing the future expansion of power systems as well as in determining the best operation of existing systems*". This repeated meaning gives a solid conclusion that all electric power systems need some sorts of PF studies to have the ability to measure, monitor, analyze, estimate, predict, and control many variables and parameters to maintain these systems secure and at their optimal conditions [109]. It is clearly highlighted by El-Hawary in page 319 of [125]: "*An ubiquitous EMS application software is the power flow program, which solves for network state given specified conditions throughout the system.*"

Therefore, to understand the importance of power flow studies, the following questions should be raised first: *What does "power flow" mean? What are the techniques used to solve it? What are the pros and cons of each one of these techniques?*

It is well known that modern electric power systems are highly interconnected. These systems are represented by branches and nodes with some **injected sources** and **consumption points**. The injected sources represent the generating units (such

as nuclear, thermal[2], tidal, hydro, wind farms, solar stations, etc) connected to the grid. The branches are called powerlines (transmission, sub-transmission, and distribution lines), which are connected between each other through some nodes called busbars. The consumption points are defined as loads; they could be sub-transmission customers (26 kV to 69 kV), primary customers (13 kV to 4 kV), secondary customers (120 V / 60 Hz "**American Standard**" or 240 V / 50 Hz "**European Standard**"), or even power station auxiliary plants (air and gas compressors, lube oil cooling systems, lightings, etc). Batteries, **ultracapacitors**[3], and **flywheels** are special power components. These bi-directional elements act as loads when there is enough power flowing in the grid, and act as power sources in case there is a shortage in the production of electricity.

After modeling these electric power components from their physical structures to some mathematical expressions, they can then be represented as a power electric circuit. Although network equations can be formulated in different forms, the most common one used for power system analysis is called the **node voltage method** [108, 125]. If the given network is formulated in a **nodal admittance form**, then by the node currents the network can be expressed as **linear algebraic equations**. But practically, electric systems are represented by power values instead of currents, which results in a set of **algebraic nonlinear equations** called "**power flow equations**". These equations can be solved by **iterative techniques** [175, 310, 326]. That is, solving power flow equations leads to knowing the voltage magnitude $|V|$ and its phase angle δ at each busbar, and the real power P and reactive power Q flowing through each branch. Moreover, from these essential data, much other information can be easily calculated in some sub-algorithms embedded within EMS, which can be used later for many other studies [155].

Based on this brief introduction, it can be said that the power flow analysis is carried out to ensure that the following requirements are satisfied [147, 149, 310]:

1. Each bus voltage magnitude is close to its rated value:

$$|V_i| \approx V_i^{\text{rated}} \tag{9.1}$$

2. The total power generation P_G should meet the total consumed power (i.e., the summation of power demand P_D and power losses P_L) as follows:

[2]Some types are GTs, STs, gas generator, diesel generator, etc.
[3]Also called **supercapacitors** (**SCs**).

Table 9.1: Bus Types and Their Known/Unknown PF Variables

Busbar Type	**Known**	**Unknown**
Swing bus[a]	$\|V_i\|$ and δ_i	P_i and Q_i
Generator bus[b]	P_i and $\|V_i\|$	Q_i and δ_i
Load bus[c]	P_i and Q_i	$\|V_i\|$ and δ_i,

[a] Also called **slack** or **reference bus**
[b] Also called **voltage-controlled** or **PV bus**
[c] Also called **PQ bus**

$$P_G = P_D + P_L \tag{9.2}$$

3. All the generators should not exceed the specified real and reactive power limits:

$$P_G^{\min} \leq \quad P_G \quad \leq P_G^{\max} \tag{9.3}$$

$$Q_G^{\min} \leq \quad Q_G \quad \leq Q_G^{\max} \tag{9.4}$$

4. Lines and transformers are not overloaded:

$$I \leq I_L^{max} \times \text{OLF} \quad , \quad \text{where} \quad \text{OLF} = 1.25 \text{ to } 1.5 \tag{9.5}$$

Thus, the independent variables here are: the voltage magnitude "$|V|$", the angle "δ", the real power "P" and the reactive power "Q" [149, 175]. In PF, each bus has two **known** (**specified**) **variables** and two **unknown** (**unspecified**) **variables**. The busbar types depend upon the known variables and can be summarized in Table 9.1.

Nowadays, there are many PF techniques proposed in the literature. Some of these are reported in [131,207,209,231,255,323,326,357,363]. In terms of accuracy, the worst method is the **DC load flow**, which becomes the best method in terms of processing time. This method is just used in some special applications, like contingency analysis and quick optimal pricing calculations. Also, it is good for getting a general figure or initial point to estimate some online scenarios where the processing time is the most critical factor and at the same time some decimal places of tolerance can be sacrificed. It is important to say that, for accurate and precise calculations, this method is totally discarded [99]. If **Tellegen's theorem** is applied here instead, as in [109, 131], then it might provide good results with very limited usage of memory. The **Gauss**

methods have simple calculation steps, which make them easy to program. Also, they require less memory and processing time. However, their sensitivity can be affected by the selection of the slack bus. Also, as the network size increases the algorithms utilize more iterations, which is the case faced with real power networks. This phenomenon creates a bold usage limitation [175]. The most popular one is the Newton-Raphson (NR) method. Some of its main advantages are its high accuracy and quick convergence rate without depending on the network size or the slack bus selection. However, this technique is very hard to implement in some applications because it consumes a large amount of CPU time and data storage [175]; especially with radial systems where most of the Jacobian matrix elements are zero. This means that if the lower and upper off-diagonal non-zero elements σ are equal or three times that of the diagonal elements (i.e., $\sigma = 1 \rightarrow 3$), then with a 5000×5000 matrix the following useless memory can be faced if it is implemented in a practical application:

- The total matrix elements: $m^2 = 5000^2 = 25,000,000$
- The total non-zero elements: $m + 2\sigma m = (1 + 2\sigma)\, m = 15,000$ to $35,000$

Imagine! There are 24,965,000 elements, which are saved just as zeros!! This logical astonishment can be clearly seen during coding NR in any specialized numerical programming language.

As a summary, PF analysis can be translated as a "frozen" picture of one moment, condition, or scenario of a dynamic interconnected electric power system [184].

This subsection tries to solve the entire PF problem by using ANNs that accept the same input and output variables of classical PF solvers. To validate the working principle of this scheme, two numerical experiments are covered here. One is conducted based on the WSCC 9-bus test system. The other one is a virtual very large test system, which is just used to test the processing speed of ANNs.

9.1.1.1 Stage No.1: Creating ANNs Dataset

To make ANNs applicable in any numerical problem, it is important to feed these networks with a matrix of input values (predictors or independent variables) and a matrix of output values (responses, **targets**, or dependent variables). Thus, to implement ANNs to solve PF problems, the preceding strategy should be applied

here. Many studies have been reported in the literature, which use ANNs to solve many highly complicated power system problems. For this particular problem, it is important to say that some wise steps should be considered during creating the dataset. Some of these essential steps are:

- Different settings of generating units and loads should be provided through a random process.
- Reasonable predictors should be added in the **input matrix**:
  - $|V|$ and δ of the slack bus.
  - P and $|V|$ of the generator buses.
  - P and Q of the load buses.
- The status of all the branches should be considered too.
- The **output matrix**, which contains the actual responses or targets, should be produced by using some highly accurate PF solvers, such as NR.

Summing all these steps together in a systematic process will result in constructing a general flowchart similar to that shown in Figure 9.1.

In addition, highly advanced ANN-based power flow (NNPF) solvers can be created by considering the variations on system parameters due to the dynamic disturbances on the system frequency and the surrounding weather conditions [12, 13]. Furthermore, real-time readings measured by EMS can be directly utilized to have a realized NNPF solver that can strongly stand against the uncertainty of the system.

9.1.1.2 Stage No.2: Learning Process of ANNs

Although the input/output dataset can be created by many options, this study uses the same configuration implemented in classical PF solvers. That is, to estimate PF solutions, the I/O variables tabulated in Table 9.1 should be provided. This **instruments-free power estimator** (**IFPE**) is illustrated in Figure 9.2.

Now, by combining Figure 9.2 with the guidance given before, any network topology can be easily constructed. For example, Figure 9.3 explains how to generate a dataset of the WSCC 9-bus test system by varying its settings. For this particular

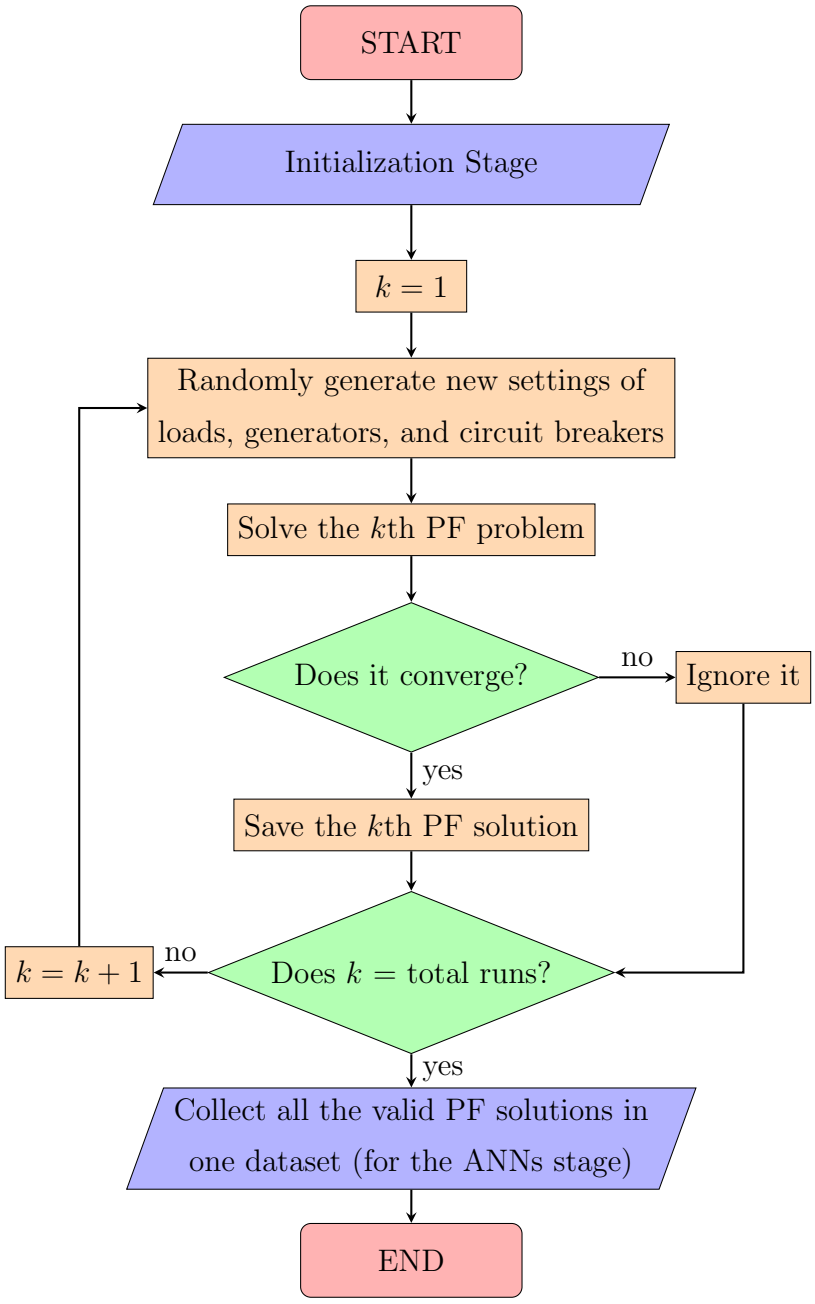


Figure 9.1: General Flowchart Used to Create a Dataset for NNPF

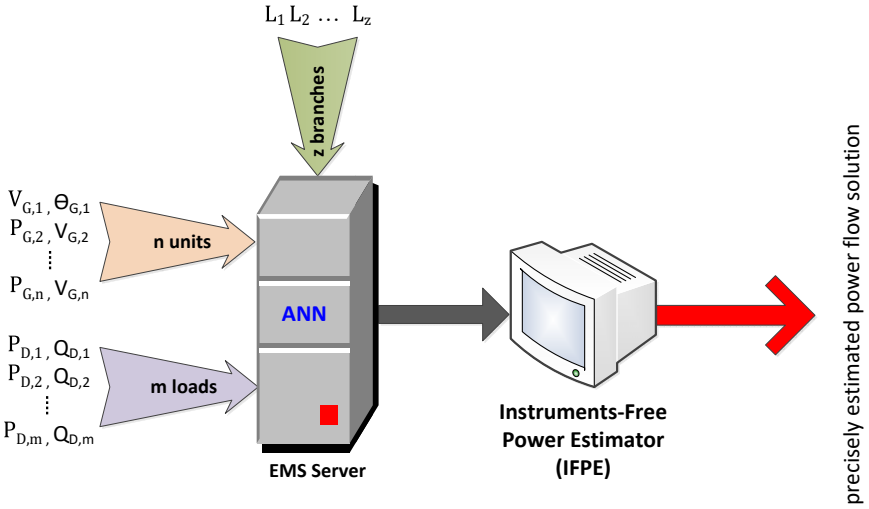


Figure 9.2: Mechanism of the Proposed IFPE Technique for PF

test system, there are 18 dependent variables and 18 independent variables. Thus, the input and output layers of any ANN will have 18 connections for each. Thus, if one layer composed of 30 neurons is used, then this shallow topology can be depicted by Figure 9.4.

It has to be noted that the classical PF solver used during constructing the dataset is executed only one time and it is an offline process. The other option is to use the real-data fed by EMS.

9.1.1.3 Numerical Experiments and Discussion

To validate the process of the proposed NNPF technique, the WSCC 9-bus test system shown in Figure 9.3 is used in the first experiment. The dataset, created by the algorithm shown in Figure 9.1, has a size of 60,000 PF solutions, which is generated by the classical NR algorithm with a minimum acceptable tolerance of $\varepsilon = 10^{-14}$ (i.e., early stopping criterion). In this experiment, the same topology shown in Figure 9.4 is adopted here. Also, the Resilient back-propagation (BP) algorithm is used to train ANNs with the following hyperparameters:

- Maximum number of epochs to train: 100,000
- Performance goal: 0
- Maximum validation failures: 6

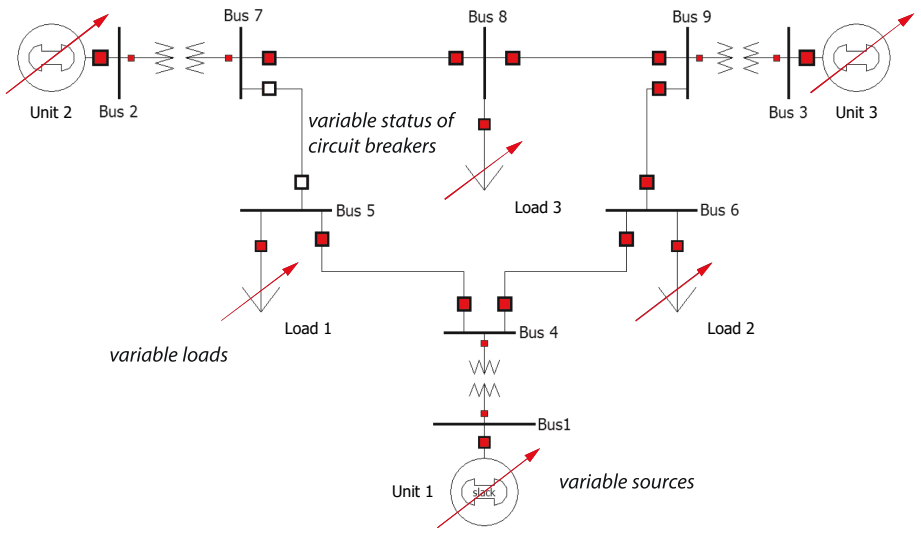


Figure 9.3: Operational and Topological Changes on the WSCC 9-Bus Test System

- Minimum performance gradient: 1×10^{-7}
- Learning rate: 0.01
- Increment to weight change: 1.2
- Decrement to weight change: 0.5
- Initial weight change: 0.07
- Maximum weight change: 50
- Maximum time to train: ∞
- Ratio of vectors for training = 70%
- Ratio of vectors for validation = 15%
- Ratio of vectors for testing = 15%

The program is coded in MATLAB R2017b using a computing machine having the following specifications: ALIENWARE M14x, 64-bit Windows 10 OS, Intel Core i7-4700MQ CPU 2.4 GHz, and 16 GB RAM. The NNPF performance is evaluated using all the fifteen activation functions listed in Chapter 5/Section 5.4.

It is recommended to set the output layer of ANNs with the linear transfer function (i.e., `purelin`) if the goal is to approximate functions [250]; which is the case here[4].

[4]This is just general guidance suggested by some references. The best output activation function type can be obtained by optimizing all the hyperparameters of ANNs. Such process has been seen before in Chapter 5/Section 5.4.

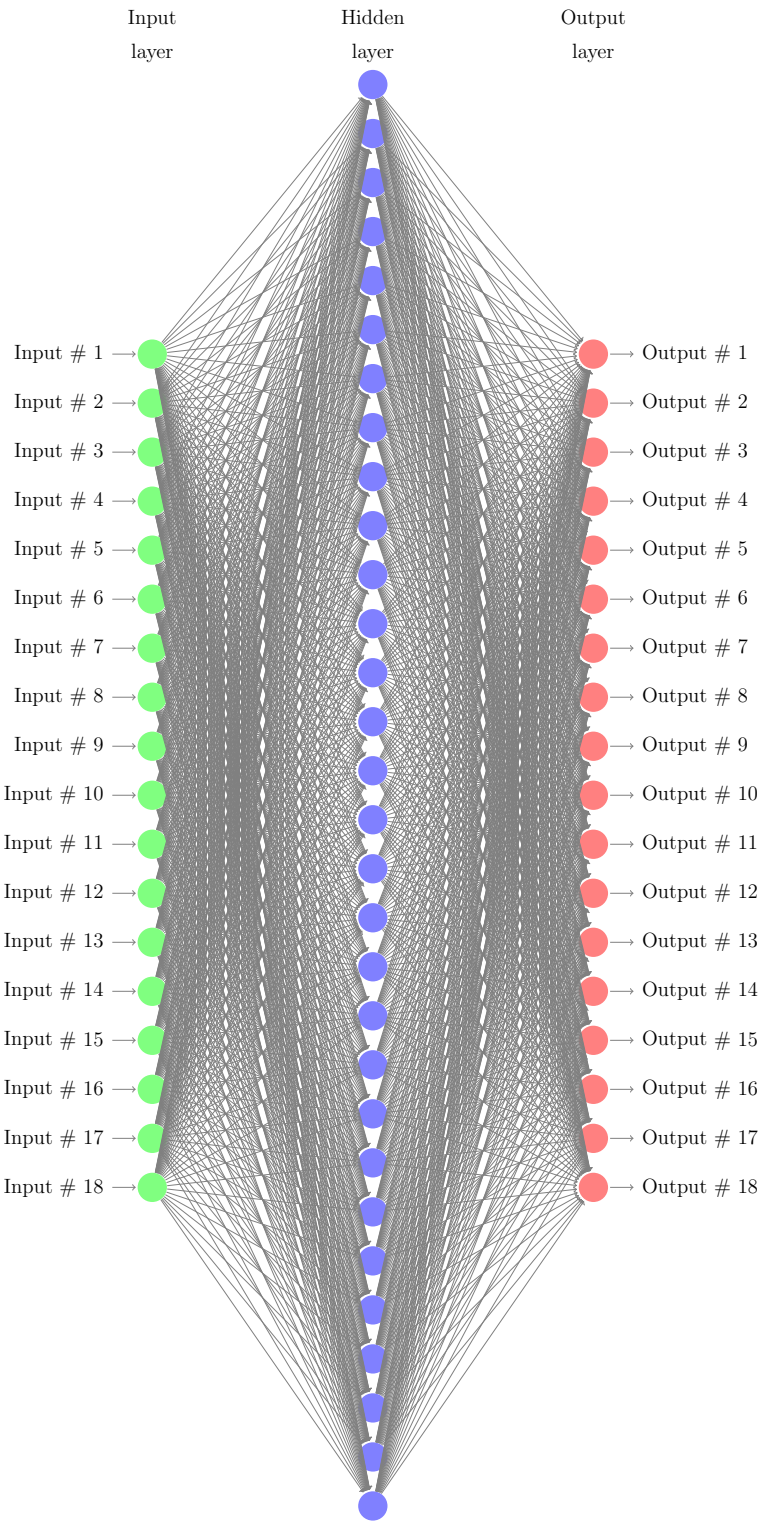


Figure 9.4: Neural Network of the WSCC 9-Bus Test System

Table 9.2: Performance Comparison of Different Activation Functions Used to Solve PF Problems

Activation Function	MSE Performance	No. of Epochs	CPU Time (s)
`compet`	0.044999396280284	29	**1.979**
`elliotsig`	0.000078274898843	**31204**	1862.850
`hardlim`	0.017793944764191	1097	64.805
`hardlims`	0.017740511053517	2394	138.288
`logsig`	0.000084297101917	22002	1405.743
`netinv`	0.101490225612570	51	3.810
`poslin`	0.000153187291679	2738	164.170
`purelin`	0.002112044010414	780	43.317
`radbas`	0.000074912310478	9299	584.633
`radbasn`	0.000047250271178	13091	1707.674
`satlin`	0.000125866986436	9466	602.772
`satlins`	0.000111690367707	17045	1107.460
`softmax`	**0.000038978384340**	14631	1863.899
`tansig`	0.000077143655316	15297	978.837
`tribas`	0.000099901261563	6699	429.498
Best	`softmax`	`elliotsig`	`compet`

Table 9.2 shows the performance of these 15 activation functions in terms of MSE, learning ability, and processing speed. It is obvious that the `softmax` activation function is the winner in terms of MSE. This function consumes 14631 epochs to reach its optimal solution. It is much less than that recorded for `elliotsig` by around 46.9%. However, both transfer functions consume almost the same CPU time, which makes `softmax` slower than `elliotsig`. In contrast, `compet` is the fastest one, but it also has the second-worst MSE value after `netinv`.

As an overall performance of all the 18 output channels of NNPF, Figure 9.5 graphically shows the reduction in MSE per epochs of the train, validation, and test sets. The coefficient of determination scored for the `softmax` activation function is $R^2 = 99.988\%$, which is shown in Figure 9.6. It is impressive with this primitive neural network structure. To view this highly precise approximation, the plots shown

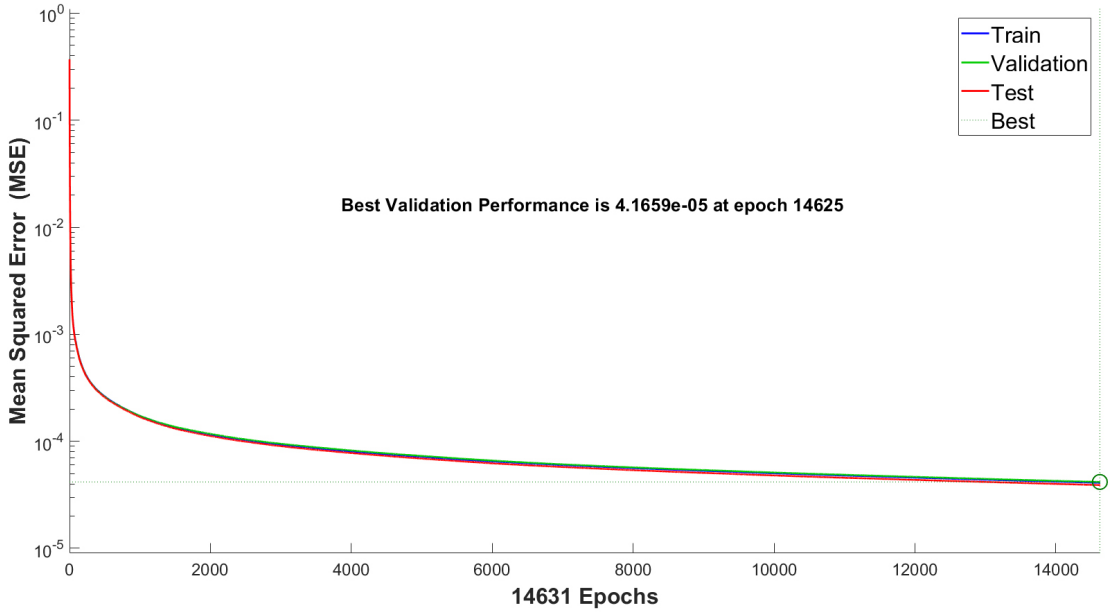


Figure 9.5: MSE Performance of the `softmax`-Based NNPF

in Figure 9.7 depict the relation between the actual and predicted readings at some busbars of the WSCC 9-bus test system.

To see the benefit of NNPF, let's repeat the I/O matrices of the preceding test system until reaching 100,000 variables for each matrix. This virtual data represents a very large network; specifically 50,000-bus system. With its trained neural network, the processing time required to test 100 PF conditions is just 0.933979 second. On the opposite side, the conventional NR solver requires between 33 and 80 seconds to test only one condition of a 20,000-bus system [189]. This simple simulation reveals a possible very important application of NNPF, which is about expediting the processing speed of contingency and other crucial analysis with solving their inherent accuracy problem. By applying NNPF, no need to use any kind of approximations, like DC and AC-DC load flows. That is, all the inherent weaknesses of existing techniques can be solved, permanently, and - at the same time - it can test hundreds of possible PF scenarios within just a very short time.

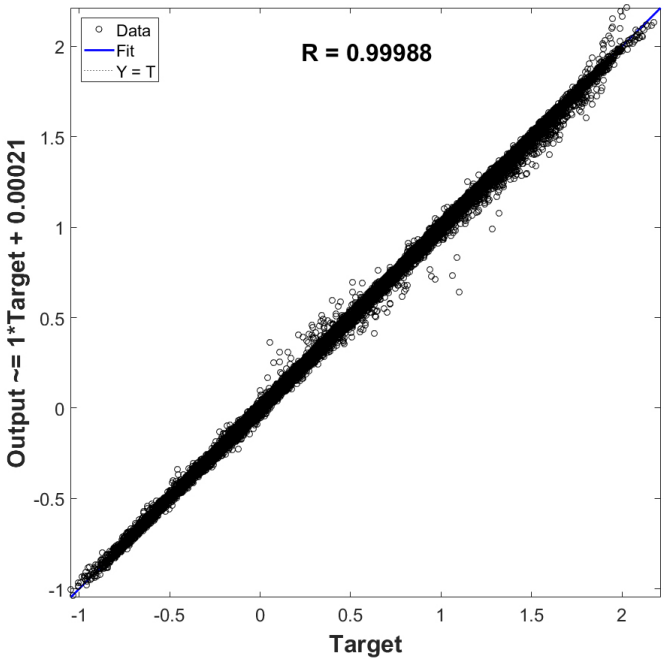


Figure 9.6: Overall Regression Performance of the `softmax`-Based PF Solver

9.1.2 Power Flow Magnitude Estimation

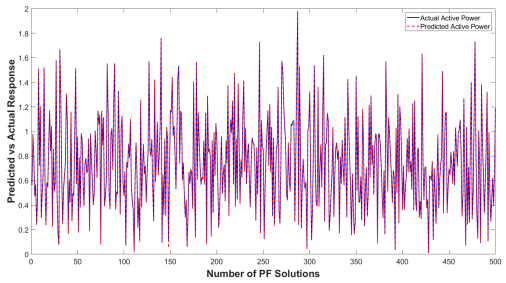
Knowing complex power magnitudes of transmission, sub-transmission, and distribution lines is very important. Practically, a per-second estimate of these online measurements (i.e., complex power magnitudes) is recommended for both ends of each line. This practice is known as a **thermal margin estimation** [83]. The kth measurement can be estimated for the branch between the ith and jth busbars as follows:

$$\mathrm{L}^{\mathrm{th}}_{ij,k} = \frac{S^{\max}_{ij} - \hat{S}_{ij,k}}{S^{\max}_{ij}} \tag{9.6}$$

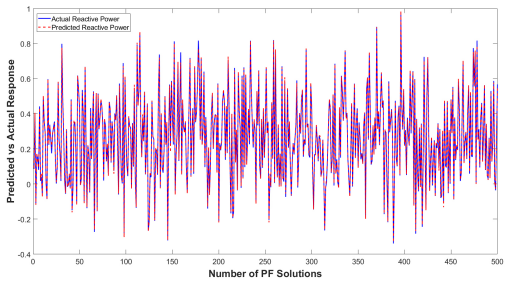
$$\mathrm{L}^{\mathrm{th}}_{ji,k} = \frac{S^{\max}_{ji} - \hat{S}_{ji,k}}{S^{\max}_{ji}} \tag{9.7}$$

where $\mathrm{L}^{\mathrm{th}}_{ij,k}$ and $\mathrm{L}^{\mathrm{th}}_{ij,k}$ are respectively the kth thermal indices of the 1^{st} and 2^{nd} ends of the branch located between the ith and jth busbars. Also, $S^{\max}_{ij}$ and $S^{\max}_{ji}$ are respectively maximum apparent power of the 1^{st} and 2^{nd} ends of that branch. Similarly, $\hat{S}_{ij,k}$ and $\hat{S}_{ji,k}$ are the 1^{st} and 2^{nd} jth estimates.

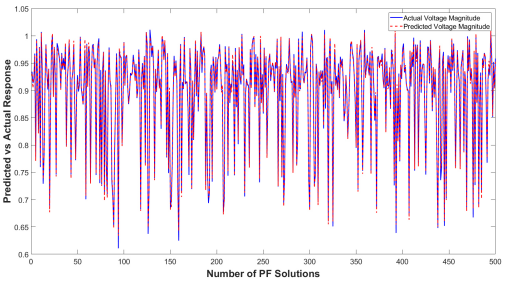
This analysis and the related concepts have many applications in electric power systems, such as reliability [83], ELD [293], UC [275], OPF [142], protection [221],



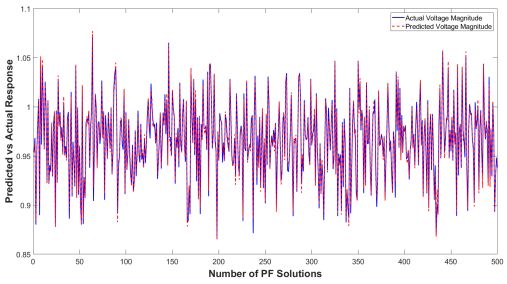
(a) Active Power at Bus 1



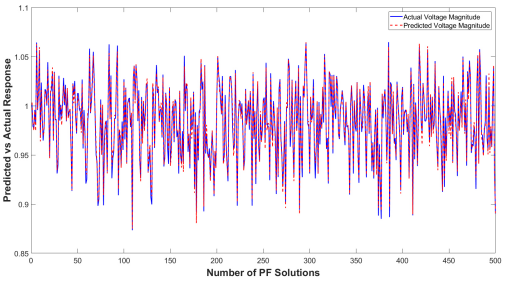
(b) Reactive Power at Bus 3



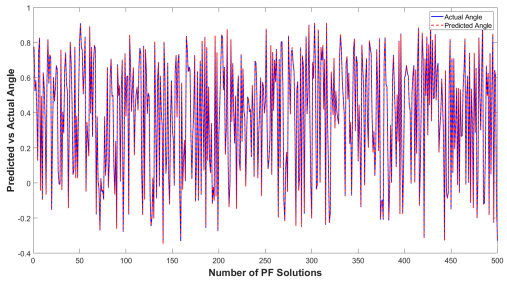
(c) Voltage Magnitude at Bus 5



(d) Voltage Magnitude at Bus 6



(e) Voltage Magnitude at Bus 8



(f) Angle at Bus 9

Figure 9.7: Actual and Estimated PF Measurements at Different Buses of the WSCC 9-Bus Test System

and spot pricing of electricity markets [169, 294].

In real-world applications of electric power systems, the complex power magnitude (or apparent power) can be determined by taking some fundamental measurements through some electric equipment and instrument devices, such as **potential/voltage transformers** (**PTs** or **VTs**), current transformers (CTs), **phasor measurement units** (**PMUs**), **power transducers**, **signal conditioners**, **transmitters**, etc. These signals are sent from the **field level** (level 0) to the **automation level** (level 2) through a middle level called the **control level** (level 1) - as shown in Figure 9.8; plus two top and highly secured levels called the **supervisory level** (level 3) and the

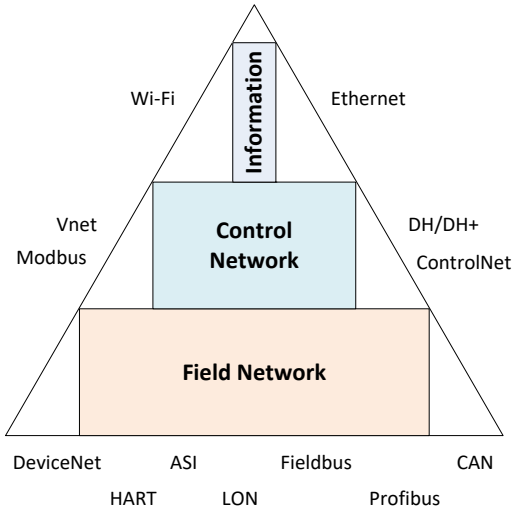


Figure 9.8: Process Automation Levels and Protocols

managerial level (level 4). Different automation systems and sub-systems could be integrated with the energy management system (EMS). Such systems are **supervisory control and data acquisition** (**SCADA** and **microSCADA**), distributed control system (DCS); and other third-party sub-systems, such as **programmable logic controllers** (**PLCs**) and **remote terminal units** (**RTUs**). These online measurements are transmitted between these five levels through some standard protocols (**Fieldbus**, **Profibus**, **HART**, **Modbus**, **Ethernet**, etc) using wired, fiber-optic, or even wireless communication. The final measurements are processed in these automation and energy management systems, and then stored in **archiving servers**. Some of these readings and records are displayed to the respective operators, supervisors, and engineers through some human-machine interface (HMI) stations [290, 335].

As can be clearly seen from this complicated process, the chance of failure to any of these equipment, devices, communications, and systems always exist. Thus, even if there is no any problem with PTs and CTs, the interruption between the actual signals and their corresponding values displayed on HMIs could happen because of signal conditioners, transmitters, communications, automation systems/sub-systems (DCS, SCADA, microSCADA, PLCs, RTUs, etc), or EMS. Depending only on online measurements could lead to a fatal problem. Based on this, different approaches are presented in the literature to either estimate these actual measurements, received from the field, or to ensure that the suspected measurements are not wrong. However, these

techniques require many real online readings to correctly perform their estimations.

This study tries to solve the entire problem by using a totally different technique that does not depend on any of the preceding hardware. This novel technique requires just to know the present status of branches and the power settings (i.e., the real and reactive power $\rightarrow \quad P, Q$) of units and loads to give its precise estimation of complex power magnitudes flowing through all the branches of the given network. This approach is highly practical and feasible, because of the following three safety and economic facts:

- Generation: there is no power plant allowed to operate its units without knowing their actual power settings.
- Transmission/Sub-Transmission: the status of all overhead, underground, and underwater lines/cables are continuously updated and monitored.
- Distribution/Utilization: all power consumptions by customers and end-users are accurately measured and billed.

The working principle of this IFPE can be graphically described in Figure 9.9. Two essential stages are required to fulfill this process:

1. Creating a very large dataset of input and target variables, which can be attained by collecting PF solutions of many randomly generated settings.
2. Training different ANNs to select the best configuration based on the lowest MSE or any other metric.

9.1.2.1 Stage No.1: Creating ANNs Dataset

One of the fundamental steps to have an efficient ANN is to train it with a well-prepared dataset. To make the technique shown in Figure 9.9 feasible, the dataset can be attained by two approaches[5]:

1. Real PF Solutions: collecting all online records stored in EMS servers.

[5]Doing combination/hybridization between them is also valid.

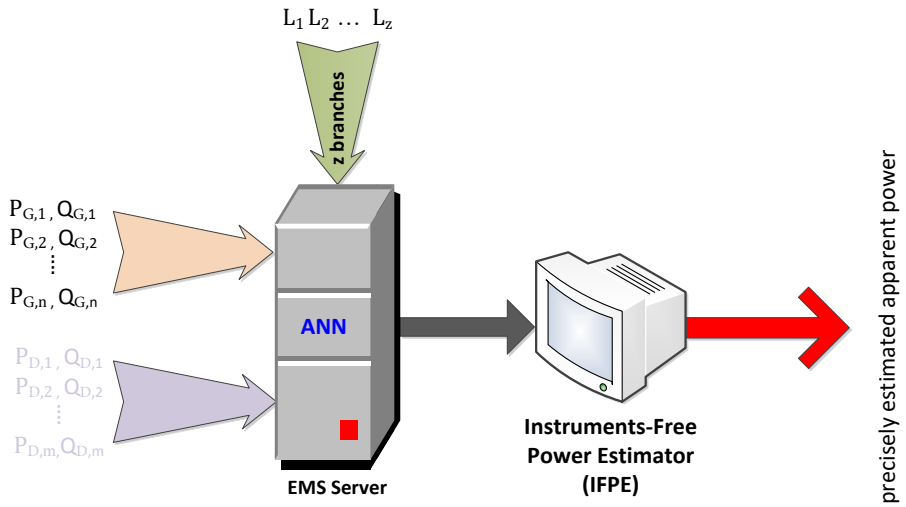


Figure 9.9: Mechanism of the Proposed IFPE Technique to Estimate Power Flow Magnitudes

2. Calculated PF Solutions: collecting PF solutions of many randomly generated operational and topological settings.

If the records of the first approach are vast and diverse[6], then it will definitely provide very accurate results; especially if the surrounding weather conditions and other phenomena are taken into account to reduce the total uncertainty level. However, if all these effects and disturbances are considered, then the second approach is also feasible and applicable [12, 13]. The second approach has many distinct advantages. For example:

- The user can generate many PF solutions as much as he/she wants.
- All the scenarios (operational and topological settings) can be covered without facing any technical problem or safety issue.
- The search space or domain of the problem can be effectively covered instead of depending on some specific patterns of real power system operation.

Based on all these facts and features, the second approach is more flexible, and thus it is selected to demonstrate this innovative technique. The first approach has

[6]To cover the search space of the problem.

the same concept, and thus it can be applied at any time once the mechanism of the proposed technique is fully understood.

To illustrate how this stage is performed, let's use the WSCC 9-bus test system shown in Figure 9.3. The slant arrows mean that the settings of these three generating units and loads are exposed to many changes. Such actions are known as operational configurations. The closing and opening status of circuit breakers (CBs) are respectively indicated by filled and unfilled boxes beside each end of branches. If the both end boxes, of any branch, are unfilled, then they mean that the corresponding CBs take that branch out of service. Such actions are known as topological configurations. This stage can be graphically described by the flowchart given in Figure 9.1. From the given data and PF solutions, this stage can be accomplished by carrying out the following steps [32]:

1. The net active and reactive power at each ith busbar (including the slack bus) can be calculated as follows:

$$P_i^{\text{net}} = P_{G,i} - P_{D,i} \tag{9.8}$$

$$Q_i^{\text{net}} = Q_{G,i} - Q_{D,i} \tag{9.9}$$

where $P_{G,i}$ and $Q_{G,i}$ are respectively the source active and reactive power injected to the ith busbar, and $P_{D,i}$ and $Q_{D,i}$ are respectively the load active and reactive power consumed from that busbar. Both ($P_i^{\text{net}} = Q_i^{\text{net}} = 0$) if the ith busbar is not connected to any source or load.

2. Calculate the complex voltage of each ith busbar:

$$V_i = |V_i| \angle \delta_i = |V_i| \cos(\delta_i) + j\,|V_i| \sin(\delta_i) \tag{9.10}$$

3. Calculate the current flow between the ith and jth busbars:

$$I_{ij} = y_{ij}(V_i - V_j) \tag{9.11}$$

$$I_{ji} = y_{ji}(V_j - V_i) = -I_{ij} \tag{9.12}$$

where y_{ij} is the series admittance between the ith and jth busbars, and vice versa for y_{ji}. Under a steady-state condition, the network nodal admittance matrix is symmetrical, and thus $y_{ij} = y_{ji}$.

4. Calculate the complex power flow from each direction:

$$S_{ij} = V_i I_{ij}^* \tag{9.13}$$

$$S_{ji} = V_j I_{ji}^* \tag{9.14}$$

where $S_{ij} \neq S_{ji} \rightarrow$ because $V_i \neq V_j$.

5. Identify the branch status:

$$\mathfrak{L}_{ij} = \begin{cases} 1 & \text{if } y_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.15}$$

where $\mathfrak{L}_{ij}$ is the status of the branch located between the ith and jth busbars. The branch is considered under outage (i.e., taken out of service) if it shows 0, and energized (i.e., returned back into service) if it shows 1.

The output variables (which are the targets of ANNs) are the apparent power flowing in all the branches, which are the magnitudes of the complex power calculated by (9.13)-(9.14). These magnitudes can be determined as follows:

6. Identify the apparent power flowing from the ith to the jth busbar:

$$|S_{ij}| = \sqrt{P_{ij}^2 + Q_{ij}^2} = \sqrt{\left(\Re\left\{V_i I_{ij}^*\right\}\right)^2 + \left(\Im\left\{V_i I_{ij}^*\right\}\right)^2} \tag{9.16}$$

7. Identify the apparent power flowing from the jth to the ith busbar:

$$|S_{ji}| = \sqrt{P_{ji}^2 + Q_{ji}^2} = \sqrt{\left(\Re\left\{V_j I_{ji}^*\right\}\right)^2 + \left(\Im\left\{V_j I_{ji}^*\right\}\right)^2} \tag{9.17}$$

8. Many studies calculate the both end thermal margins using the apparent power $|S_{ij}|$ and $|S_{ji}|$ [83]. However, this study focuses on the biggest magnitude on each branch, which can be directly determined by applying the following filter:

$$\left|\bar{S}_{ij}\right| = \left|\bar{S}_{ji}\right| = \max\left\{|S_{ij}|, |S_{ji}|\right\} \tag{9.18}$$

By focusing on the most severe power magnitudes, the total number of targets used in ANNs can be reduced to the half, which accelerates the overall processing speed of ANNs. If the both end power magnitudes (i.e., $|S_{ij}|$ and $|S_{ji}|$) are required, then this slight modification can be easily achieved by just deactivating or bypassing (9.18).

9.1.2.2 Stage No.2: Learning Process of ANNs

A brief introduction to ANNs is given in Chapter 5. The objective of this study is to estimate all the apparent power flowing in each branch of electric power networks by just knowing the present status of these branches and the real/reactive power settings of units and loads. That is, ANNs are used to find a relationship between the predictors (or inputs) and the responses (or targets). The dataset of these inputs and targets is created in the 1st stage, which is further explained through the pseudocode given in Algorithm 11. This dataset is split into two matrices: the input matrix is denoted by X, and the output matrix is denoted by U.

9.1.2.3 Numerical Experiment and Discussion

Before starting, it is important to say that analyzing the performance of ANNs under many configurations (different topologies of ANNs: hidden layers, neurons, transfer functions, and BP algorithms) is very important. Thus, the one that has the best performance should be selected as the final confirmed configuration. This approach is covered in Chapter 5. However, the main goal of this study is to prove the feasibility of this innovative IFPE technique. Therefore, it uses the following simple configuration: feed-forward neural network with only one hidden layer composed of 20 neurons. Advanced configurations with optimal hyperparameters could be considered in future work to enhance the overall performance.

Now, let's use the WSCC 9-bus test system shown in Figure 9.3 in this experiment. The number of input and output variables of each PF solution are 18 and 9, respectively. The general neural network of this test system is shown in Figure 9.10. For the 1st stage (i.e., solving multiple PF problems), the NR method is used with the following initialization parameters:

- 60,000 runs with a chance of one outage at each 10 runs.
- Maximum iterations of 50 per each run of NR.
- Minimum acceptable tolerance of $\varepsilon = 10^{-14}$.

For the 2nd stage (i.e., training neural networks), the basic configuration shown in Figure 9.10 is used. To compromise between the processing speed and training quality, the Resilient BP algorithm is used to train ANNs with the same hyperparameters

Algorithm 11 Pseudocode to Create ANN Input and Output Matrices Based on PF Solutions for Estimating Power Magnitudes

Require: the voltage magnitude $|V|$, angle δ, active power P, and reactive power Q of all the buses

for $k \leftarrow 1$ *to* Run **do** {where Run = the number of PF solutions}
  if MTH $> \varepsilon$ **then** {where MTH is the power mismatch between the last two iterations of the kth PF solution}
    Ignore the kth PF solution "because it is invalid"
  else
    for $i \leftarrow 1$ *to* NB **do** {where NB = the number of busbars}
      Apply (9.8) and (9.9) for P_i^{net} and Q_i^{net}
      Apply (9.10) for V_i
      for $j \leftarrow 1$ *to* NIB **do** {where NIB = the maximum number of busbars interconnected with the ith busbar ($j \neq i$)}
        Apply (9.11) and (9.12) for I_{ij} and I_{ji}
        Apply (9.13) and (9.14) for S_{ij} and S_{ji}
        Apply (9.15) for $\mathfrak{L}_{ij}$
        Apply (9.16) and (9.17) for $|S_{ij}|$ and $|S_{ji}|$
        Apply (9.18) for $\left|\bar{S}_{ij}\right|$
      end for
    end for
  end if
  Save the vectors of P^{net}, Q^{net}, and $\mathfrak{L}_{ij}$ in the kth row of the matrix X (i.e., the input matrix of ANNs)
  Save the vector of $\left|\bar{S}_{ij}\right|$ in the kth row of the matrix U (i.e., the output matrix of ANNs)[a]
end for

[a] Or $|S_{ij}|$ and $|S_{ji}|$ in case the both end apparent power are required.

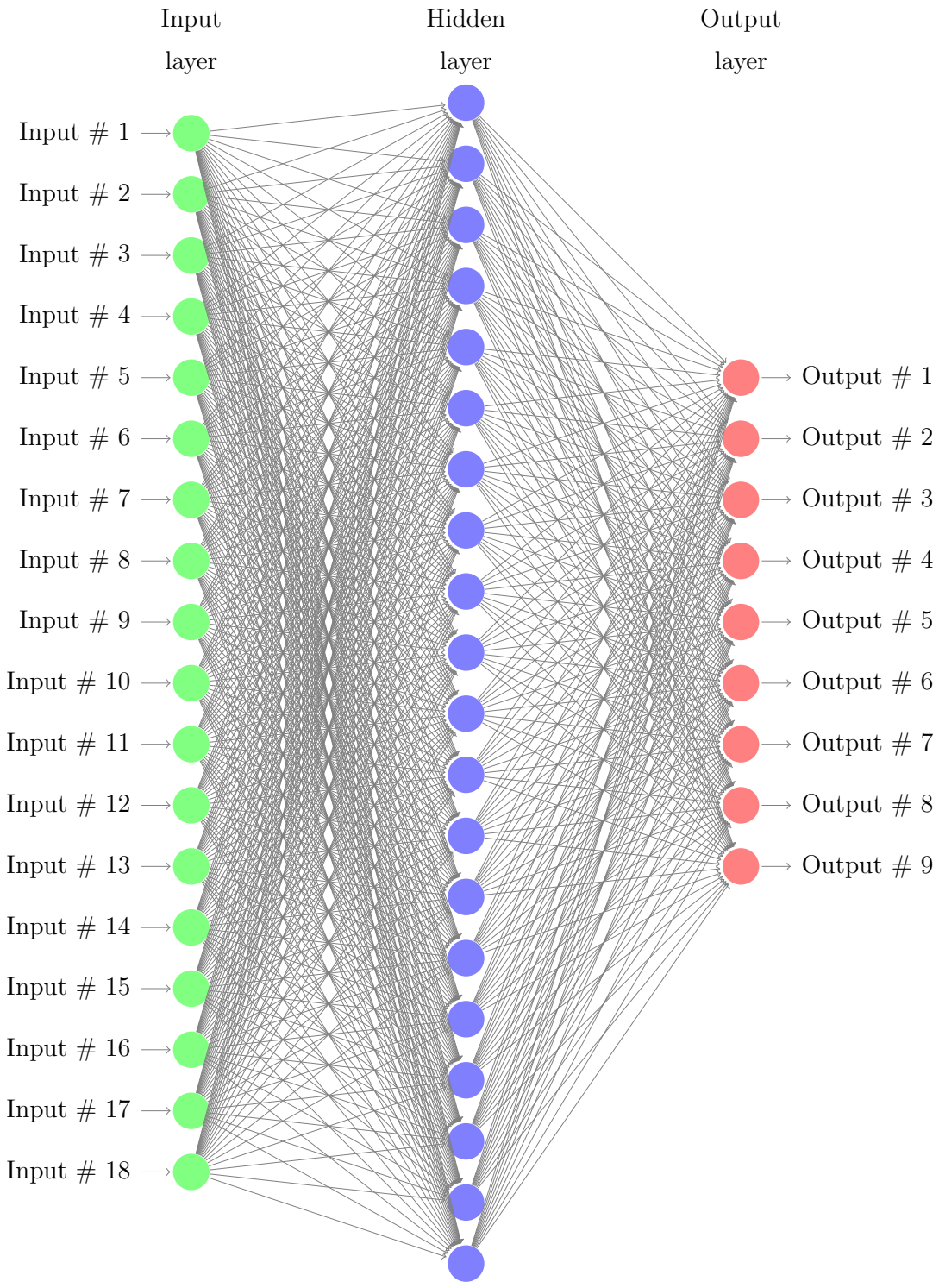


Figure 9.10: Neural Network Used to Estimate Power Flow Magnitudes of the WSCC 9-Bus Test System

used in the preceding subsection. The performance of this ANN is evaluated using the fifteen activation functions listed in Chapter 5.

It is recommended to use the linear transfer function (i.e., `purelin`) for the output layer if the goal is to approximate or regress one or multiple functions [250]; which is the case here.

Table 9.3 shows the performance of these 15 activation functions in terms of MSE, learning ability, and processing speed. As can be clearly seen from that table, the `radbasn` activation function beats all the other 14 functions in terms of MSE. With this activation function, the neural network can be trained with fewer epochs compared with `logsig`, `poslin`, `tansig`, and `radbas` activation functions. However, `radbasn` scores the second slowest processing speed. The reason behind this phenomenon comes from two factors: 1. its internal mechanism and computing strategy, and 2. the total epochs utilized during the learning process.

From the preceding initialization settings, it is obvious that the maximum epochs limit is set equal to 100,000. However, this limit has not been reached by any one of these 15 activation functions, as can be seen in the 3rd column of Table 9.3. The reason for selecting this very large limit is to ensure eliminating the effect of epochs in deciding the best activation function. The goal here is to let ANNs be trained freely without any restriction on their epochs until reaching their saturated performance so one of the early stopping criteria will be triggered. For all these activation functions, only the "**maximum allowable limit of validation failures**" stopping criterion is invigorated. This restriction is not relaxed to avoid **overfitting phenomenon**[7].

The MSE performance of the `radbasn` activation function is graphically shown in Figure 9.11. It can be clearly seen that the three curves are close to each other with very small MSE, which means that the model is highly significant and thus it can precisely estimate power flow magnitudes. This observation can also be proved by plotting the overall regression, as seen in Figure 9.12. From this regression plot, the linearized output has almost a unity slope and zero intercept, which means that the predicted or estimated variables are mirrors of their actual or target variables.

[7]Overfitted curves could show impressive MSE results, but this is just a false performance. Thus, having overfitting in neural networks could show bad behaviors if they are exposed to new input variables. That is, this stopping criterion tries to balance overfitting against **underfitting** concerns optimality.

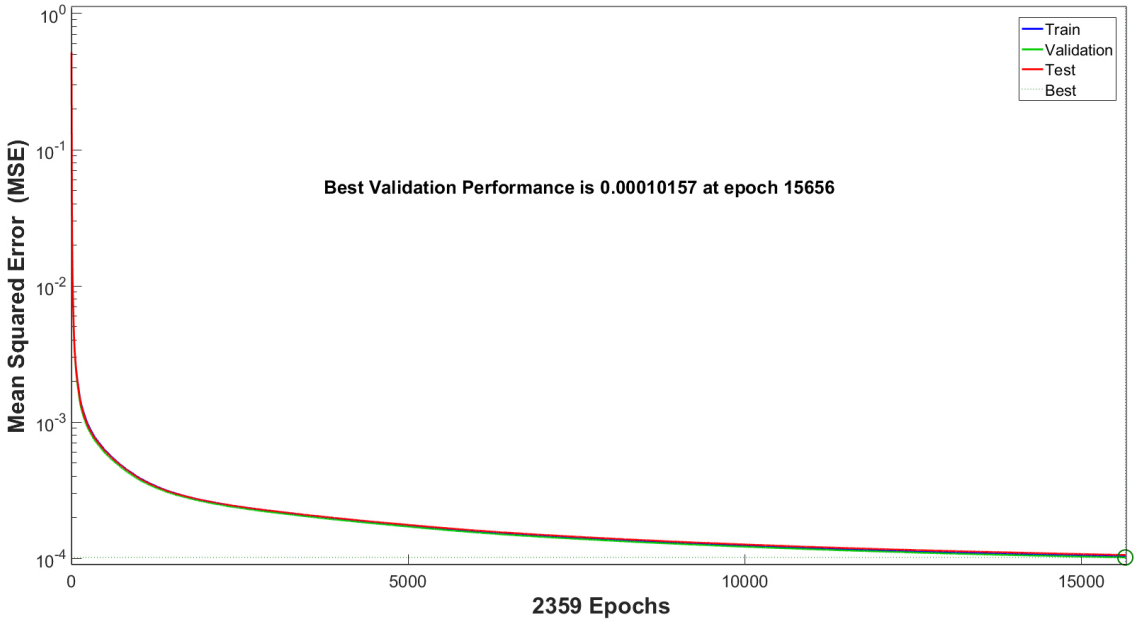


Figure 9.11: MSE Performance of the `radbasn`-Based Power Magnitude Estimator

This can be further proved by looking to the overall explanation level ($R = 0.99987 = 99.987\%$). The difference between actual and estimated values is graphically presented by the **error histogram** with 20 bins shown in Figure 9.13.

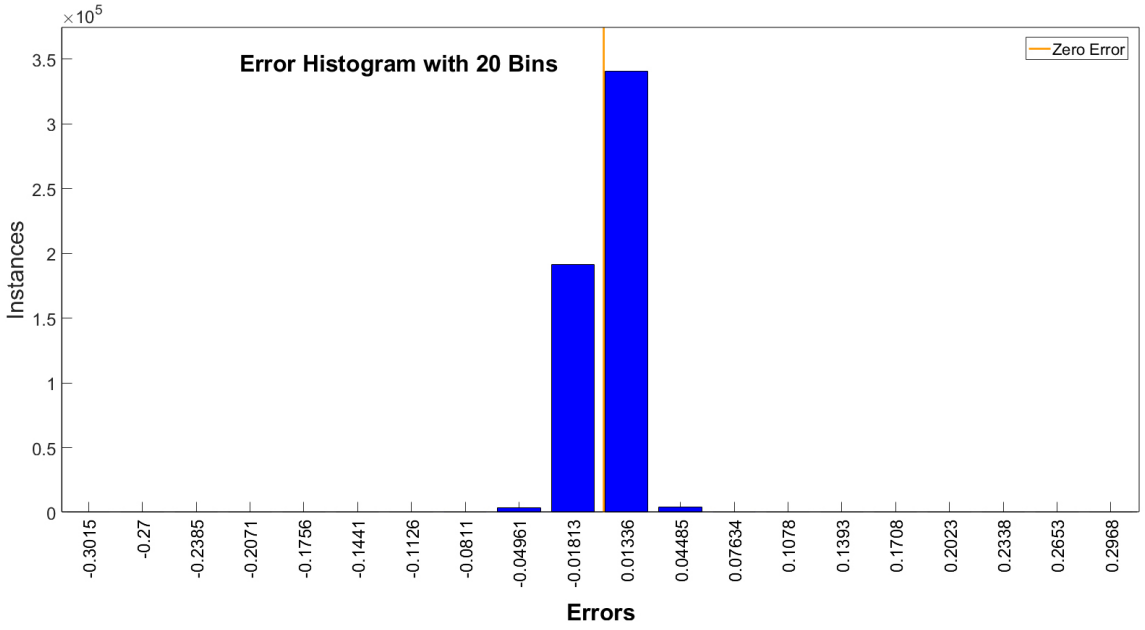


Figure 9.13: Error Histogram of the `radbasn`-Based Power Magnitude Estimator

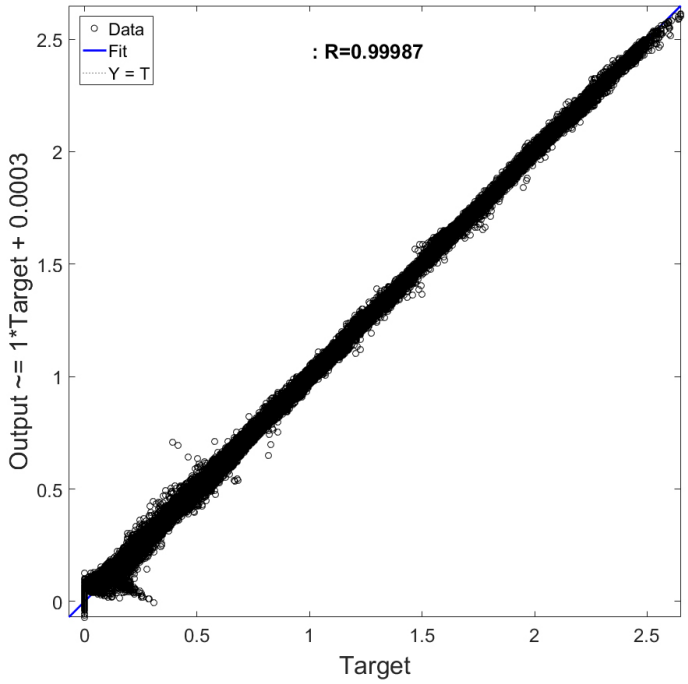


Figure 9.12: Overall Regression Performance of the `radbasn`-Based Power Magnitude Estimator

This `radbasn`-based neural network is supposed to be used for estimating the complex power magnitudes flowing in all the branches of the WSCC 9-bus test system, shown in Figure 9.3, by just knowing the status of the branches and active/reactive power settings of the three generators and loads. Figure 9.14 shows 500 samples of the actual and estimated complex power magnitudes measured in six branches of Figure 9.3. It is obvious that the estimated measurements are highly precise. To support this claim, Table 9.4 shows the average absolute difference errors, average relative errors, and average total errors of all the branches. These errors are respectively calculated, for a dataset of length N, using the following three expressions:

$$\text{AbsDiff}(\%) = \frac{100\%}{N}\sum_{k=1}^{N}\left|U_{ij,k} - \hat{u}_{ij,k}\right| \tag{9.19}$$

$$\text{RelErr}(\%) = \frac{100\%}{N}\sum_{k=1}^{N}\left|\frac{U_{ij,k} - \hat{u}_{ij,k}}{U_{ij,k}}\right| \tag{9.20}$$

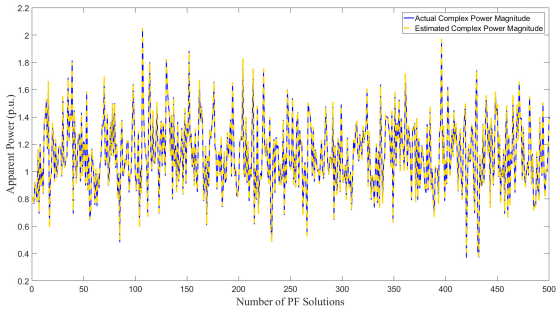
$$\text{TotErr}(\%) = \left|100\% - \frac{100\%}{N}\sum_{k=1}^{N}\left(\frac{\hat{u}_{ij,k}}{U_{ij,k}}\right)\right| \tag{9.21}$$

Table 9.3: Performance Comparison of Different Activation Functions Used to Estimate Power Flow Magnitudes

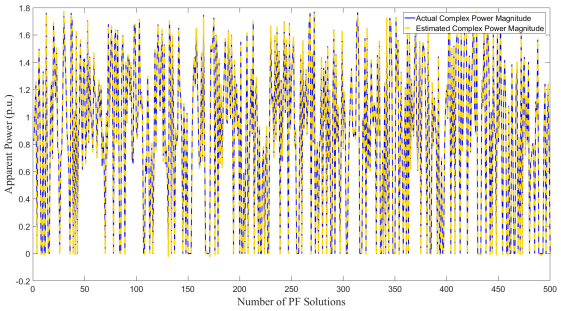
Activation Function	MSE Performance	No. of Epochs	CPU Time (s)
compet	0.144318932938725	25	**0.906**
elliotsig	0.000374303084096	14950	493.987
hardlim	0.064873654423995	663	21.418
hardlims	0.064837793650098	482	15.614
logsig	0.000139615067840	**29412**	1102.677
netinv	0.148531883652532	55	2.142
poslin	0.000131836150440	23739	847.961
purelin	0.005237947727067	2382	70.229
radbas	0.000212415916202	18218	663.829
radbasn	**0.000105986794580**	15662	1010.466
satlin	0.000492111244642	7069	255.400
satlins	0.001019263700212	4276	153.315
softmax	0.000205063555391	8644	555.698
tansig	0.000188347950611	22964	865.376
tribas	0.000302779066838	14664	556.748
Best	radbasn	logsig	compet

where $U_{ij,k}$ and $\hat{u}_{ij,k}$ are respectively the actual and estimated complex power magnitudes flowing in the i-jth branch.

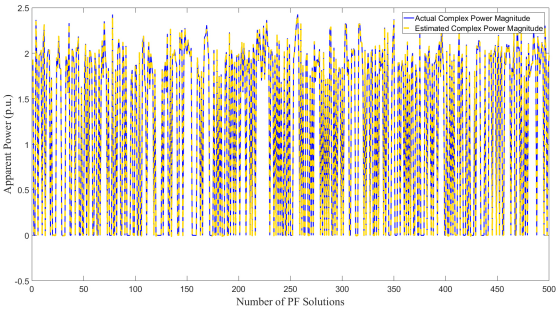
As can be clearly seen from Table 9.4, the overall performance of this innovative technique is pretty impressive where the estimated complex power magnitudes are very close to their actual values. Based on all these facts, the proposed IFPE technique can solve one of the complicated power system estimation problems without using any instrument devices. Add to that, the results shown here are just an example of what could we get in the future from advanced IFPE techniques. Solving this problem by using optimal or semi-optimal hyperparameters could also improve the estimation quality of ANNs with more amazing results.



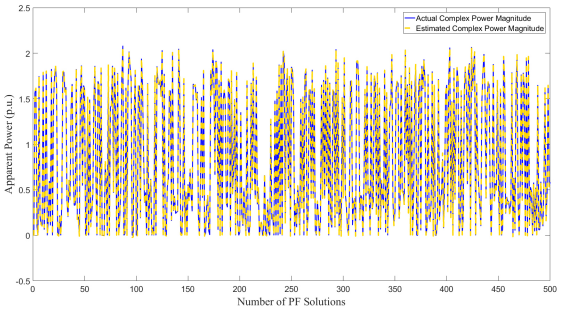
(a) Apparent Power on Bus 1-4 "$\bar{S}_{14}$"



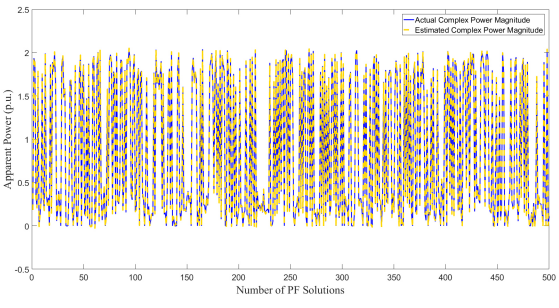
(b) Apparent Power on Bus 4-5 "$\bar{S}_{45}$"



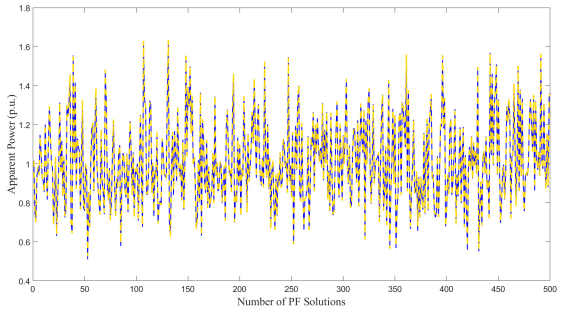
(c) Apparent Power on Bus 5-7 "$\bar{S}_{57}$"



(d) Apparent Power on Bus 6-9 "$\bar{S}_{69}$"



(e) Apparent Power on Bus 7-8 "$\bar{S}_{78}$"



(f) Apparent Power on Bus 8-9 "$\bar{S}_{89}$"

Figure 9.14: Actual and Estimated Apparent Power Measured in Six Branches of the WSCC 9-Bus Test System

Table 9.4: Absolute Difference, Relative, and Total Errors of Power Flow Magnitudes Detected for Each Branch Among 60,000 PF Solutions

Apparent Power Flow	Absolute Difference (%)	Relative Error (%)	Total Error (%)
$\bar{S}_{14}$	0.60960	0.60980	0.01490
$\bar{S}_{27}$	0.52740	0.27570	0.00030
$\bar{S}_{39}$	0.60280	0.67700	0.01360
$\bar{S}_{45}$	0.72740	0.84000	0.03400
$\bar{S}_{46}$	0.68650	0.73370	0.01890
$\bar{S}_{57}$	0.60660	0.41300	0.01620
$\bar{S}_{69}$	0.90320	6.76130	3.95570
$\bar{S}_{78}$	0.95780	5.97500	2.31540
$\bar{S}_{89}$	0.48740	0.51440	0.01380

9.1.3 Power Flow Direction Estimation

Knowing the directions of active and reactive power flows is very important and it has many applications in power system operation, stability, control, automation, and protection. For example, these directions can be effectively utilized in the following applications:

- Active power flow directions can explain the difference in phase angle, and reactive power flow directions can explain the difference in voltage magnitude between both ends of each branch [320].
- These directions are used to activate reverse power protective relays; especially for generators [48, 74].
- They provide useful information to identify the sources of harmonic signals [168].
- Their impacts on the stability of HVDC voltage-source converters [50].
- They have some applications in metering bidirectional power [354].
- Active power flow directions have some effects on real-time spot prices [379].
- Reactive power flow directions can be utilized for having good reactive power compensation [93].

The practical way to identify these directions is by analyzing complex power flows. The signs of their real and imaginary parts can indicate the directions of active and reactive power, respectively. These complex power flows can be calculated based on the fundamental current and voltage measurements. These quantities can be measured by using current and potential transformers (CTs and PTs), and then processed by some field-mounted instruments called "power transducers".

From the last subsection, different levels of power automation and energy management systems have been covered where many possible failures could happen in any part or level of these systems. Thus, if any of these failure possibilities exists, then both real online and classical offline techniques cannot be used anymore to determine power flow directions.

This study tries to solve this technical problem by linking predefined offline power flow solutions with ANNs. This innovative state-estimation technique works independently of the actual sensors where the training phase of its neural network is done before the occurrence of instrument failures. Thus, it provides two distinct features: independent highly precise instruments-free backup measurements with fast processing speed.

9.1.3.1 Stage No.1: Creating ANNs Dataset

The same steps used in the preceding power magnitude estimator are applied here, except the last three steps. The procedure is carried out as follows [33]:

1. Calculate the net active and reactive power at each ith busbar using (9.8) and (9.9), respectively.

2. Calculate the complex voltage of each ith busbar using (9.10).

3. Calculate the current flow from each direction using using (9.11) and (9.12), respectively.

4. Calculate the complex power flow from each direction using (9.13) and (9.14), respectively.

5. Identify the branch status using (9.15).

The output variables (which are the targets of ANNs) are the active and reactive power directions. These directions can be determined as follows:

6. Identify the direction of active power flows:

$$\mathrm{D}_{ij}^{P} = \mathrm{sign}\left(\Re\left\{S_{ij}\right\}\right) = \begin{cases} 1 & \text{if } P_{ij} : i \ \rightarrow \ j \\ 0 & \text{if } \mathfrak{L}_{ij} = 0 \\ -1 & \text{if } P_{ij} : j \ \rightarrow \ i \end{cases} \tag{9.22}$$

where D_{ij}^{P} is the active power flow direction of the branch located between the ith and jth busbars.

7. Identify the direction of reactive power flows:

$$\mathrm{D}_{ij}^{Q} = \mathrm{sign}\left(\Im\left\{S_{ij}\right\}\right) = \begin{cases} 1 & \text{if } Q_{ij} : i \ \rightarrow \ j \\ 0 & \text{if } \mathfrak{L}_{ij} = 0 \\ -1 & \text{if } Q_{ij} : j \ \rightarrow \ i \end{cases} \tag{9.23}$$

where D_{ij}^{Q} is the reactive power flow direction of the branch located between the ith and jth busbars.

9.1.3.2 Stage No.2: Learning Process of ANNs

The PF solutions dataset obtained from the 1st stage is used in ANNs to map its input variables (i.e., the active and reactive power of generators and loads and the status of branches) to its desired output variables (i.e., the active and reactive power flow directions of each branch). There are many ways to represent these input and output variables based on the PF solutions stored in that dataset. The pseudocode given in Algorithm 12 shows one of the possible methods to create the input matrix X and the output matrix U of ANNs. This pseudocode is implemented just after the completion of the 1st stage.

9.1.3.3 Numerical Experiment and Discussion

As said before, the expected solution quality could be enhanced by using different ANN topologies with different hidden layers, neurons, transfer functions, and BP algorithms. However, this study uses the following very simple ANN configuration:

Algorithm 12 Pseudocode to Create ANN Input and Output Matrices Based on PF Solutions for Estimating Active and Reactive Power Directions

Require: the voltage magnitude $|V|$, angle δ, active power P, and reactive power Q of all the buses

for $k \leftarrow 1$ *to* Run **do** {where Run = the number of PF solutions}
  if MTH $> \varepsilon$ **then** {where MTH is the power mismatch between the last two iterations of the kth PF solution}
    Ignore the kth PF solution "because it is invalid"
  else
    for $i \leftarrow 1$ *to* NB **do** {where NB = the number of busbars}
      Apply (9.8) and (9.9) for P_i^{net} and Q_i^{net}
      Apply (9.10) for V_i
      for $j \leftarrow 1$ *to* NIB **do** {where NIB = the maximum number of busbars interconnected with the ith busbar ($j \neq i$)}
        Apply (9.11) and (9.12) for I_{ij} and I_{ji}
        Apply (9.13) and (9.14) for S_{ij} and S_{ji}
        Apply (9.15) for $\mathfrak{L}_{ij}$
        Apply (9.22) and (9.23) for D_{ij}^{P} and D_{ij}^{Q}
      end for
    end for
  end if
  Save the vectors of P^{net}, Q^{net}, and $\mathfrak{L}_{ij}$ in the kth row of the matrix X (i.e., the input matrix of ANNs)
  Save the vectors of D_{ij}^{P} and D_{ij}^{Q} in the kth row of the matrix U (i.e., the output matrix of ANNs)
end for

a shallow feed-forward neural network composed of only one hidden layer and 30 neurons. Optimal hyperparameters of advanced configurations could be considered in future work. Based on the WSCC 9-bus test system shown in Figure 9.3, the number of input and output variables of each PF solution is 18; for each side. The overall neural network of this system is similar to that shown in Figure 9.4.

For the 1st stage (i.e., solving multiple PF problems), the NR method is used with the same initialization parameters used in the power magnitude estimator. For the 2nd stage (i.e., training neural networks), the Resilient BP algorithm is used to train ANNs with the same hyperparameters used in the power magnitude estimator, except the maximum number of validation failures which is 12 instead of 6.

The learning process can be stopped early if any one of the following criteria is triggered: maximum epochs, maximum validation failures, minimum performance gradient. Again, the maximum allowable epochs is set to a very high value. The reason behind this is to ensure that each activation function is allowed to use any number of epochs it needs to converge to its best approximation. Although the maximum validation failures setting is recommended[8] to be in the range of 5-7, this study uses 12 instead. The reason behind this is that the output variables of ANNs are rounded to the nearest digit $\{-1, 0, 1\}$ to indicate the active and reactive power directions; where 1 means P_{ij} or/and Q_{ij} flows from the ith busbar to the jth busbar, and vice versa for -1, and 0 means that the branch between the ith and jth busbars is out of service. Therefore, the maximum validation failures setting can be increased to avoid any termination due to some small plateau hits. This network configuration is evaluated using the same fifteen activation functions listed in Chapter 5. Similar to the preceding ANN techniques, the output layer is always set with the linear transfer function (i.e., `purelin`) because this activation function is recommended for function approximation problems [250].

Table 9.5 shows the performance of these 15 activation functions in terms of MSE, learning ability, and processing speed. It is obvious that the `softmax` activation function scores the best position in terms of MSE, and it consumes fewer epochs and processing time compared with many other activation functions. Although the `compet` and `netinv` are the first and second fastest functions, they are also the worst

[8] To avoid having overfitted model [249].

in terms of MSE. The `poslin` activation function is the slowest one. However, many activation functions need fewer epochs to give better MSE performance. This can be directly reflected in the total processing speed as can be observed when looking to the CPU time column of that table.

However, this is not a final judgment, because the training phase of any one of these ANNs could be terminated by many early stopping criteria; as described earlier. Also, the hyperparameters of the Resilient BP algorithm used to train these networks could affect the final performance. This means that the other activation functions could win by fine-tuning these hyperparameters. Add to that, the performance of these activation functions will change when other BP algorithms (such as Levenberg-Marquardt, Bayesian Regularization, Fletcher-Powell Conjugate Gradient, etc) are involved. Furthermore, the overall performance is affected by the network topology,

Table 9.5: Performance Comparison of Different Activation Functions Used to Estimate Active and Reactive Power Flow Directions

Activation Function	**MSE Performance**	**No. of Epochs**	**CPU Time (s)**
`compet`	0.125699186050990	37	**2.119**
`elliotsig`	0.006569816142747	2302	120.548
`hardlim`	0.084814179882969	2147	111.308
`hardlims`	0.084771192532242	1944	100.933
`logsig`	0.005957980536595	1661	94.514
`netinv`	0.211642744466152	64	3.643
`poslin`	0.006773810499944	**37586**	2088.949
`purelin`	0.036298622040385	2933	139.526
`radbas`	0.006431592735277	8062	455.240
`radbasn`	0.005107402440330	7087	859.234
`satlin`	0.007817321790783	3375	192.624
`satlins`	0.011967274350299	3608	205.059
`softmax`	**0.004126880746314**	2359	264.252
`tansig`	0.005235015968675	5726	330.301
`tribas`	0.007935401444795	5146	308.564
Best	`softmax`	`poslin`	`compet`

the number of hidden layers, and their associated neurons.

Based on this simple shallow configuration and its settings, the Resilient/`softmax`-based neural network is used for estimating the active and reactive power directions of the WSCC 9-bus test system shown in Figure 9.3. However, to be able to use this trained neural network in predicting these directions, the output readings should be rounded to give either -1 (flowing from the ith busbar to the jth busbar), 0 (no power flow; i.e., out of service "outage"), or 1 (flowing from the jth busbar to the ith busbar). After this essential correction to that trained neural network, the active and reactive power flow directions can be estimated for any branch of Figure 9.3 just by knowing the present settings of the three units and loads and also which element, if any, is taken out of service.

Figure 9.15 shows 500 samples of actual and estimated power flow directions measured in four branches selected from Figure 9.3. It is obvious that these comparisons are highly precise. To support this claim, Table 9.6 shows the total errors detected in all the branches. These errors are calculated, for a dataset of length $N = 60,000$ using the following formula:

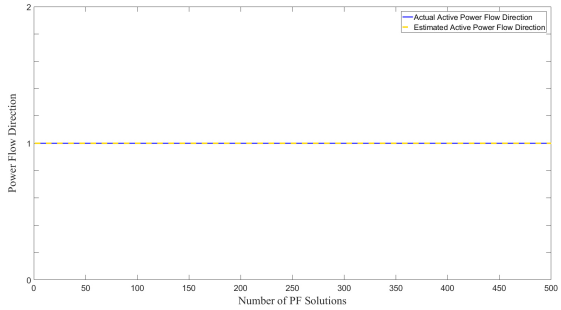
$$\text{PerErr}(\%) = 100\% \times \frac{\text{No. of correct } \lfloor \hat{u}_{ij} \rceil}{\text{No. of } U_{ij} \text{ (i.e., } N)} \tag{9.24}$$

where U_{ij} and $\lfloor \hat{u}_{ij} \rceil$ are respectively the actual and rounded estimated power flow directions of the i-jth branch.

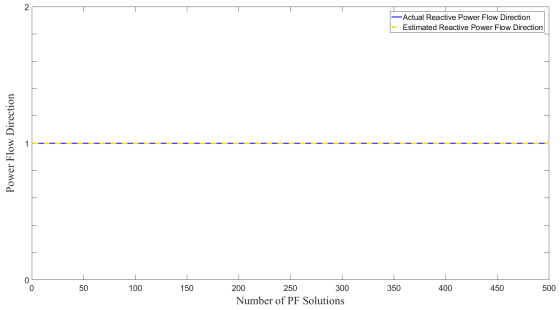
As can be clearly seen from Table 9.6, the overall performance of this innovative technique is pretty impressive. This performance is graphically shown in Figure 9.16. The maximum errors happen on the branches (6-9) and (7-8), which are also very small. These errors could be further minimized by employing more advanced neural networks and optimizing their hyperparameters.

9.1.4 Active and Reactive Power Losses

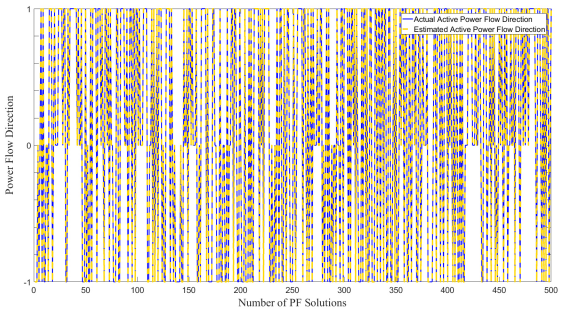
Real transmission, sub-transmission, and distribution lines, and power and inter-bus transformers all are lossy mediums. Thus, it is not revealing a secret to say that knowing active and reactive power losses are very crucial issues in modern electric power systems. These losses can be seen from different views. For example:



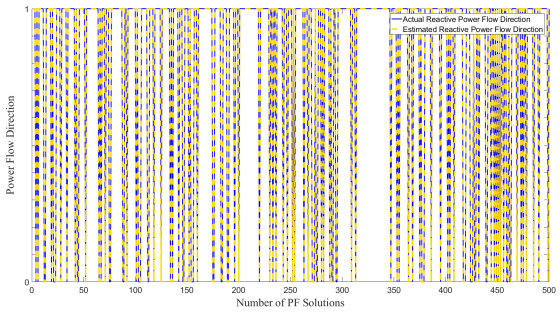
(a) Active Power Flow Direction Between Bus 2 and 7 "D_{27}^{P}"



(b) Reactive Power Flow Direction Between Bus 2 and 7 "D_{27}^{Q}"



(c) Active Power Flow Direction Between Bus 4 and 5 "D_{45}^{P}"



(d) Reactive Power Flow Direction Between Bus 4 and 5 "D_{45}^{Q}"

Figure 9.15: Actual and Estimated Power Flow Directions Measured in Four Branches of the WSCC 9-Bus Test System

- Losing power across branches means an additional power should be generated to cover that shortage.
- Incorrect measurement of these power losses could lead to non-optimal/infeasible solutions of some power operation studies.
- Producing extra power, to compensate the network losses, means losing extra money in power production.
- Knowing resistive and inductive branches helps in determining the best producer in the electricity markets.
- Knowing resistive and inductive branches helps in determining the optimal placements of capacitors and distributed generators (DGs).

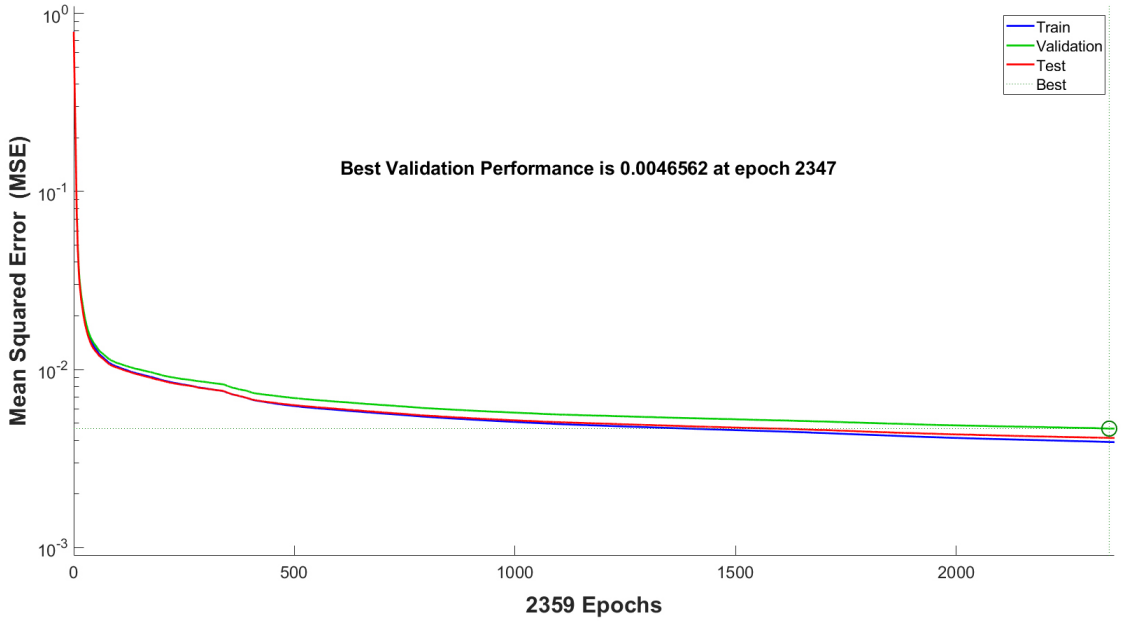


Figure 9.16: MSE Performance of the `softmax`-Based Power Direction Estimator

Table 9.6: Total Errors of Active and Reactive Power Flow Directions Detected for Each Branch Among 60,000 PF Solutions

Active Power		**Reactive Power**	
Direction	**Total Error (%)**	**Direction**	**Total Error (%)**
D_{14}^{P}	0.000	D_{14}^{Q}	0.628
D_{27}^{P}	0.000	D_{27}^{Q}	0.000
D_{39}^{P}	0.000	D_{39}^{Q}	0.055
D_{45}^{P}	0.143	D_{45}^{Q}	0.002
D_{46}^{P}	0.018	D_{46}^{Q}	0.495
D_{57}^{P}	0.017	D_{57}^{Q}	0.020
D_{69}^{P}	1.263	D_{69}^{Q}	1.272
D_{78}^{P}	2.227	D_{78}^{Q}	0.783
D_{89}^{P}	0.018	D_{89}^{Q}	0.682

- Active power losses can give an initial figure about the current thermal margin of each branch, and whether it is close to its maximum limit or not.
- Reactive power losses can give an initial figure about voltage drops in some buses, which in turn helps to decide whether a **capacitor bank(s)** is needed.
- Both active and reactive power losses can give some information about the characteristics of the branch.

Based on this, active and reactive power losses can be seen in many power system studies, such as economic load dispatch (ELD) [30, 41], unit commitment (UC) [312], optimal power flow (OPF) [72], electricity markets [224], voltage instability [86], **optimal location** of **unified power flow controllers** (**UPFCs**) [317], **line expansion planning** [396], **optimal placement** of capacitors [96], optimal placement of DGs [129], etc.

This fact forced many researchers to think about how to correctly and precisely measure these active and reactive power losses, so the overall uncertainty can be minimized and hence the chance to get feasible solutions is increased. There are two main methods can be used to measure or estimate these power losses:

9.1.4.1 Online Active and Reactive Power Losses Calculation Approach

This method is straightforward. It is actually based on the law of conservation of energy where the energy can neither be created nor destroyed. That is, the losses can be calculated by subtracting the power consumption from the power production. It can be carried out for both the entire system and each i-j branch. Also, it can be done for both active and reactive power. For the entire system, this method can be mathematically expressed as follows:

$$P_L = P_{\text{in}} - P_{\text{out}} \tag{9.25}$$

$$Q_L = Q_{\text{in}} - Q_{\text{out}} \tag{9.26}$$

where P_L and Q_L are respectively the total active and reactive power losses in the network. P_{in} and Q_{in} are respectively the total real and reactive power injected to the network. P_{out} and Q_{out} are respectively the total real and reactive power consumed from the network.

For the active and reactive power losses across each branch, the following expressions can be used:

$$P_{L,ij} = P_{ij} - P_{ji} \tag{9.27}$$

$$Q_{L,ij} = Q_{ij} - Q_{ji} \tag{9.28}$$

where $P_{L,ij}$ and $Q_{L,ij}$ are respectively the active and reactive power losses in the branch located between the ith and jth busbars where the flow starts from the ith bus to the jth bus.

Some of the technical problems associated with this approach are:

- It depends on many electrical and instrument devices to measure real-time network losses and send these measurements from the field to the substation, local control room (LCR), or main control room (MCR); where both the energy management system (EMS) and automation system(s) are installed in. These devices are exposed to many failures where some of them occur due to unpredicted/unexpected events.

- These real-time measurements are valid just for the current moment. Thus, this approach cannot be used to solve the previous power system studies, because their optimal settings can be extracted by investigating many scenarios and configurations of the given network.

- It is not allowed to simulate risky scenarios, such as studying the power losses behavior during a partial black-out or faulty/unbalanced state of the network.

As can be clearly concluded, although this approach can give precise measurements of power losses, all the preceding technical problems made it unsuitable in determining power losses of many applications, embedded as packages, in EMS, such as ELD, UC, OPF, etc. Therefore, the second approach is selected in this study, which is explained below.

9.1.4.2 Offline Active and Reactive Power Losses Calculation Approach

There are many techniques presented in the literature, which are listed as offline methods. Some of them are given in Chapter 7. By comparing the methods of

this approach with the one presented earlier (i.e., online approach), many technical issues should be carefully addressed to be able to use them in real-world problems. Such issues are: 1. the gap between real systems and their mathematical models, 2. many simplifications and assumptions are considered in most of these methods, 3. their algorithms require different complicated steps, and 4. they are slack dependent methods. The last issue can be mathematically described by the following equations:

$$\sum_{i=1}^{n} P_{G,i} = \sum_{j=1}^{m} P_{D,j} + P_L \tag{9.29}$$

$$\sum_{i=1}^{n} Q_{G,i} = \sum_{j=1}^{m} Q_{D,j} + Q_L \tag{9.30}$$

where $P_{G,i}$ and $Q_{G,i}$ are respectively the net active and reactive power generated by the ith unit (i.e., after subtracting the auxiliary power consumptions of these n units). $P_{D,j}$ and $Q_{D,j}$ are respectively the active and reactive power consumed by the jth load. P_L and Q_L are respectively the total active and reactive power losses in the network.

The main problem is that the last two terms (i.e., P_L and Q_L) are functions of many independent variables. Two of these variables are the slack active and reactive power (i.e., $P_{G,1}$ and $Q_{G,1}$ if the 1st unit is taken as a slack unit). These stiff equality constraints can be translated as "***Which Came First: the Chicken or the Egg?***".

9.1.4.3 The Proposed AI-Based Active and Reactive Power Losses Calculation Approach

This study tries to solve all the technical problems of both the online and offline approaches. This novel IFPE is graphically described in Figure 9.17.

By this new technique, both active and reactive power losses can be precisely estimated for any power network configuration[9]. More than that, this technique can estimate the active and reactive power produced by the slack unit. Thus, no more power balance equality constraints will be faced in ELD, UC, or OPF. This means a big improvement to the existing techniques used in the literature to satisfy this stiff

[9]i.e. the operational and topological changes, which are described in Figure 9.3.

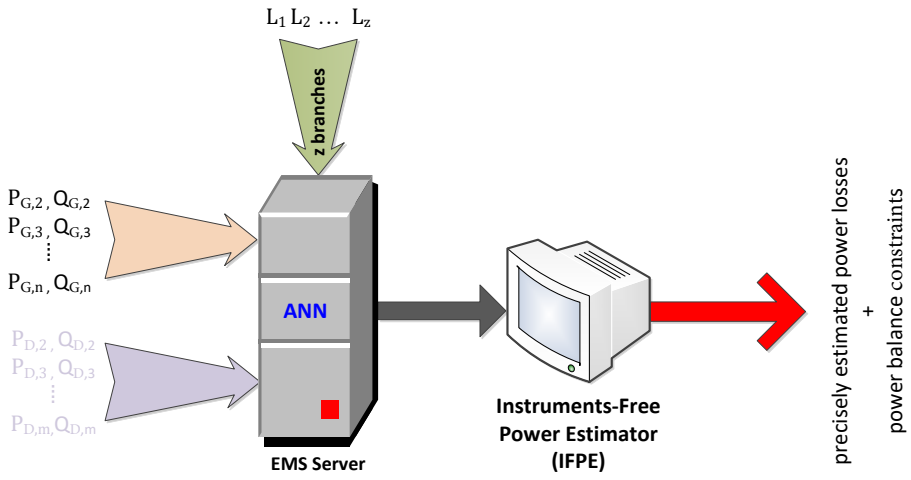


Figure 9.17: Mechanism of the Proposed IFPE Technique to Estimate Active and Reactive Power Losses and Satisfy System Power Balance Equality Constraints

constraint[10]. Add to that, all the points reported in [7,12,13] can be considered to reduce the gap between real systems and their mathematical models[11]. This realization phase could be covered in future work.

Based on all these features, the IFPE technique proposed in this study can satisfy all the key performance criteria: feasibility, accuracy, precision, speed, simplicity, and flexibility.

9.1.4.4 Stage No.1: Dataset Creation

The dataset of inputs and targets shown in Figure 9.17 can be created by using the offline PF approach. Hybridizing it with the online approach could be a good future improvement. As can be clearly seen from Figure 9.17, the inputs of ANNs are: 1. the active and reactive power of non-slack units, 2. the active and reactive power of loads, and 3. the status of branches. On the other side, the targets of ANNs are: 1. the slack active and reactive power, 2. the overall active and reactive power losses dissipated from/absorbed in the network, and 3. the active and reactive power losses of each branch. As said before, the slack unit is shifted to the output side to solve the stiff power balance equality constraint, which is one of the main contributions of

[10]Please, refer to Chapters 3 and 7.
[11]i.e., to minimize the error between the online and offline approaches. Please, refer to Chapter 6.

this study. Thus, this IFPE can hit many birds with one stone. Such birds are: OPF, ELD, UC, and electricity markets.

Similar to the preceding AI-based power estimation techniques, the flowchart shown in Figure 9.1 gives a graphical representation of how to create the ANN dataset in this stage. The mathematical backbone of this flowchart is briefly described by the following steps [35]:

1. The net active and reactive power at each ith busbar (including the slack bus) can be calculated using (9.8) and (9.9), respectively.

2. Calculate the complex voltage of each ith busbar using (9.10).

3. Calculate the current flow from each direction using (9.11) and (9.12), respectively.

4. Calculate the complex power flow from each direction using (9.13) and (9.14), respectively.

5. Identify the branch status using (9.15).

The output variables (which are the targets of ANNs) are the active and reactive power of the slack unit and the active and reactive power losses in the entire network and in each branch. These output variables can be determined by carrying out the following steps:

6. Practically, the slack unit is the one that has the largest power capacity. For simplicity, let's take it as the first unit. Therefore, these two target vectors can be determined by (9.8) and (9.9).

7. The overall active and reactive power losses of the network, which are the third and fourth targets of ANNs, can be easily computed by merging (9.25) with (9.29) and (9.26) with (9.30) as follows:

$$P_L = P_{\text{in}} - P_{\text{out}} = \sum_{i=1}^{n} P_{G,i} - \sum_{j=1}^{m} P_{D,j} \tag{9.31}$$

$$Q_L = Q_{\text{in}} - Q_{\text{out}} = \sum_{i=1}^{n} Q_{G,i} - \sum_{j=1}^{m} Q_{D,j} \tag{9.32}$$

8. The other targets are the active and reactive power losses occurring in all the network branches. This can be determined by extracting the real and imaginary parts of (9.13) and (9.14) as follows:

$$P_{L,ij} = \Re\left\{V_i I_{ij}^*\right\} + \Re\left\{V_j I_{ji}^*\right\} = \Re\left\{S_{ij}\right\} + \Re\left\{S_{ji}\right\} \tag{9.33}$$

$$Q_{L,ij} = \Im\left\{V_i I_{ij}^*\right\} + \Im\left\{V_j I_{ji}^*\right\} = \Im\left\{S_{ij}\right\} + \Im\left\{S_{ji}\right\} \tag{9.34}$$

Since the power losses are the residuals, obtained by subtracting $\{P_D, Q_D\}$ from $\{P_G, Q_G\}$, so someone could notice that (9.33) and (9.34) use "+" instead of "-". The subtraction occurs because one of the terms (i.e., S_{ij} or S_{ji}) will be a negative value due to "-" of (9.12).

9.1.4.5 Stage No.2: Function Approximation

To approximate the actual responses of power losses, the dataset created in the preceding stage is effectively regressed using ANNs. The best trained ANN is used to explain the variability of the dataset. All these outputs should be accurately and precisely estimated based on the present network topology and the active and reactive power of all non-slack units and loads. That is, the best ANN acts as a regression unit to find a relationship between the predictors (or inputs) and the responses (or targets). The dataset of these inputs and targets is further explained through the pseudocode given in Algorithm 13. The matrix of this dataset is split into four submatrices: the first and second ones are called the input matrices and they are denoted by X_1 and X_2 for the active and reactive inputs, respectively. The third and fourth ones are called the output matrices and they are denoted by U_1 and U_2 for the active and reactive targets, respectively.

9.1.4.6 Numerical Experiments and Discussion

Again, the performance of any neural network could be enhanced by finding the optimal hyperparameters[12], which could be considered in future work. The objective of this study is to show that the innovative estimator is feasible and applicable. Therefore, the performance of the IFPE technique is evaluated using the following

[12] Also, EAs can be adopted here to find the global optimal parameters (i.e., the weights and biases associated with each neuron), which is out of the scope of this study.

Algorithm 13 Pseudocode to Create ANN Input and Output Matrices Based on PF Solutions for Estimating Slack Power and Network Losses

Require: the voltage magnitude $|V|$, angle δ, active power P, and reactive power Q of all the buses

for $k \leftarrow 1$ *to* Run **do** {where Run = the number of PF solutions}
  if MTH $> \varepsilon$ **then** {where MTH is the power mismatch between the last two iterations of the kth PF solution}
    Ignore the kth PF solution "because it is invalid"
  else
    for $i \leftarrow 1$ *to* NB **do** {where NB = the number of busbars}
      Apply (9.8) and (9.9) for P_i^{net} and Q_i^{net}
      Apply (9.10) for V_i
      for $j \leftarrow 1$ *to* NIB **do** {where NIB = the maximum number of busbars interconnected with the ith busbar ($j \neq i$)}
        Apply (9.11) and (9.12) for I_{ij} and I_{ji}
        Apply (9.13) and (9.14) for S_{ij} and S_{ji}
        Apply (9.15) for $\mathfrak{L}_{ij}$
        Apply (9.31) and (9.32) for P_L and Q_L
        Apply (9.33) and (9.34) for $P_{L,ij}$ and $Q_{L,ij}$
      end for
    end for
  end if
  Except $P_{G,1}$, save the vectors of P_G, P_D, and $\mathfrak{L}_{ij}$ in the kth row of the matrix X_1 (i.e., the active input matrix of ANNs)
  Except $Q_{G,1}$, save the vectors of Q_G, Q_D, and $\mathfrak{L}_{ij}$ in the kth row of the matrix X_2 (i.e., the reactive input matrix of ANNs)
  Save the singular value of $P_{G,1}$ and the vectors of P_L and $P_{L,ij}$ in the kth row of the matrix U_1 (i.e., the active output matrix of ANNs)
  Save the singular value of $Q_{G,1}$ and the vectors of Q_L and $Q_{L,ij}$ in the kth row of the matrix U_2 (i.e., the reactive output matrix of ANNs)
end for

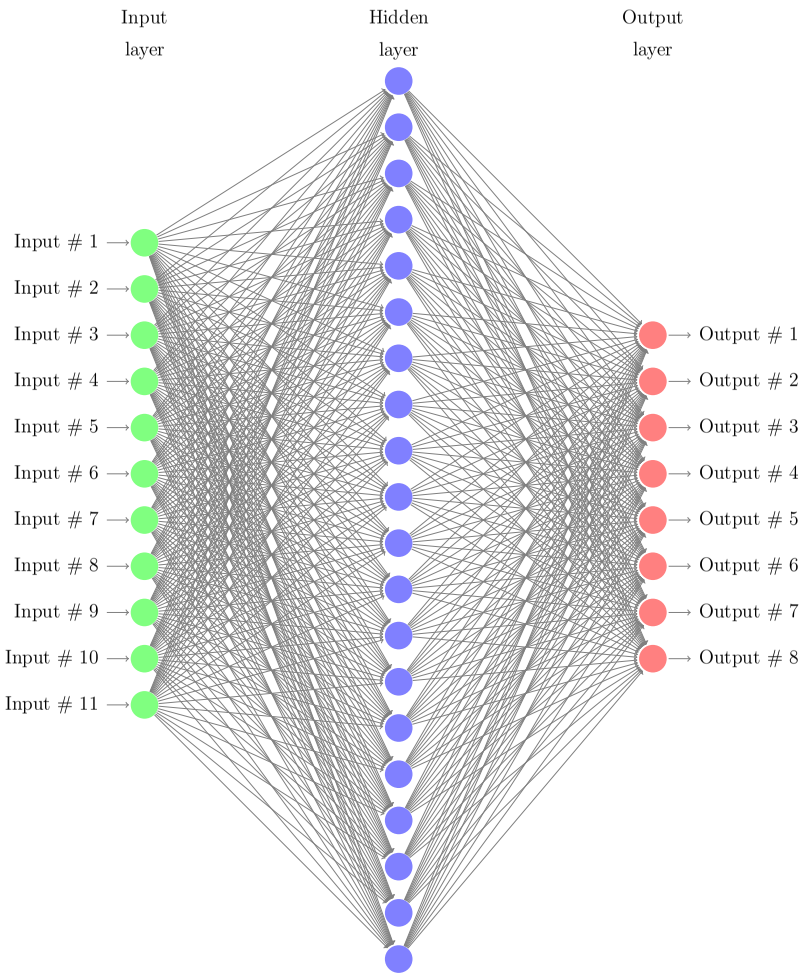
Table 9.7: Overall Performance of ANNs to Estimate Active Power Losses

Activation Function	MSE Performance	No. of Epochs	CPU Time (s)
compet	0.011822125000279	25	**0.804**
elliotsig	0.000052719041495	1287	35.751
hardlim	0.007014592796666	432	12.011
hardlims	0.006987850017624	1164	32.312
logsig	0.000004453022899	19917	635.095
netinv	0.001213853362603	612	18.727
poslin	0.000008868317961	11880	356.392
purelin	0.000269991154902	4227	105.640
radbas	0.000005950547674	9186	283.307
radbasn	0.000005490789963	4011	256.063
satlin	0.000008818409111	10393	332.152
satlins	0.000015205352514	3321	104.183
softmax	**0.000002755418597**	11170	692.881
tansig	0.000004177501344	**23282**	761.528
tribas	0.000008802852260	3844	125.353
Best	softmax	tansig	compet

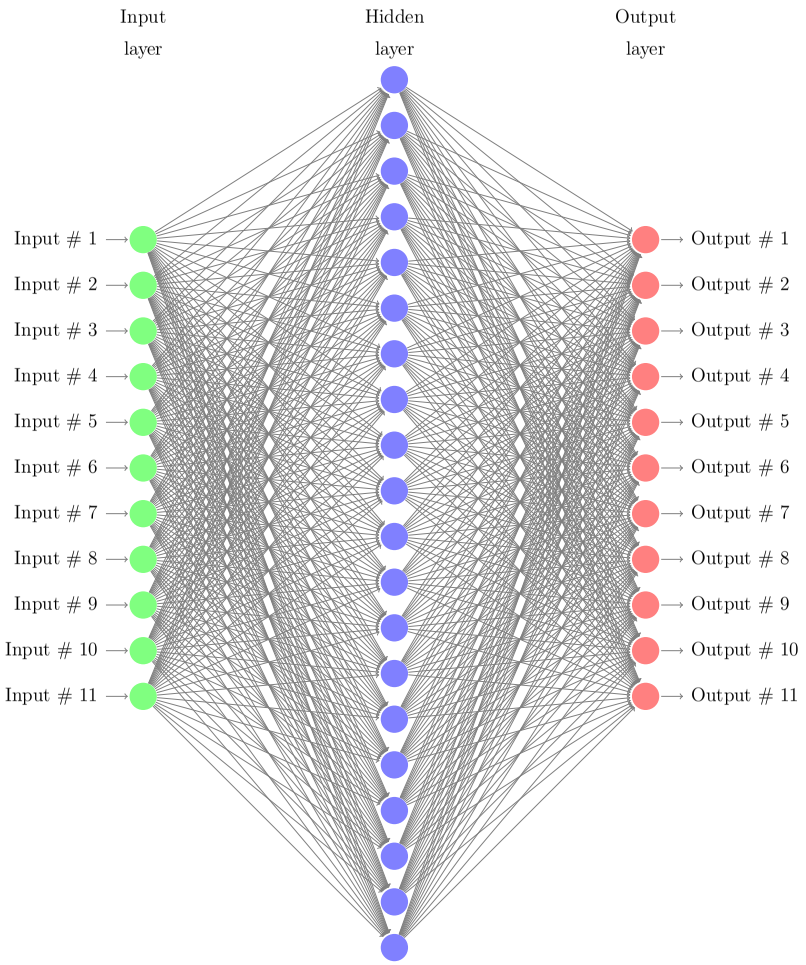
simple configuration: a shallow feed-forward neural network composed of only one hidden layer and 20 neurons. In this experiment, the WSCC 9-bus test system shown in Figure 9.3 is used to evaluate the performance of the IFPE technique shown in Figure 9.17. Based on the system data given in [26], the three power transformers are taken with zero resistance. The proposed ANNs for both the active and reactive power of the WSCC 9-bus test system are shown in Figure 9.18.

The same initialization hyperparameters used in the power magnitude ANN-based estimator are applied here. Also, all the fifteen activation functions listed in Chapter 5 are evaluated before selecting the best type.

Based on Algorithm 13, the active power production of the slack unit and the active power losses of the network and branches can be estimated using Figure 9.18a with the preceding 15 activation functions. Table 9.7 shows the overall performance, which is tabulated in terms of MSE, learning ability, and processing speed. As can



(a) Active Power Estimator



(b) Reactive Power Estimator

Figure 9.18: Neural Networks Used to Estimate the Active and Reactive Power of the Slack Unit and Network/Branches Losses of the WSCC 9-Bus Test System

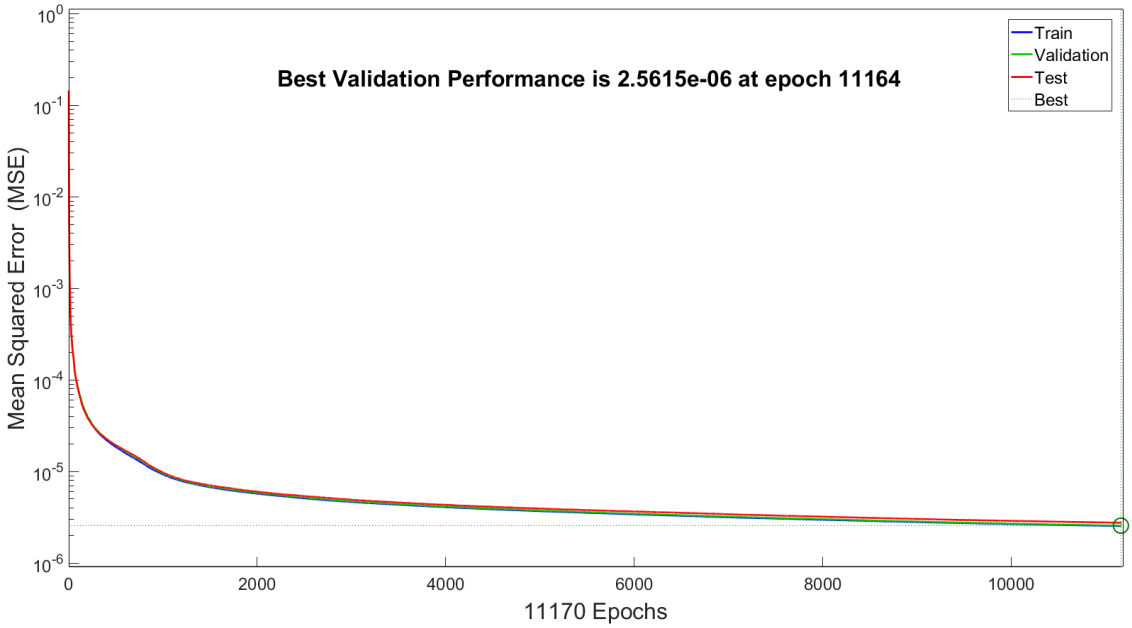
Table 9.8: Overall Performance of ANNs to Estimate Reactive Power Losses

Activation Function	MSE Performance	No. of Epochs	CPU Time (s)
compet	0.022163332681796	31	**1.191**
elliotsig	0.000072225119239	15791	575.027
hardlim	0.009264731152065	551	19.709
hardlims	0.009241042213464	1190	42.366
logsig	0.000067354969881	15228	617.391
netinv	0.519720192592982	157	6.114
poslin	0.000065834355253	9520	361.258
purelin	0.001097059279541	334	11.079
radbas	0.000039444855919	20763	834.299
radbasn	0.000048681163452	9891	670.766
satlin	0.000068356876743	22524	917.911
satlins	0.000069817455129	15062	612.351
softmax	**0.000038224888516**	21362	1455.092
tansig	0.000046850840343	**29697**	1230.219
tribas	0.000047278460221	14131	579.745
Best	radbasn	tansig	compet

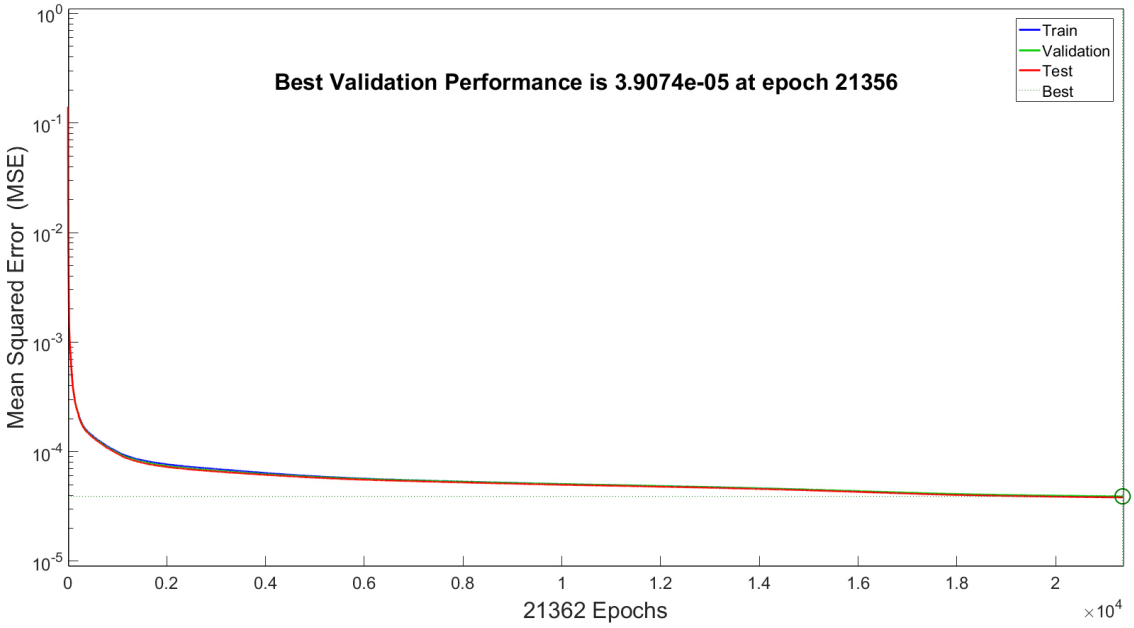
be clearly seen from that table, the `softmax` activation function scores the first position among all the other 14 functions in terms of MSE. More than that, although the activation functions `poslin`, `logsig`, and `tansig` consume more epochs in their training process, they all fail to beat `softmax`. On the opposite side, `compet` is the fastest activation function. It required less than one second to complete the training, but it has the worst MSE.

Similar thing, Table 9.8 shows the overall performance of the reactive power; i.e., the performance of Figure 9.18b.

From the 3rd column of both tables, it is obvious that no one of these 15 activation functions reaches the maximum epochs limit, which is 100,000 as can be seen from the preceding initialization settings. This very large limit is set to avoid stopping the learning process of ANNs if there is still a chance to learn more. Thus, with this large limit, ANNs can be trained freely without any restriction on their epochs



(a) Active Power Estimator



(b) Reactive Power Estimator

Figure 9.19: MSE Performance of the `softmax`-Based Power Loss Estimator

until reaching their saturated performance where one of the early stopping criteria is triggered. During recording the performance of both Tables 9.7 and 9.8, it has been observed that the learning process of each activation functions is terminated due to exceeding the maximum allowable limit of validation failures. As said before, this early stopping criterion is not relaxed to avoid having overfitted models.

The MSE performance of the `softmax` activation function for both power estimators are graphically shown in Figure 9.19. From both subplots, it is obvious that the three curves (i.e., the train, validation, and test curves) are close to each other with very small MSE, which means that the models can provide highly accurate estimations.

These `softmax`-based neural networks are supposed to do the tasks illustrated in Figure 9.17. To validate that, seven random samples are selected. The input variables of these samples are given in Table 9.9. As can be clearly seen from the results tabulated in Tables 9.10-9.11, the differences between the actual readings calculated by the **Newton-Raphson power flow** (**NRPF**) method and the estimated readings calculated by the proposed instruments-free power estimators (IFPEs) are very small and highly acceptable. Figure 9.20a shows 500 samples of the estimated and actual active power produced by the slack unit. Beside all these promising results, this IFPE technique has the following distinct features: 1. very fast estimations, 2. no need to know bus voltage magnitudes and phase angles, 3. applicable to any operational and/or topological changes, 4. losses in all branches can be estimated, and 5. P_L and Q_L are not functions of $P_{G,1}$ and $Q_{G,1}$ anymore. The last one is a very important feature, which can be effectively utilized to solve one of the stiff technical problems[13] associated with ELD, UC, OPF, and other power system operation studies.

The last important point is about the performance of this IFPE technique as compared with classical inefficient methods, such as Kron's loss formula given in (3.18). The direct answer is that our proposed technique is immune against all the serious assumptions taken in Kron's loss formula, which are listed in [123]. The IFPE technique can be used with any topological or operational change, and it ensures getting highly precise estimations. The second thing is that our technique can also estimate the losses in all the branches without referring to the output of the slack

[13] i.e., the active and reactive power balance equality constraints.

Table 9.9: Sample of Seven Random Input Variables Taken from the Input Matrices X_1 and X_2

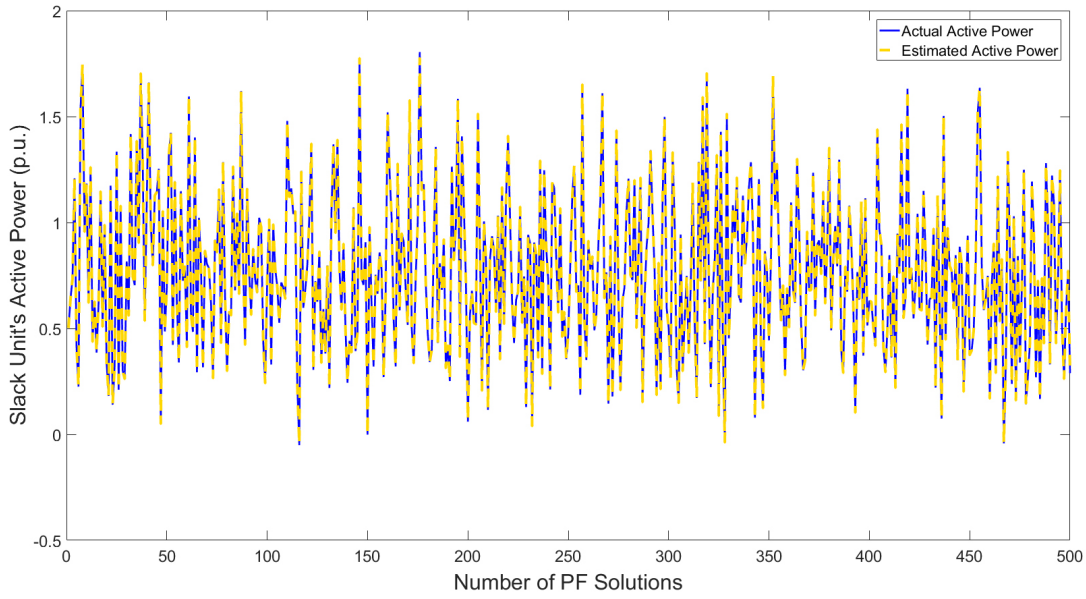
	Unit 2		**Unit 3**		**Load 1**		**Load 2**		**Load 3**		**Line Status**					
Sample	$P_{G,2}$	$Q_{G,2}$	$P_{G,3}$	$Q_{G,3}$	$P_{D,5}$	$Q_{D,5}$	$P_{D,6}$	$Q_{D,6}$	$P_{D,8}$	$Q_{D,8}$	$\mathfrak{L}_{45}$	$\mathfrak{L}_{46}$	$\mathfrak{L}_{57}$	$\mathfrak{L}_{69}$	$\mathfrak{L}_{78}$	$\mathfrak{L}_{89}$
1	2.0358	0.3676	0.7955	0.2205	-1.4248	-0.5699	-0.7905	-0.2635	-0.9482	-0.3319	1	1	1	1	0	1
2	1.7247	0.2511	0.9749	0.1480	-1.4063	-0.5625	-0.7021	-0.2340	-1.1772	-0.4120	1	1	0	1	1	1
3	1.6999	0.3250	0.9952	0.0860	-1.3371	-0.5348	-1.1233	-0.3744	-1.0688	-0.3741	1	1	1	0	1	1
4	1.8571	0.3084	1.0235	0.2077	-1.1973	-0.4789	-0.8117	-0.2706	-1.2159	-0.4256	1	1	0	1	1	1
5	1.7270	0.2295	0.7733	0.1340	-1.4118	-0.5647	-0.8706	-0.2902	-1.0740	-0.3759	1	1	0	1	1	1
6	1.9948	0.3592	0.9872	0.0959	-1.0243	-0.4097	-0.8170	-0.2723	-1.1417	-0.3996	1	1	1	0	1	1
7	1.8950	0.3325	0.9395	0.0585	-1.1924	-0.4770	-0.7717	-0.2572	-0.9392	-0.3287	1	1	1	0	1	1

Table 9.10: Sample of Seven Random Active Output Variables Taken from the 1st Output Matrix U_1

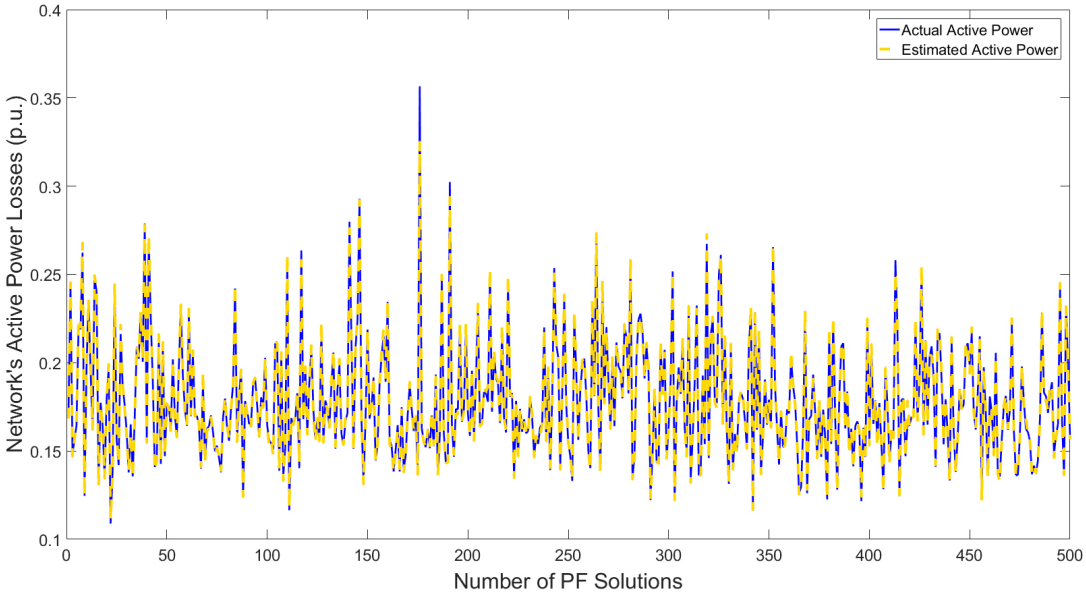
Sample	**Slack Unit**		**Network Losses**		**Branch Losses**											
	$P_{G,1}$		P_L		$P_{L,45}$		$P_{L,46}$		$P_{L,57}$		$P_{L,69}$		$P_{L,78}$		$P_{L,89}$	
	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE
1	0.503	0.502	0.170	0.168	0.009	0.009	0.016	0.015	0.132	0.132	0.002	0.001	0.000	0.000	0.012	0.012
2	0.734	0.734	0.148	0.147	0.025	0.024	0.010	0.009	0.000	-0.001	0.084	0.085	0.025	0.023	0.004	0.006
3	0.959	0.956	0.125	0.125	0.003	0.003	0.026	0.025	0.083	0.082	0.000	0.003	0.000	-0.001	0.012	0.012
4	0.509	0.508	0.165	0.165	0.018	0.017	0.012	0.011	0.000	0.000	0.101	0.101	0.029	0.027	0.005	0.008
5	0.990	0.988	0.134	0.135	0.025	0.026	0.005	0.005	0.000	-0.002	0.074	0.075	0.025	0.025	0.005	0.005
6	0.140	0.140	0.139	0.139	0.008	0.008	0.013	0.013	0.106	0.107	0.000	0.001	0.000	0.001	0.011	0.009
7	0.211	0.210	0.142	0.143	0.007	0.008	0.011	0.012	0.113	0.113	0.000	0.000	0.000	0.000	0.010	0.010

Table 9.11: Sample of Seven Random Reactive Output Variables Taken from the 2nd Output Matrix U_2

Sample	Slack Unit		Network Losses		Branch Losses															
	$P_{G,1}$		P_L		$Q_{L,14}$		$Q_{L,27}$		$Q_{L,39}$		$Q_{L,45}$		$Q_{L,46}$		$Q_{L,57}$		$Q_{L,69}$		$Q_{L,78}$		$Q_{L,89}$	
	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE	NRPF	IFPE
1	0.632	0.635	0.055	0.054	0.035	0.032	0.255	0.259	0.038	0.041	0.058	0.058	0.086	0.081	0.665	0.667	0.008	0.012	0.000	0.000	0.101	0.098
2	0.882	0.884	0.073	0.075	0.070	0.076	0.181	0.180	0.054	0.053	0.171	0.168	0.055	0.056	0.000	-0.002	0.376	0.378	0.167	0.169	0.034	0.031
3	0.907	0.897	0.035	0.022	0.093	0.091	0.178	0.188	0.056	0.049	0.022	0.019	0.143	0.145	0.419	0.426	0.000	-0.002	0.002	0.000	0.098	0.088
4	0.823	0.825	0.164	0.171	0.050	0.053	0.211	0.212	0.061	0.061	0.120	0.124	0.063	0.067	0.000	-0.002	0.451	0.451	0.195	0.197	0.046	0.042
5	0.879	0.877	0.012	0.009	0.093	0.092	0.181	0.180	0.034	0.035	0.173	0.169	0.027	0.027	0.000	-0.002	0.329	0.328	0.167	0.168	0.046	0.046
6	0.727	0.724	0.101	0.103	0.029	0.024	0.244	0.240	0.055	0.056	0.057	0.061	0.070	0.069	0.532	0.533	0.000	0.001	0.003	0.003	0.096	0.101
7	0.757	0.755	0.085	0.082	0.033	0.035	0.220	0.222	0.049	0.049	0.051	0.051	0.062	0.062	0.568	0.563	0.000	0.002	0.001	0.001	0.086	0.086



(a) Slack Active Power Generation



(b) Network Active Power Losses

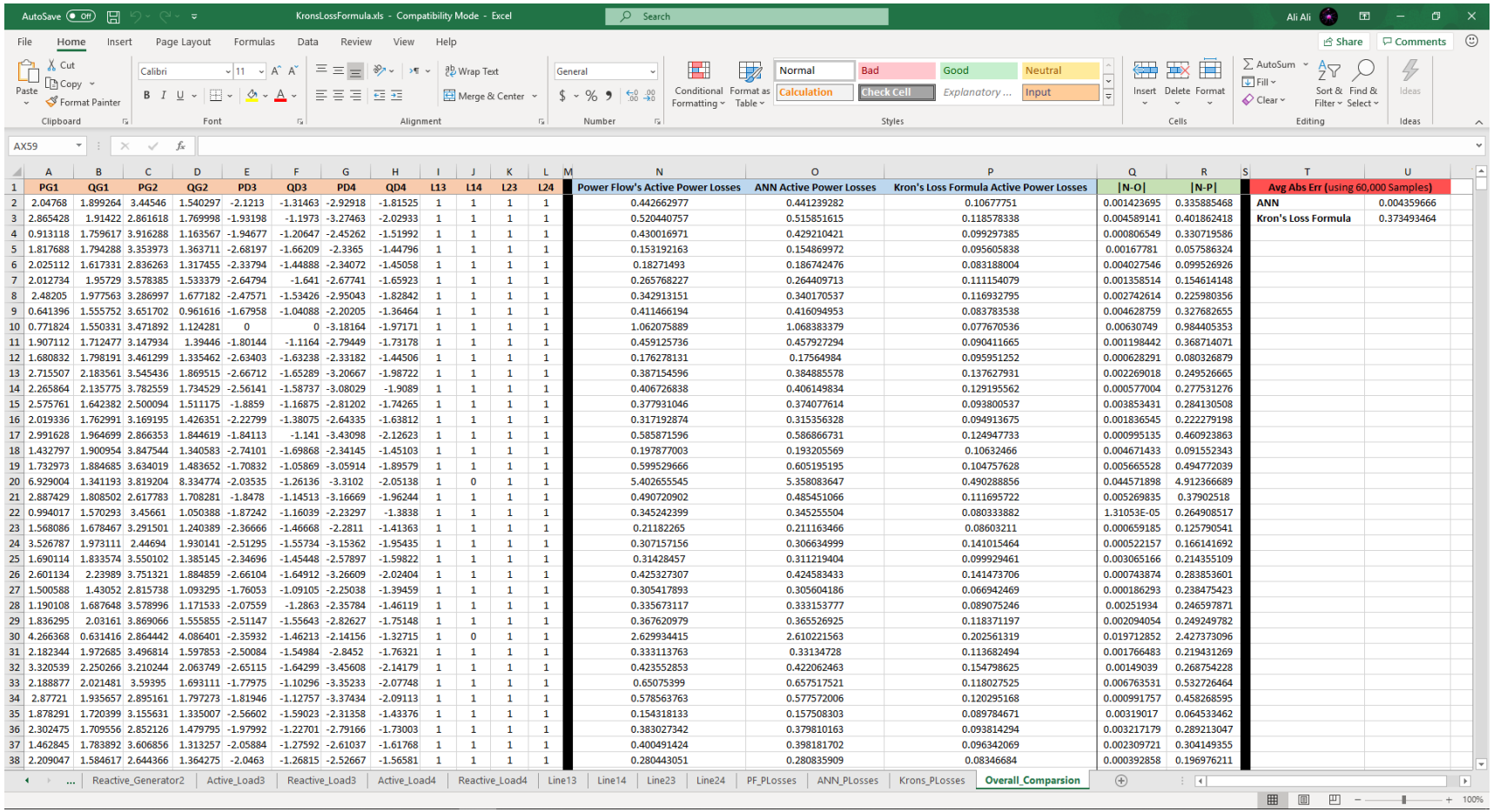
Figure 9.20: Active Power Generation and Losses of the Slack Unit and Network of the WSCC 9-Bus Test System

Figure 9.21: The Grainger-Stevenson's 4-Bus Test System

unit. To prove the first claim, which is about the deviation in the solution quality when the preceding assumptions are broken, Grainger-Stevenson's 4-bus test system given in [155] is considered, which is shown in Figure 9.21. The accurate power loss calculation of NRPF is compared with those estimated by our IFPE technique and Kron's loss formula. The performance comparison conducted in MS Excel is shown in Figure 9.22 where all the readings are in per-unit quantities. From the PF solutions of 60,000 random configurations, it is clear that our estimation method is much superior to Kron's loss formula. The latter fails to estimate some conditions when a topological or an operational change is not close to the initial state of the network. For example, the 20th row shows that the absolute difference between NRPF and IFPE is just 0.04457 p.u. while the absolute difference between NRPF and Kron's loss formula is 4.9124 p.u. Actually, there are some worse conditions than that! For example, the maximum absolute difference recorded for Kron's loss formula is 7.9036 p.u., which is totally unacceptable. Thus, again, the IFPE technique proves itself as a highly competitive power loss estimator that can be used in future applications of modern electric power systems.

9.2 Optimization-Free Economic Load Dispatcher

Solving economic load dispatch (ELD) problems correctly and efficiently is a common concerning issue faced by many electric power companies and research labs around the world. The terms "correct" and "efficient" can be translated as "practically feasible" and "optimal", respectively. Based on the literature, many studies have



PG1	QG1	PG2	QG2	PD3	QD3	PD4	QD4	L13	L14	L23	L24	Power Flow's Active Power Losses	ANN Active Power Losses	Kron's Loss Formula Active Power Losses	|N-O|	|N-P|	Avg Abs Err (using 60,000 Samples)	
2.04768	1.899264	3.44546	1.540297	-2.1213	-1.31463	-2.92918	-1.81525	1	1	1	1	0.442662977	0.441239282	0.10677751	0.001423695	0.335885468	ANN	0.004359666
2.865428	1.91422	2.861618	1.769998	-1.93198	-1.1973	-3.27463	-2.02933	1	1	1	1	0.520440757	0.515851615	0.118578338	0.004589141	0.401862418	Kron's Loss Formula	0.373493464
0.913118	1.759617	3.916288	1.163567	-1.94677	-1.20647	-2.45262	-1.51992	1	1	1	1	0.430016971	0.429210421	0.099297385	0.000806549	0.330719586		
1.817688	1.794288	3.353973	1.363711	-2.68197	-1.66209	-2.3365	-1.44796	1	1	1	1	0.153192163	0.154869972	0.095605838	0.00167781	0.057586324		
2.025112	1.617331	2.836263	1.317455	-2.33794	-1.44888	-2.34072	-1.45058	1	1	1	1	0.18271493	0.186742476	0.083188004	0.004027546	0.099526926		
2.012734	1.95729	3.578385	1.533379	-2.64794	-1.641	-2.67741	-1.65923	1	1	1	1	0.265768227	0.264409713	0.111154079	0.001358514	0.154614148		
2.48205	1.977563	3.286997	1.677182	-2.47571	-1.53426	-2.95043	-1.82842	1	1	1	1	0.342913151	0.340170537	0.116932795	0.002742614	0.225980356		
0.641396	1.555752	3.651702	0.961616	-1.67958	-1.04088	-2.20205	-1.36464	1	1	1	1	0.411466194	0.416094953	0.083783538	0.004628759	0.327682655		
0.771824	1.550331	3.471892	1.124281	0	0	-3.18164	-1.97171	1	1	1	1	1.062075889	1.068383379	0.077670536	0.00630749	0.984405353		
1.907112	1.712477	3.147934	1.39446	-1.80144	-1.1164	-2.79449	-1.73178	1	1	1	1	0.459125736	0.457927294	0.090411665	0.001198442	0.368714071		
1.680832	1.798191	3.461299	1.335462	-2.63403	-1.63238	-2.33182	-1.44506	1	1	1	1	0.176278131	0.17564984	0.095951252	0.000628291	0.080326879		
2.715507	2.183561	3.545436	1.869515	-2.66712	-1.65289	-3.20667	-1.98722	1	1	1	1	0.387154596	0.384885578	0.137627931	0.002269018	0.249526665		
2.265864	2.135775	3.782559	1.734529	-2.56141	-1.58737	-3.08029	-1.9089	1	1	1	1	0.406726838	0.406149834	0.129195562	0.000577004	0.277531276		
2.575761	1.642382	2.500094	1.511175	-1.8859	-1.16875	-2.81202	-1.74265	1	1	1	1	0.377931046	0.374077614	0.093800537	0.003853431	0.284130508		
2.019336	1.762991	3.169195	1.426351	-2.22799	-1.38075	-2.64335	-1.63812	1	1	1	1	0.317192874	0.315356328	0.094913675	0.001836545	0.222279198		
2.991628	1.964699	2.866353	1.844619	-1.84113	-1.141	-3.43098	-2.12623	1	1	1	1	0.585871596	0.586866731	0.124947733	0.000995135	0.460923863		
1.432797	1.900954	3.847544	1.340583	-2.74101	-1.69868	-2.34145	-1.45103	1	1	1	1	0.197877003	0.193205569	0.10632466	0.004671433	0.091552343		
1.732973	1.884685	3.634019	1.483652	-1.70832	-1.05869	-3.05914	-1.89579	1	1	1	1	0.599529666	0.605195195	0.104757628	0.005665528	0.494772039		
6.929004	1.341193	3.819204	8.334774	-2.03535	-1.26136	-3.3102	-2.05138	1	0	1	1	5.402655545	5.358083647	0.490288856	0.044571898	4.912366689		
2.887429	1.808502	2.617783	1.708281	-1.8478	-1.14513	-3.16669	-1.96244	1	1	1	1	0.490720902	0.485451066	0.111695722	0.005269835	0.37902518		
0.994017	1.570293	3.45661	1.050388	-1.87242	-1.16039	-2.23297	-1.3838	1	1	1	1	0.345242399	0.345255504	0.080333882	1.31053E-05	0.264908517		
1.568086	1.678467	3.291501	1.240389	-2.36666	-1.46668	-2.2811	-1.41363	1	1	1	1	0.21182265	0.211163466	0.08603211	0.000659185	0.125790541		
3.526787	1.973111	2.44694	1.930141	-2.51295	-1.55734	-3.15362	-1.95435	1	1	1	1	0.307157156	0.306634999	0.141015464	0.000522157	0.166141692		
1.690114	1.833574	3.550102	1.385145	-2.34696	-1.45448	-2.57897	-1.59822	1	1	1	1	0.31428457	0.311219404	0.099929461	0.003065166	0.214355109		
2.601134	2.23989	3.751321	1.884859	-2.66104	-1.64912	-3.26609	-2.02404	1	1	1	1	0.425327307	0.424583433	0.141473706	0.000743874	0.283853601		
1.500588	1.43052	2.815738	1.093295	-1.76053	-1.09105	-2.25038	-1.39459	1	1	1	1	0.305417893	0.305604186	0.066942469	0.000186293	0.238475423		
1.190108	1.687648	3.578996	1.171533	-2.07559	-1.2863	-2.35784	-1.46119	1	1	1	1	0.335673117	0.333153777	0.089075246	0.00251934	0.246597871		
1.836295	2.03161	3.869066	1.555855	-2.51147	-1.55643	-2.82627	-1.75148	1	1	1	1	0.367620979	0.365526925	0.118371197	0.002094054	0.249249782		
4.266368	0.631416	2.864442	4.086401	-2.35932	-1.46213	-2.14156	-1.32715	1	0	1	1	2.629934415	2.610221563	0.202561319	0.019712852	2.427373096		
2.182344	1.972685	3.496814	1.597853	-2.50084	-1.54984	-2.8452	-1.76321	1	1	1	1	0.333113763	0.33134728	0.113682494	0.001766483	0.219431269		
3.320539	2.250266	3.210244	2.063749	-2.65115	-1.64299	-3.45608	-2.14179	1	1	1	1	0.423552853	0.422062463	0.154798625	0.00149039	0.268754228		
2.188877	2.021481	3.59395	1.693111	-1.77975	-1.10296	-3.35233	-2.07748	1	1	1	1	0.65075399	0.657517521	0.118027525	0.006763531	0.532726464		
2.87721	1.935657	2.895161	1.797273	-1.81946	-1.12757	-3.37434	-2.09113	1	1	1	1	0.578563763	0.577572006	0.120295168	0.000991757	0.458268595		
1.878291	1.720399	3.155631	1.335007	-2.56602	-1.59023	-2.31358	-1.43376	1	1	1	1	0.154318133	0.157508303	0.089784671	0.00319017	0.064533462		
2.302475	1.709556	2.852126	1.479795	-1.97992	-1.22701	-2.79166	-1.73003	1	1	1	1	0.383027342	0.379810163	0.093814294	0.003217179	0.289213047		
1.462845	1.783892	3.606856	1.313257	-2.05884	-1.27592	-2.61037	-1.61768	1	1	1	1	0.400491424	0.398181702	0.096342069	0.002309721	0.304149355		
2.209047	1.584617	2.644366	1.364275	-2.0463	-1.26815	-2.52667	-1.56581	1	1	1	1	0.280443051	0.280835909	0.08346684	0.000392858	0.196976211		

Figure 9.22: The Power Loss Estimation Performance of IFPE and Kron's Loss Formula For Different Operational and Topological Changes on Grainger-Stevenson's 4-Bus Test System

been presented as ELD solvers, which can be divided into two main streams[14]:

1. Classical or traditional optimization techniques.

2. Modern or non-traditional optimization techniques.

Regardless of which method is the best, which is also a dialectical or controversial question, all these methods are built based on the same concept where the objective function(s) is minimized with satisfying all the equality, inequality, and side constraints. The main objective function is to find the optimal schedule that sets all the generating units at the most economical operating cost. This can be achieved by minimizing the total price predicted by regression-based fuel-cost functions. If all the design constraints are satisfied without any violation, then the ELD solution can satisfy the power demand with the maximum earning from that operation. Also, if the emission rates of NOx, SOx, COx, Soot, and unburned hydrocarbons (UHC) are considered as a second objective that needs to be minimized, then the ELD problem is translated as a multi-objective optimization problem; which is out of the scope of this study.

Returning to the solutions obtained by existing ELD optimizers presented in the literature. Many questions can be raised about the feasibility and optimality of these solutions because the gap between real electric operations and their mathematical models always exists as a mixture of uncertainties [7]. Also, there is a hidden fact that can be realized if someone works in real electric power stations. The operation department crew increases or decreases the set-points of generators in fixed steps (i.e., discrete settings) after being received from **automation centers**. That is, if these generators cannot be adjusted with continuous settings, then the solutions provided by existing ELD optimizers can be considered as infeasible or impractical solutions[15]. Based on this, some special types of optimization algorithms called **combinatorial optimization algorithms** (**COAs**) could be used to deal with this discrete search space [345]. However, if the given ELD problem has a limited number of units, then all the possible solutions can be obtained. This can be easily done by using a **sliced**

[14]These two main streams will be further categorized in the next section.

[15]It is exactly similar to optimal coordination problems of electromechanical directional overcurrent relays (DOCRs) where their plug and time multiplier settings are available in discrete forms, and thus the optimal solutions obtained for state-of-the-art numerical relays are infeasible for old relays [280].

fuel-cost curve (**SFCC**) for each unit and then applying a **pairwise** or **element-by-element summation** to create a full dataset of the given ELD problem [9].

At first glance, this method is technically not practical with many generating units, because the search space increases exponentially as the number of units or/and slicing resolution increases. This argument is correct if it is seen from the traditional view. But, in real-world applications, this challenge happens if each generating unit has its own busbar and all the units are owned by the same power company; which is incorrect and rarely happens [7].

In this study, a new method is proposed to resolve the existing ELD problems by considering one real fact that the feasibility of an optimal solution is achieved if the generating units are scheduled with discrete set-points. This method is designed based on the **brute-force search** concept, but with some essential modifications inspired by real operations of power stations. The drawbacks of this **exhaustive search method** can be converted to benefits. Some numerical experiments are given to describe how this innovative method works and its main strengths and weaknesses.

The main contributions achieved in this study are:

1. A new **optimization-free** ELD solver is introduced where the obtained solutions are optimal and feasible from the real power stations point of view.

2. Its main drawback concentrated in *high*-dimensional ELD problems is solved by re-structuring the classical ELD model so that the power production of multiple groups of units is fed to the grid via some common busbars.

3. It shows how each power station can have a map of full possible/feasible solutions. These maps could be used in many future studies, including hybridization with **optimization-based** ELD solvers to boost their exploitation and exploration levels.

9.2.1 Mechanism of the Proposed Technique

The classical ELD models presented in Chapter 3 have been extensively solved using very strong classical and advanced optimization algorithms. However, the solutions provided by those algorithms come with continuous values, while many real power stations deal with discrete power settings received from their automation centers. To

clarify this point, Figure 9.23 illustrates the general working principle of an electric power system. The automation center (or system control) carries out all the necessary calculations of power system studies, and based on the final results it sends specific commands and instructions to power stations to feed some common busbars with their optimal power [7]. Based on our long observations in MCRs of different power stations, we have not seen any ELD command contains very precise settings, such as $P_i = 83.212345$ MW. Even, if they tried to achieve that value it is very hard, because of hardware limitations and/or uncertainties due to disturbances and ignorable parameters/variables. Instead, the commands are given in practical discrete values, such as $P_i = 83$ or 84 MW. Based on this simple but very critical fact, the solutions offered by all existing ELD optimizers are impractical and could be infeasible or/and shifted away from the optimal point. To overcome this challenge, two approaches can be offered here:

1. Upgrading existing optimization algorithms to act as COAs [345], or
2. Using SFCC to model all n units, so the operating cost C_i is represented as a dependent vector of n P_i vectors. Then, finding all the possible solutions through a pairwise or element-by-element summation of these vectors. This technique is well known as "**generate and test search**", "brute-force search", or "exhaustive search".

At first glance, COAs are preferable if the problem dimension or/and slicing resolution is high. The resolution part can be dropped in this study because the discrete values are proposed to be integers, so the minimum step-size of any P_i is $\Delta P = P_{i+1} - P_i = 1$ MW.

The preceding argument is correct if the ELD problem is formulated based on the classical mathematical models presented in Chapter 3. But, if the configurations of real power stations described in [7] are taken into account when using the second approach (i.e., SFCC technique), then the previous dimensionality problem is solvable if it is seen from that practical perspective; as will be seen later.

9.2.1.1 Combinatorial Optimization Algorithms

This approach is not the core of this study, and thus it is used here just to give an idea about the effectiveness of the proposed SFCC method. The existing continuous

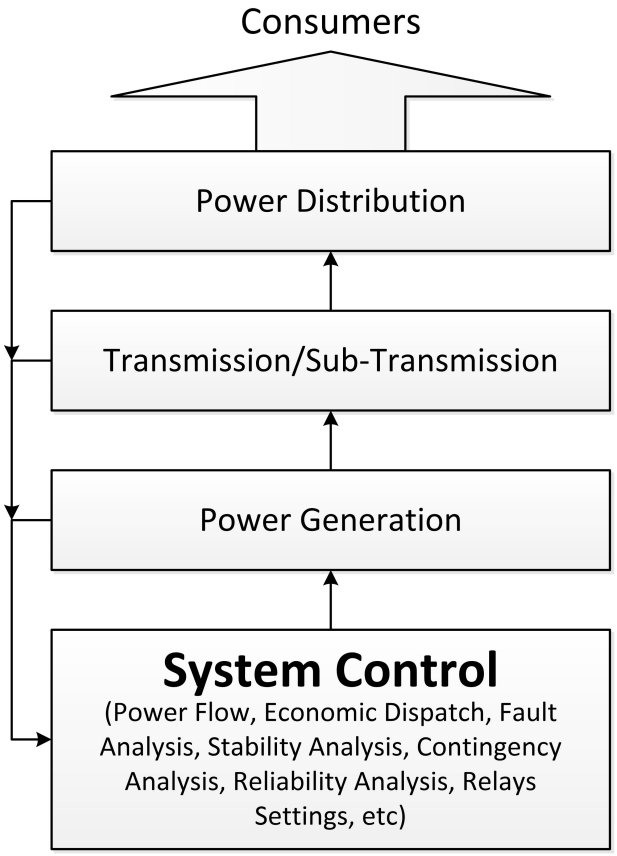


Figure 9.23: Flowchart of a General Electric Power System

optimization algorithms can be easily upgraded to their combinatorial versions by some tricky codes. For example, in MATLAB, the command `randi` can be used to randomly select one element of each P_i vector. The ith vector is defined as follows:

$$P_i = \left[0, P_i^{\min}, P_{i,1}, P_{i,2}, \cdots, P_i^{\max}\right] \tag{9.35}$$

where 0 means that the ith unit is totally OFF. The terms $P_{i,1}$ and $P_{i,2}$ respectively mean $P_i^{\min}$ plus one step and two steps (i.e., $P_{i,1} = P_i^{\min} + \Delta P$ and $P_{i,2} = P_i^{\min} + 2\Delta P$). For this mission, the hybrid algorithm presented in [30] is used here to act as a COA, but after disabling the SQP stage; i.e. to have the same algorithm presented in [31].

9.2.1.2 SFCC Technique

To be able to apply this technique, the dependent vector or operating cost of (9.35) should be obtained as follows:

$$C_i = \left[C_i(0), C_i(P_i^{\min}), C_i(P_{i,1}), \cdots, C_i(P_i^{\max})\right] \tag{9.36}$$

To describe its mechanism, let's take the following simple 2-unit ELD example:

$$C_1 = 253 + 7.57P_1 + 0.05538P_1^2 \tag{9.37}$$

$$C_2 = 362 + 7.78P_2 + 0.04968P_2^2 \tag{9.38}$$

where $P_1 \in [60, 75]$ MW, $P_2 \in [100, 115]$ MW, $P_D = 170$ MW, and the slicing resolution between P_i and P_{i+1} is 5 MW. Thus, (9.35) and (9.36) respectively become:

- For Unit No.1:

$$P_1 = [0, 60, 65, 70, 75]$$
$$C_1 = [253.0, 906.6, 979.0, 1054.3, 1132.3]$$

- For Unit No.2:

$$P_2 = [0, 100, 105, 110, 115]$$
$$C_2 = [362.0, 1636.8, 1726.6, 1818.9, 1913.7]$$

If the pairwise summation is applied to each two vectors of P and C, then the following matrices are attained:

$$P_T = \begin{bmatrix} 0 & 100 & 105 & 110 & 115 \\ 60 & 160 & 165 & \mathbf{170} & 175 \\ 65 & 165 & \mathbf{170} & 175 & 180 \\ 70 & \mathbf{170} & 175 & 180 & 185 \\ 75 & 175 & 180 & 185 & 190 \end{bmatrix}$$

$$C_T = \begin{bmatrix} 615.0 & 1889.8 & 1979.6 & 2071.9 & 2166.7 \\ 1268.6 & 2543.4 & 2633.2 & \mathbf{2725.5} & 2820.3 \\ 1341.0 & 2615.8 & \mathbf{2705.7} & 2798.0 & 2892.7 \\ 1416.3 & \mathbf{2691.1} & 2780.9 & 2873.2 & 2968.0 \\ 1494.3 & 2769.1 & 2858.9 & 2951.2 & 3046.0 \end{bmatrix}$$

These two matrices can be generated by using outer- and inner-`for` loops. The other quick way is to use the **binary singleton expansion function** (i.e., using `bsxfun` in MATLAB with `plus` operator) to effectively accelerate the computational speed. The other advanced alternative is to use **multithreading** and **parallel computing**.

From the preceding two matrices, it is obvious that there are three possible ELD solutions to satisfy $P_D = 170$ MW. The best solution is the one that has the cheapest fuel-cost, which is $C_T(4, 2) = 2691.1$ \$/hr and occurs when $\{P_1, P_2\} = \{70, 100\}$ MW.

Therefore, the main stages of this optimization-free ELD solver are shown in Figure 9.24. As can be clearly seen from this flowchart, creating the full dataset in the first stage consumes considerable CPU time. This is why it is sometimes called an exhaustive or brute-force search. However, this stage is initiated only one time to collect all the possible solutions from the discrete search space. Once this dataset is created, then any other exact optimal solution can be easily and quickly obtained without referring to that stage anymore. That is, the brute-force search is executed just to have something like a look-up table, which is repeatedly recalled to know the updated best optimal settings of all generating units based on the current load demand P_D.

9.2.2 Numerical Experiments and Discussion

In this experiment, two test systems are taken. The first one is an example of a *low*-dimensional ELD problem, while the other is considered as a *high*-dimensional ELD problem. For both test systems, a new combinatorial version of the MpBBO algorithm, described in Chapter 2, is used. Because the first test system has just three units, so the proposed optimization-free SFCC technique is also applied to solve it. For the second test system, an illustration is given to practically solve it by the SFCC technique without facing the computational problem mentioned in the initial stage of Figure 9.24.

All these simulations are coded in MATLAB 2016a using ALIENWARE M14x machine, 64-bit Windows 10 OS, Intel Core i7-4700MQ CPU - 2.4 GHz, and 16 GB RAM. The initialization parameters of MpBBO are similar to those presented in [30] where the population size and the total number of generations are respectively set to

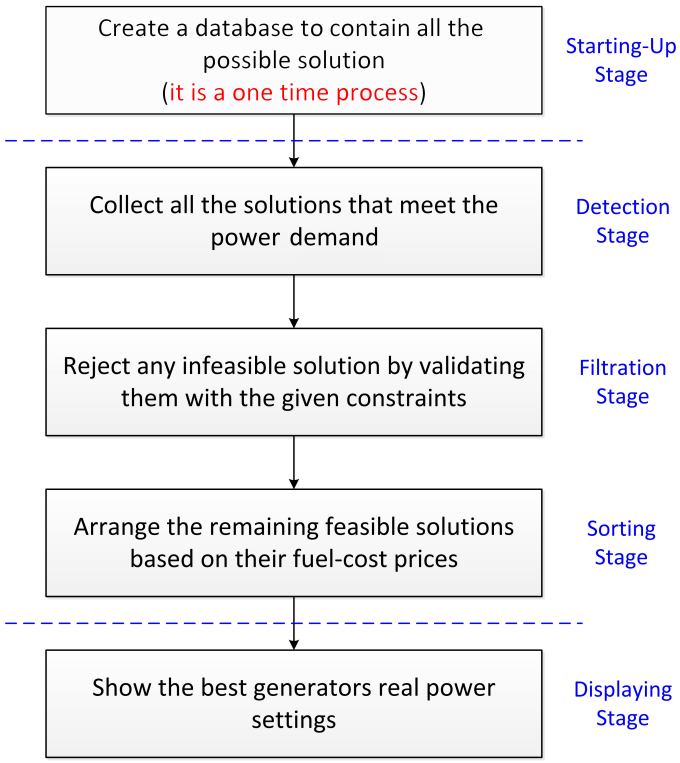


Figure 9.24: Flowchart of the SFCC Technique

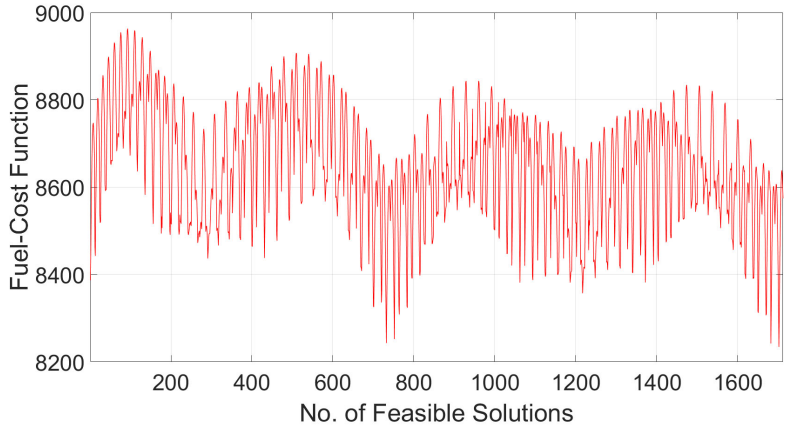
$\{20, 50\}$ for the 3-unit problem, and $\{20, 250\}$ for the 40-unit problem. The detailed information about these two test systems can be found in [28].

9.2.2.1 Test Case I - 3 Generating Units

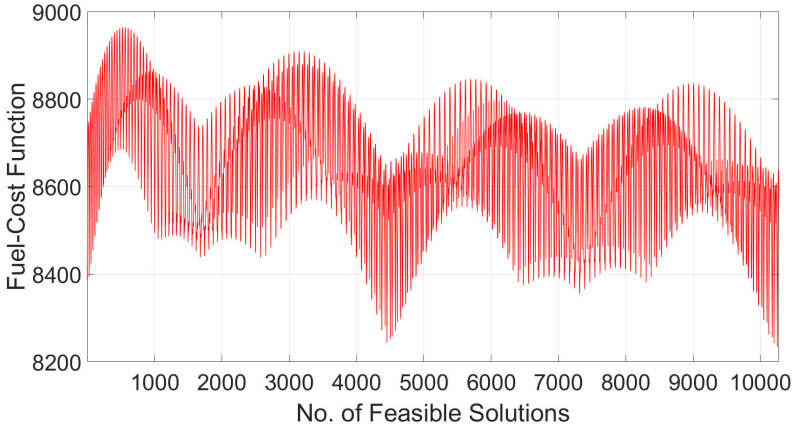
The objective is to minimize the total fuel-cost while the load demand of 850 MW is attained. The results obtained by the proposed optimization-free SFCC algorithm and the combinatorial MpBBO algorithm, using the exponential cooling strategy with a cooling factor of $c = 0.6$ for the simulated annealing (SA) stage[16], are tabulated in Table 9.12. For SFCC, three different slicing resolutions are used.

As can be clearly seen from the preceding table, SFCC can provide the exact optimal solution to this *low*-dimensional discrete ELD problem. Although the combinatorial MpBBO algorithm succeeded to reach the exact optimal solution, it achieved that after consuming 50 trials where the mean, median, and standard deviation values indicate the possibility that the optimization algorithm could trap into near-optimal

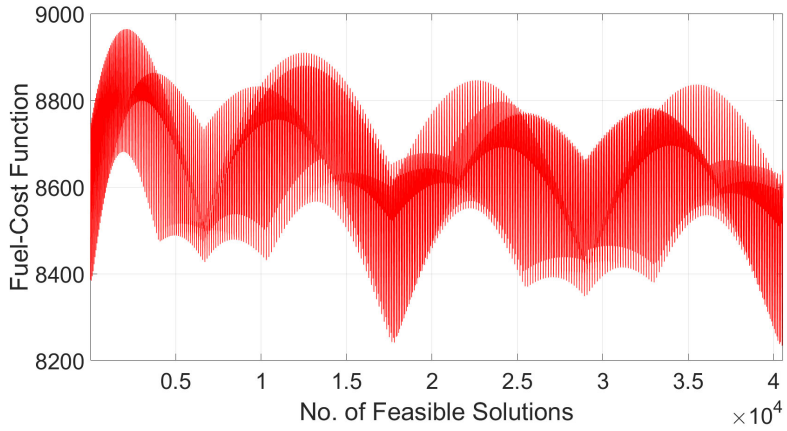
[16]Please, refer to (2.68).



(a) $SR = 5$ MW



(b) $SR = 2$ MW



(c) $SR = 1$ MW

Figure 9.25: Effect of the Slicing Resolution on the Number of Solutions of SFCC

Table 9.12: Simulation Results of the 3-Unit ELD Problem

Unit	**MpBBO**	**SFCC Technique**		
No.	(SR[a]$= 1$)	($SR = 5$)	($SR = 2$)	($SR = 1$)
P_1 **(MW)**	300	300	300	300
P_2 **(MW)**	150	150	150	150
P_3 **(MW)**	400	400	400	400
$\sum P_i$ **(MW)**	850	850	850	850
Best Cost ($/hr)	**8234.221**	**8234.221**	**8234.221**	**8234.221**
Mean ($/hr)	8258.136	**8234.221**	**8234.221**	**8234.221**
Median ($/hr)	8254.956	**8234.221**	**8234.221**	**8234.221**
StDev ($/hr)	23.176	**0**	**0**	**0**
Avg. CPU Time (s)	0.061113	**0.021917**	0.453040	6.837385

[a] SR: Slicing Resolution, in MW.

Table 9.13: The First 10 Best Solutions Obtained by SFCC with a Slicing Resolution of 1 MW for the 3-Unit ELD Problem

Unit No.	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**
P_1 **(MW)**	300	301	300	499	400	499	500	299	399	400
P_2 **(MW)**	150	150	151	100	50	101	100	151	51	51
P_3 **(MW)**	400	399	399	251	400	250	250	400	400	399
$\sum P_i$ **(MW)**	850	850	850	850	850	850	850	850	850	850
Best Cost ($/hr)	8234.221	8238.074	8238.629	8241.448	8242.160	8242.612	8243.161	8243.585	8245.211	8245.610

solutions. As the slicing resolution increases (i.e., from 5 MW to 1 MW step-size), the number of feasible solutions increases, and thus the algorithm requires more CPU time. Apparently, MpBBO is faster than SFCC with a slicing resolution of less than 5 MW. But, this could not be true with *high*-dimensional ELD problems because COAs require more generations and a bigger population size.

The other interesting things about the proposed SFCC technique can be observed when looking at the full maps of these feasible solutions shown in Figure 9.25. From these plots, it can be obviously concluded that the total number of feasible solutions is proportional to the slicing resolution. In addition to the exact optimal solution, the SFCC technique detects all the other good solutions as well. For example, Table 9.13 shows the first 10 best solutions obtained by SFCC with $SR = 1$ MW. This unique

feature can be practically utilized if some generating units are planned to be taken out of service for major/minor maintenance or if any one of these units exceeds the maximum EOH limit[17,18] [188]. Also, these three plots give an initial idea of the expected optimal location(s). These ELD maps could be hybridized with classical ELD optimizers to focus only on the suspected optimal location(s) of the search space. Thus, more pressure can be applied to explore and exploit these spots, which could accelerate the processing speed and save a significant amount of memory.

9.2.2.2 Test Case II - 40 Generating Units

This test system is known as one of the *high*-dimensional ELD problems available in the literature. The load demand that needs to be satisfied, without any violation of the design constraints, is 10500 MW. Again, the initialization parameters of this test system are taken from [30]; when the SQP stage is disabled. Also, the detailed information about this test system is given in [28, 348].

As said before, *high*-dimensional discrete problems can be easily solved by using COAs. Table 9.14 shows the solutions obtained by MpBBO for this 40-unit ELD problem. The performance is investigated using four different slicing resolutions.

Table 9.14 gives us the reason why COAs are preferred to solve optimization problems having finite sets of objects. From that result, the CPU time is independent of the slicing resolution and it is almost constant in all the cases. However, the optimal solution is affected by that resolution because the optimization algorithm requires more generations as the length of the finite set increases.

The SFCC technique can also be applied to solve *high*-dimensional ELD problems, but it is highly unrecommended to use without some essential modifications; as will be seen later. The reason behind this limitation comes from the size of the dataset in the 1st stage of Figure 9.24 because the discrete search space increases exponentially with the incremental increase in the problem dimension or/and step-size resolution. Thus, for the 40-unit ELD problem, the size of the SFCC dataset is extremely big. Instead, the next paragraph shows our innovative scheme to apply SFCC in any real *high*-dimensional ELD problem.

[17] Please, refer to Chapter 7.

[18] If there is no choice except to use it, then an additional penalty cost should be applied to this weary machine.

Table 9.14: Simulation Results Obtained by MpBBO for the 40-Unit ELD Problem

Unit No.	$SR = 5$	$SR = 2$	$SR = 1$	$SR = 0.1$
P_1 **(MW)**	106	110	73	111.3
P_2 **(MW)**	111	114	112	112.5
P_3 **(MW)**	95	60	115	100.9
P_4 **(MW)**	180	188	170	80.2
P_5 **(MW)**	87	95	93	83.9
P_6 **(MW)**	73	102	140	139
P_7 **(MW)**	285	236	299	256.1
P_8 **(MW)**	235	297	299	286.4
P_9 **(MW)**	290	291	290	296.4
P_{10} **(MW)**	130	142	142	141.5
P_{11} **(MW)**	229	166	150	248
P_{12} **(MW)**	234	240	164	161.6
P_{13} **(MW)**	133	214	200	193.2
P_{14} **(MW)**	295	401	396	398.5
P_{15} **(MW)**	400	395	411	368.8
P_{16} **(MW)**	480	321	470	390.2
P_{17} **(MW)**	485	494	399	492.7
P_{18} **(MW)**	500	486	475	480.4
P_{19} **(MW)**	522	510	505	513.5
P_{20} **(MW)**	437	486	513	536.5
P_{21} **(MW)**	539	490	506	517.2
P_{22} **(MW)**	524	540	524	527.2
P_{23} **(MW)**	524	524	524	516.3
P_{24} **(MW)**	529	530	538	523.9
P_{25} **(MW)**	544	526	532	535
P_{26} **(MW)**	534	532	544	501.8
P_{27} **(MW)**	15	14	10	11.8
P_{28} **(MW)**	15	14	15	18.7
P_{29} **(MW)**	15	34	21	14
P_{30} **(MW)**	97	89	87	88.5
P_{31} **(MW)**	175	184	190	175.2
P_{32} **(MW)**	170	146	190	185.2
P_{33} **(MW)**	185	190	167	175.4
P_{34} **(MW)**	195	170	172	174.1
P_{35} **(MW)**	180	170	200	189.4
P_{36} **(MW)**	175	180	166	164.1
P_{37} **(MW)**	85	105	92	105.7
P_{38} **(MW)**	85	97	110	69.9
P_{39} **(MW)**	100	107	75	68.6
P_{40} **(MW)**	507	510	421	546.4
$\sum P_i$ **(MW)**	10500	10500	10500	10500
Best Cost ($/hr)	124691.898	124640.675	124743.860	124751.397
Avg. CPU time (s)	1.496571	1.498222	1.500169	1.530221

9.2.2.3 How to Apply SFCC in Real *High*-Dimensional ELD Problems

Although SFCC has many distinct features, it cannot be applied in *high*-dimensional ELD problems. This is why nobody thinks about it in solving numerical engineering problems. However, if this technique is addressed from the practical perspective, then the technical problem associated with the database of Figure 9.24 can be easily solved.

In real electric power networks, the generated power is supplied from some power stations, wind farms, **solar farms**, and other **renewable energy sources**. Thus, common busbars are used to feed that power to the gird as shown[19] in Figure 9.26. To accomplish this, we dismantle the ELD solver into sub-solvers. The first one is a **global ELD solver**, which should be applied to determine the configurations of the w power stations shown in that figure. For each xth power station, there is at least one **local ELD solver**, which should be applied to determine the settings of the k units connected to the same busbar. That is, the xth power station has k units and these units need to be economically dispatched. Therefore, from this real point of view, any *high*-dimensional ELD problem can be practically split into the preceding two stages. The first one has only one w-dimensional ELD problem that needs to be solved by the global ELD solver. Once the economical settings of these w power stations are obtained from their automation center, there are w local ELD problems with k variables for each power station[20]. Thus, there are w local ELD solvers that should be used to find the economical way to generate power, and that power should equal to the sum of the net power supplied by the w power stations minus the power losses in the network. Because the k units of the xth power station are very close to each other, so the power losses can be neglected in the local ELD solvers. That is, the global ELD solver is responsible to minimize the power losses by choosing the most economical settings for these w power stations[21]. More details about this real configuration are given in [7], which will also be covered in the next section of this chapter. Therefore, SFCC can be easily applied to both the global ELD solver and

[19]The same concept has been covered in Figures 7.11-7.12 shown in Chapter 7.

[20]Some power stations have more than one common busbar.

[21]This global ELD solver is also responsible to find the cheapest configurations of these power stations. Moreover, the objective of emission rate limits could be integrated into advanced global ELD solvers.

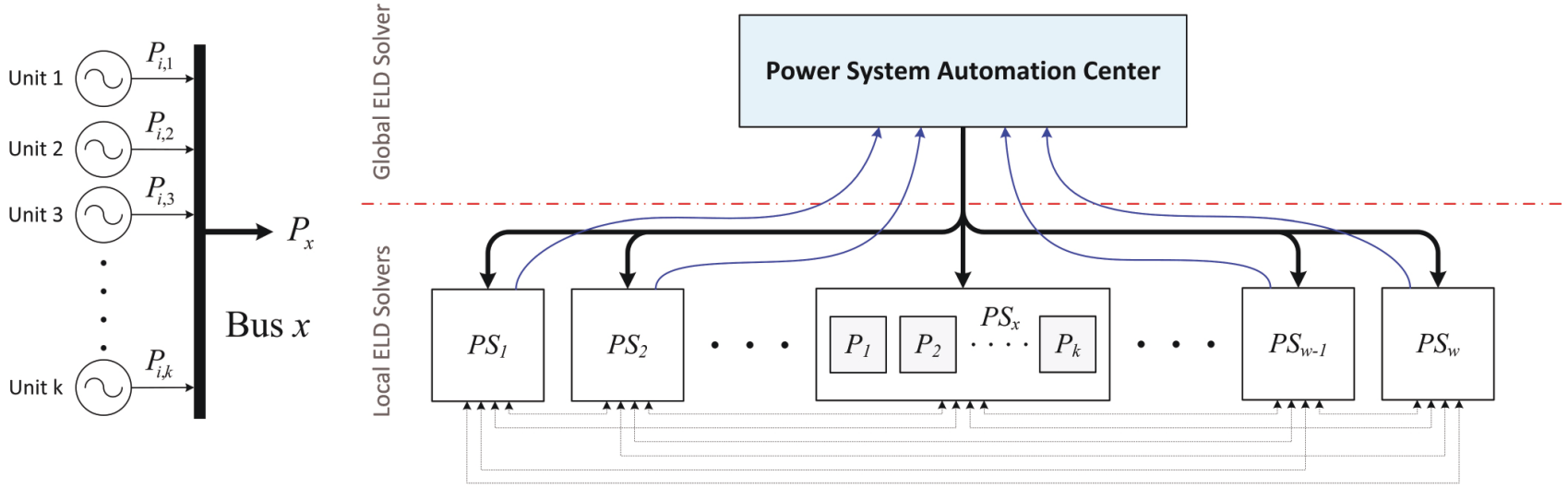


Figure 9.26: Simplified Arrangement of Global-Local ELD Solvers for a Real Electric Power System

the w local ELD solvers because the problem dimensions of all these solvers are now small.

The realized structure shown in Figure 9.26 reveals a serious hidden fact that for w power stations, having common busbars, there are at least w equality constraints that need to be satisfied. This serious fact will be extensively covered in the other innovative ELD technique given next.

9.2.2.4 How to Apply SFCC even with Continuous Power Settings

Someone may ask: *Can SFCC be applied if the power set-points are provided in continuous instead of discrete?* Yes, there is one possible way to do that. The procedure is as follows:

1. Round the power set-points to the nearest discrete values based on the slicing resolution.

2. Solve the discretized ELD problem by using the SFCC technique.

3. The remaining amount of the continuous power demand can be covered from some distributed energy storage elements or small DGs.

This amount of power is supposed to be very small, and in many cases, it could be even less than 1 MW. Thus, the drift from the optimal point is almost ignorable. However, this claim has not been proven yet.

9.3 Optimization/Modeling-Free Economic Load Dispatcher

In Chapter 3, many traditional and modern numerical techniques have been proposed to solve ELD problems, which are briefly summarized in Section 9.2. These numerical techniques can be further categorized into four groups as follows [4, 7, 30, 63, 85, 123, 130, 141, 155, 165, 175, 202, 217, 243, 287, 298, 326, 386]:

1. Traditional or classical optimization algorithms: the lambda-iteration method, the base point and participation factors method, the gradient method, the Newton-Raphson method, and linear/nonlinear programming methods.

2. Non-traditional or modern optimization algorithms: genetic algorithm (GA), evolutionary programming (EP), differential evolution (DE), simulated annealing (SA), tabu search (TS), particle swarm optimization (PSO), ant-colony optimization (ACO), artificial bee colony (ABC), bacterial foraging optimization (BFO), biogeography-based optimization (BBO), etc. They can be divided into two sub-groups:

   - Single-solution algorithms: such as SA and TS.
   - Population-based algorithms: such as GA, PSO, DE, BBO, ACO, ABC, and BBO.

   It has to be said that this group is labeled with different names, such as stochastic, meta-heuristic, nature-inspired, and evolutionary algorithms (EAs). The last name can be considered as part of nature-inspired algorithms [7, 122, 345].

3. AI-based algorithms: such as involving artificial neural networks (ANNs), support vector machines (SVMs), fuzzy systems (FS), etc.

4. Hybrid optimization algorithms: this group can also be divided into the following sub-groups:

   - Hybridization only between traditional algorithms of the first group.
   - Hybridization only between non-traditional algorithms of the second group.
   - Hybridization only between AI-based algorithms of the 3rd group.
   - Hybridization between traditional and non-traditional algorithms of the first and second groups.
   - Hybridization between traditional and AI-based algorithms of the first and third groups.
   - Hybridization between non-traditional and AI-based algorithms of the second and third groups.
   - Hybridization between traditional, non-traditional, and AI-based algorithms of the first three groups.
   - Multi-stage hybridizations, such as our MpBBO-SQP reported in [30].

In the literature, it is well known that ELD problems are highly constrained, nonlinear, and non-convex, and thus it is very hard to minimize them. This is why sometimes classical optimization methods fail to find the global, or at least near-global, optimal solutions without violating any equality, inequality, or side constraint.

All the preceding approaches share the following core principle: they are theoretical-based approaches. Thus, if someone wants to apply them to solve a real power system problem, he/she will realize that there are many technical challenges that need to be satisfied before being able to formulate its ELD optimization problem in the EMS software. The first challenge that could be faced is the mathematical model itself! *Is it built based on some assumptions or not? Does it have a precise objective function? Do the fuel-cost functions explain the exact behaviors of the generating units? Do the modeled generating units match their real responses? Do the listed constraints cover all the aspects of the given power system? Are there any hidden or unconsidered equality/inequality constraints? What about the uncertainties due to the fuzziness, vagueness, ambiguity, and subjective judgments of operators and designers? etc.* Actually, based on these real facts, there is a highlighted doubt about the optimality and feasibility of the current solutions obtained by all the known ELD solvers presented in the literature.

Moreover, speed and memory usage of the algorithm can also create another set of challenges, because each EMS has non-upgradable hardware that should be shared by many other modules (such as PF, FA, SE, CA, and ORC). Thus, system operators are forced to suppress some features of modern optimization algorithms (such as population size, maximum iterations limit, and hybridization mode) in order to be able to design an adaptive ELD solver.

In addition, the EMS software itself may become hard to use, especially for inexperienced operators. Many times, only the OEMs or third party providers can accomplish these technical tasks within the installed EMS software under an expensive periodic contract. The other alternative is by offering high paying jobs to employ some specialists.

Furthermore, EMS could be not fully licensed. It is similar to any other engineering software where each package (i.e., power flow, fault analysis, etc) needs an additional installation cost. Add to that, the ELD package could not be available in

Table 9.15: Total Number of Power Station Configurations for Each Archiving Type

Archiving	**Manually by Manpower**			**Automatically by EMS**	
Period	**Daily**	**3 Shifts / Day**	**Hourly**	**Every Minute**	**Every Second**
Daily	1	3	24	1440	86400
Weekly	7	21	168	10080	604800
Monthly (≈ 30)	30	90	720	43200	2592000
Annually (≈ 365)	365	1095	8760	525600	31536000

some basic EMS software, such as those installed in very old systems in some developing countries where only supervisory control and data acquisition (SCADA) system or distributed control system (DCS) is used to safely automate electric components without considering the economic and/or environmental aspects.

All these restrictions make applying ELD strategy very hard, and many power stations have been seen operated without considering this strategy at all.

In most, or maybe all, power stations there is one common routine job carried out by operators and monitored by plant managers, heads, or, at least, operation senior shift-charge engineers. This routine job is simply about recording the most important real input and output readings. This routine job could be done once per day, operation manpower shift, or hourly. If modern DCS and SCADA systems are used, then more data can be automatically saved in archiving servers. Thus, with that built-in feature to automatically save the measured values in archiving servers, the existing automation system (i.e., DCS or SCADA) can save all these values within an updatable window of one minute, one second, or even less. This means that a very huge real data can be easily extracted from this process. Table 9.15 shows the total power station configurations that can be extracted from each case.

Some of these real input readings are: ambient temperature, atmospheric pressure, relative humidity, air flow-rate, gas consumption, gas pressure, heat-rate, compressor inlet temperature, compressor discharge pressure, compressor intake filter differential pressure, compressor vibration, compressor metal temperature, combustion chamber efficiency, **turbine inlet temperature** (**TIT**), **temperature after turbine** (**TAT**), turbine vibration, turbine metal temperature, generator vibration, generator metal temperature, lube oil temperature, etc. Also, the generated real and reactive

power, generator current and voltage, power factor, efficiency, turbine speed, emission rates, auxiliary power consumption, etc, can be considered as output readings.

The study reveals a totally different approach to achieve the ELD strategy within any real power station with almost zero efforts from operators. It simply utilizes the available real records stored in operation **logbooks** or EMS **archiving servers** to obtain the best actual solution of the ELD problem. It is completely different than all the known analytical and numerical approaches presented in the literature because it does not need to express any mathematical model nor applying any optimization method. Thus, it is called an **optimization/modeling-free economic load dispatch** (**OMF-ELD**) technique.

The main contributions achieved in this study are summarized in the following points:

1. Proposing a novel technique to obtain the best actual ELD solution by utilizing the available input and output records.

2. This technique can be easily implemented even if the ELD package is not available or unlicensed within the installed EMS software.

3. No need to express any mathematical model or to use any optimization technique, which in turn makes it a very fast algorithm compared with all the known ELD solvers.

4. No need to find any parameter or to satisfy any constraint since all the records are taken from real operations where each plant configuration is translated as one feasible solution.

5. This OMF-ELD solver has many distinct properties and features. For example:

    - It can be carried out in very simple software (such as MS Excel or even in free and open-source programs).
    - It does not require expert engineers to deal with it.
    - It is compatible with all the known types of electric power stations (thermal, hyrdo, nuclear, etc).

9.3.1 Mechanism of the Proposed Technique

From the mathematical formulation given in Chapter 3, many considerations have to be taken into account during designing classical economic load dispatchers. Practically, achieving all these aspects is not an easy task; especially if there is insufficient technical support from OEMs, knowledgable consultants, or if the commissioning manuals and other documents are completely or partially lost. Add to that, administrative staffs of most power stations reject doing any online training on their EMS software without direct supervision from OEMs, particularly during the winter season (in cold countries) or the summer season (in hot countries) where the energy consumption rates are at the highest levels. Instead, this novel OMF-ELD technique can be used to obtain the best actual solution based on the available real data recorded in operation logbooks or/and archiving server(s).

To know how the OMF-ELD technique works, consider Figure 9.23. It explains the master control flow of a general electric power system. From that flowchart, the power system control (or automation center) is responsible to instruct each power station to supply the grid with a specific amount of power (P_x). Thus, the network losses P_L can be minimized and the system power balance constraint can be satisfied. For **non-monopoly electricity markets**[22], the system control will not care whether the xth power station (PS_x) generates its power P_x at an optimal cost or not, because the first one buys that power as per contract agreement. Thus, in this case, each power station is responsible to solve its ELD problem. The stages are given in the next paragraphs. The first one shows how the best actual economic operation of the xth power station can be extracted using the OMF-ELD solver. The second stage gives a general methodology about how to use the MOF-ELD solver within the system control to minimize the total power losses in the network.

9.3.1.1 Stage No.1: OMF-ELD for One Individual Power Station

To clarify the concept of this approach, suppose that the xth power station contains k_x units as shown in Figure 9.27. The mathematical formulation of the classical ELD model is given in Chapter 3. However, Chapter 7 shows many limitations of that model. One of these limitations is that each ith unit is connected to one busbar.

[22] i.e., many providers serve the market demand

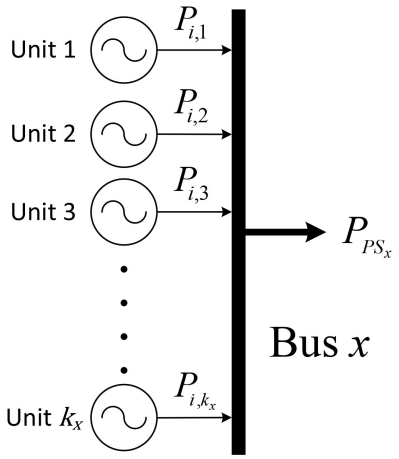


Figure 9.27: Total Electricity Production of the xth Power Station

But, in real-world applications, the generated power of many power stations are supplied to their grids through some common busbars, such as the xth busbar shown in Figure 9.27. That is, the power produced by all the k_x units of the xth power station (P_{PS_x}) is fed to the grid through the same busbar (Bus x). Therefore, in this special case, P_i should be modified to be a summation of the k_x units as given in (7.25)[23]:

$$P_i = -P_{i,\mathrm{aux}} + \sum_{j=1}^{k_x} P_{i,j} = P_{\mathrm{PS}_x} \tag{9.39}$$

Similar to Subsection 7.4.1, there are at least w new equality constraints that must be added to any ELD optimization problem:

$$P_{\mathrm{PS}_1} = P_{1,1} + P_{1,2} + \cdots + P_{1,j} + \cdots + P_{1,k_1} \tag{9.40}$$

$$P_{\mathrm{PS}_2} = P_{2,1} + P_{2,2} + \cdots + P_{2,j} + \cdots + P_{2,k_2} \tag{9.41}$$

$$\vdots$$

$$P_{\mathrm{PS}_x} = P_{x,1} + P_{x,2} + \cdots + P_{x,j} + \cdots + P_{x,k_x} \tag{9.42}$$

$$\vdots$$

$$P_{\mathrm{PS}_w} = P_{w,1} + P_{w,2} + \cdots + P_{w,j} + \cdots + P_{w,k_w} \tag{9.43}$$

[23]Some references deal with this special case by converting the fuel-cost functions of all the units to an equivalent unit. However, this simplification is inapplicable if it is applied in real-world problems where many technical considerations should be taken into account. Some of these facts are given in Chapter 7 and in this section as well.

This means that the existing ELD solvers cannot be used without this time-consuming essential modification. In this case, two sub-stages of economic dispatch are needed to improve prospective solutions. The first sub-stage is accomplished by finding the optimal setting of each ith unit, while the second sub-stage is activated if $k \neq 1$. Thus, the second sub-stage is achieved by finding the optimal configuration of all the units connected to the xth busbar so that P_{PS_x} is produced at the lowest possible operating cost. This is one of the reasons why there is a gap between the mathematical-based ELD solvers and the actual optimal solution.

The OMF-ELD technique works in a different way where all the above issues are not faced at all. The idea behind OMF-ELD technique is simply defined as how to effectively utilize the real dataset stored in the archives to extract the best actual optimal solutions of these power stations. The readings of each operating configuration (such as date, gas consumption, heat-rate, individual/overall power output, auxiliary power consumption, emission rates, ambient temperature, and relative humidity) can be recorded either manually (by entering the data from the operation logbook) or (automatically via EMS connected to one or multiple archiving servers of DCS or SCADA). Some of the possible recording frames have been already shown in Table 9.15. Figure 7.4 shows a real output taken from **ProControl-14 DCS** (manufactured by ABB) for one of the gas turbines (GTs) used in this study.

These real records are feasible and viable, and they contain a huge amount of useful information. They can be used to find the best-achieved configuration that meets the end-users' power consumption at the lowest recorded production cost. The mechanism of this technique is described through the flowchart shown in Figure 9.28 and Algorithm 14.

This pseudocode is for the simplified algorithm where the temperature, humidity, emission rates, and other less effective factors are not considered. Once the main idea is captured, any other variable can be easily inserted in the algorithm.

9.3.1.2 Stage No.2: OMF-ELD for Power System Control

With the monopoly player case where multiple power stations are owned by only one public/private power company, or if the system control cares about both the network losses and the production cost of each xth power station, then there is a global ELD

Algorithm 14 Pseudocode of the Local OMF-ELD Solver

Require: all the data → manually recorded from the daily, hourly, or per crew shift, operation logbook, or automatically recorded from the EMS server (this pseudocode is based on the daily logbook records)

1: **for** $i \leftarrow 1$ *to* T **do** {where T = number of recorded datasheets (i.e., the archiving period shown in Table 9.15)}
2: Collect the configuration dates in $\mathrm{PS}_{\mathrm{conf}}$ array
3: Collect the daily total power production in $\mathrm{PS}_{\mathrm{prod}}$ array
4: Collect the daily total auxiliary power consumption in $\mathrm{PS}_{\mathrm{aux}}$ array
5: Collect the daily total fuel cost in $\mathrm{PS}_{\mathrm{cost}}$ array
6: **end for**
7: Find the net delivered power array by $\mathrm{PS}_{\mathrm{net}} = \mathrm{PS}_{\mathrm{prod}} - \mathrm{PS}_{\mathrm{aux}}$
8: Index all the elements of $\mathrm{PS}_{\mathrm{conf}}$ and $\mathrm{PS}_{\mathrm{cost}}$ with the corresponding elements of $\mathrm{PS}_{\mathrm{net}}$
9: **for** $j \leftarrow 1$ *to* R **do** {where R = the length of $\mathrm{PS}_{\mathrm{net}}$}
10: **if** $\mathrm{PS}_{\mathrm{net}_j} = P_x$ **then** {where P_x = the xth power demand}
11: Save $\mathrm{PS}_{\mathrm{net}_j}$, $\mathrm{PS}_{\mathrm{conf}_j}$ and $\mathrm{PS}_{\mathrm{cost}_j}$ in $\mathrm{Sol}^{L}_{\mathrm{tab}}$ {where $\mathrm{Sol}^{L}_{\mathrm{tab}}$ = the local solutions table}
12: **end if**
13: **end for**
14: **for** $j \leftarrow 1$ *to* R **do** {where R = the length of $\mathrm{PS}_{\mathrm{net}}$}
15: **if** $\mathrm{PS}_{\mathrm{net}_j} > P_x$ and $\mathrm{PS}_{\mathrm{cost}_j} \leqslant \max(\mathrm{PS}_{\mathrm{cost}})$ in $\mathrm{Sol}^{L}_{\mathrm{tab}}$ **then**
16: Save $\mathrm{PS}_{\mathrm{net}_j}$, $\mathrm{PS}_{\mathrm{conf}_j}$ and $\mathrm{PS}_{\mathrm{cost}_j}$ in $\mathrm{Sol}^{L}_{\mathrm{tab}}$
17: **else**
18: Ignore
19: **end if**
20: **end for**
21: From $\mathrm{Sol}^{L}_{\mathrm{tab}}$, select the minimum value of $\mathrm{PS}_{\mathrm{cost}}$, and the corresponding elements of $\mathrm{PS}_{\mathrm{net}}$ and $\mathrm{PS}_{\mathrm{conf}}$ arrays

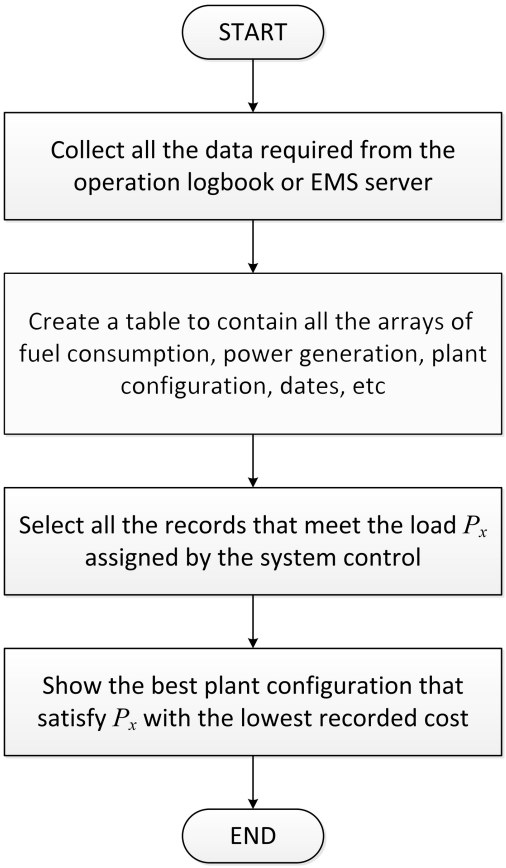


Figure 9.28: General Flowchart of the Local OMF-ELD Algorithm

solver which should be activated before carrying out the local ELD solvers for all the w power stations; which is described in Algorithm 14. The block diagram shown in Figure 9.29 clarifies this process. First, the global OMF-ELD solver is activated to compare between the system load and the past power generated from the preceding w power stations to feed the same load. The goal here is to know the amount of power that should be supplied from each xth power station with the lowest recorded losses in the network. This can be achieved by sorting all the archived configurations that match the present load. Then, each xth power station initiates its own local OMF-ELD solver to find the best actual settings of its k_x units, so P_{PS_x} can be achieved with the lowest possible operating cost. The flowchart shown in Figure 9.29 describes the mechanism of the global OMF-ELD solver, which is further explained in Algorithm 15.

Once Algorithm 15 is executed by the system control, each power station should

Algorithm 15 Pseudocode of the Global OMF-ELD Solver

Require: all the data → manually recorded from the daily, hourly, or per crew shift, operation logbook, or automatically recorded from the EMS server (this pseudocode is based on the daily logbook records)

Read all the loads connected to busbars in P_{buses} array
for $i \leftarrow 1$ *to* T **do** {where T = number of recorded datasheets (i.e., the archiving period shown in Table 9.15)}
    Collect the daily power received from w power stations in P_{NET} table (where P_{NET} = Net Power from $\{\text{PS}_1, \text{PS}_2, \cdots, \text{PS}_w\}$)
    Collect the sum of the daily power price paid to these w power stations in P_{COST} array
end for
Index all the singular values of P_{COST} array with the corresponding arrays of P_{NET} table
for $j \leftarrow 1$ *to* U **do** {where U = the length of P_{COST} array}
    if $P_{\text{NET}_j} = P_{\text{buses}}$ **then**
        Save the jth array of P_{NET} and the jth element of P_{COST} in $\text{Sol}^{G}_{\text{tab}}$ {where $\text{Sol}^{G}_{\text{tab}}$ = the global solutions table}
    end if
end for
for $j \leftarrow 1$ *to* U **do** {where U = the length of P_{COST} array}
    if $P_{\text{NET}_j} > P_{\text{buses}}$ and $P_{\text{COST}_j} \leqslant \max(P_{\text{COST}})$ in $\text{Sol}^{G}_{\text{tab}}$ **then**
        Save the array of P_{NET_j} and the element of P_{COST_j} in $\text{Sol}^{G}_{\text{tab}}$
    else
        Ignore
    end if
end for
From $\text{Sol}^{G}_{\text{tab}}$, select the minimum value of P_{COST}, and the corresponding array of PS_{NET}
Ask the respective power stations to produce a vector of power equal to PS_{NET}

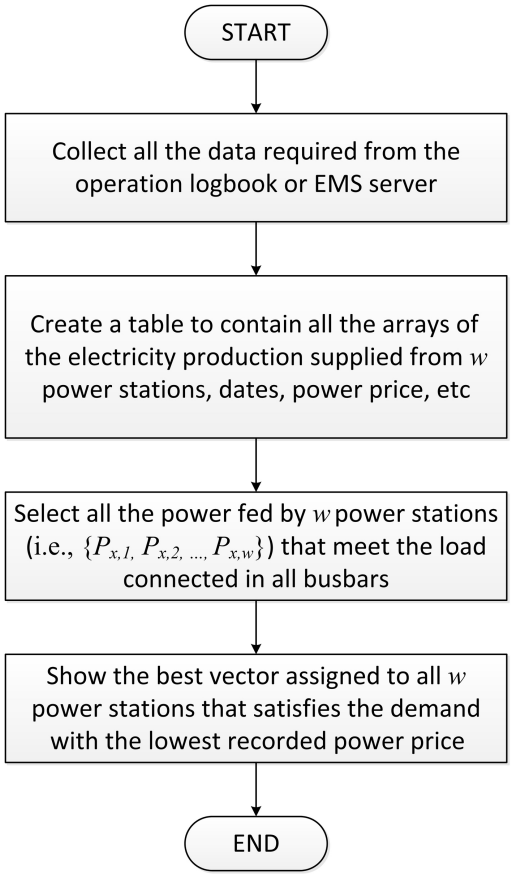


Figure 9.29: General Flowchart of the Global OMF-ELD Algorithm

execute its internal Algorithm 14 to find the best configuration to produce the required power P_{PS_x} at the lowest possible operating cost. With the monopoly player case, both Algorithm 14 and Algorithm 15 (i.e., $w+1$ filtration stages) can be carried out within the system control. Thus, the head or shift charge operation engineer of each xth power station will receive a periodic instruction, from the corresponding system control, to optimally schedule the k_x units.

Instead of using Algorithms 14 and 15, a single OMF-ELD solver can be designed to optimally schedule all the power stations and the network losses. However, this approach is inflexible and it has many drawbacks and limitations. For example:

- The system control requires full access to the data stored in all the w power stations, which is impossible if they are owned by different organizations or companies.

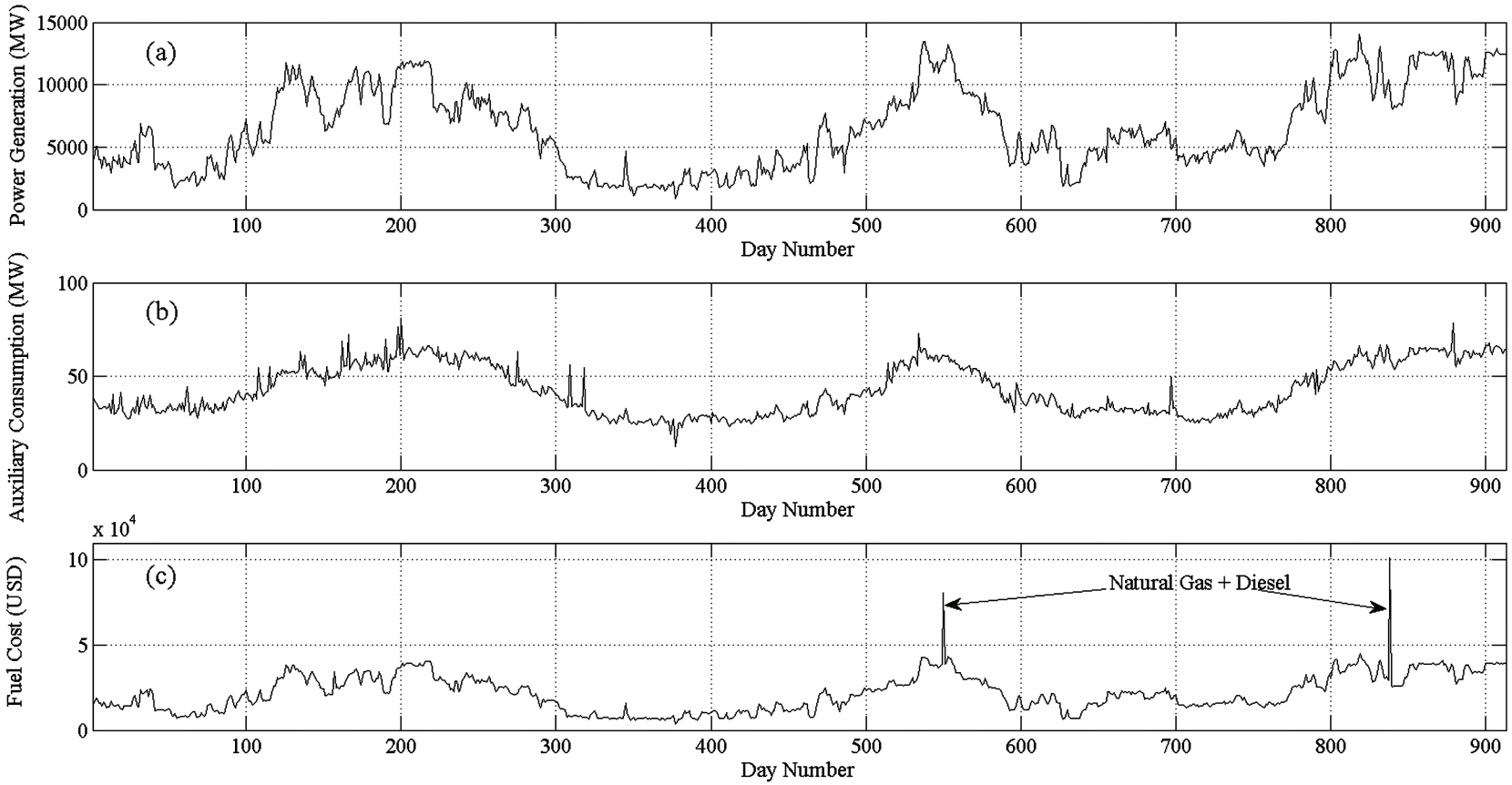


Figure 9.30: Some Data Gathered from the Operation Logbook of the Power Station Used in This Experiment. (a) Total Power Generation, (b) Auxiliary Power Consumption, and (c) Fuel Cost in US Dollar

Table 9.16: Specifications of the Power Station Used in This Experiment

	Power Plant No.1	**Power Plant No.2**
Manufacturer	Kraftwerk Union (SIEMENS)	ABB (ALSTOM)
GT Model	KWU/V723/MeiBner	GT13DM
No. of Turbines	Five (GT01-GT05)	Six (GT06-GT11)
Base-Load	5×45 MW	6×75 MW
Fuel Type	Diesel	Khuff Gas ($\mathcal{P}_G \approx 70$ bar)
	Residue Gas ($\mathcal{P}_G \approx 18$ bar)	
	Khuff Gas ($\mathcal{P}_G \approx 70$ bar)[a]	
Starting Generator	2 MVA "Diesel"	5 MVA "Diesel"

[a]Reduced down to 21 bar before being sent to the combustion chambers of GT06-GT11

- Even if these w power stations are owned by the same company (or the government), the tables mentioned in Algorithms 14 and 15 must have the same dimension to avoid many programming challenges.

- The overall structure of the program is very complicated and hard to be understood and/or traced by other programmers in case they want to modify, upgrade, or integrate it with other programs.

- The two stages approach is more flexible in case the other variables[24] are considered because the responsibility is shifted to the power stations (i.e., the local OMF-ELD solvers) to deal with these new variables. Therefore, some power stations could obtain their best actual settings by considering the limits of their emission rates, while others could consider the impacts of the ambient temperature and relative humidity on the overall efficiency.

9.3.2 Case Study and Experimental Results

To evaluate the performance of this proposed technique, it is important to use real data. For this reason, an operation logbook of a real power station is taken as a case study. That is, Algorithm 14 is used to solve this real ELD problem.

[24]i.e., emission rates, network security, temperature, etc.

Figure 9.31: Real Photograph of the Real Power Station Under Study

Figure 7.3 is a real photograph of the power station used in this experiment. This power station is built based on two **simple-cycle power plants** (**SCPPs**), which were commissioned in the seventies and eighties of the last century. The first plant contains 5 SIEMENS gas turbines with a unit baseload of 45 MW (i.e., $45 \times 5 = 225$ MW), and they can be operated by diesel, low-pressure natural gas, high-pressure natural gas, or a mixture of low- and high-pressure natural gas. The second plant contains 6 ABB gas turbines with a unit baseload of 75 MW (i.e., $75 \times 6 = 450$ MW), and they can only be operated by a high-pressure natural gas. Thus, two sources of natural gas are used for these two power plants. The diesel fuel, which is highly expensive, is used as an emergency fuel to supply the first power plant. Also, for a black-start condition, 2MVA and 5MVA diesel generators are used for the first and second power plants, respectively. These two SCPPs are shown in Figure 9.31, and their detailed specifications are tabulated in Table 9.16.

The data collected from the operation logbook of both power plants covers the power production from the 1st of January 2012 until the 31st of August 2014. The daily total power generation and auxiliary consumption recorded for that period are

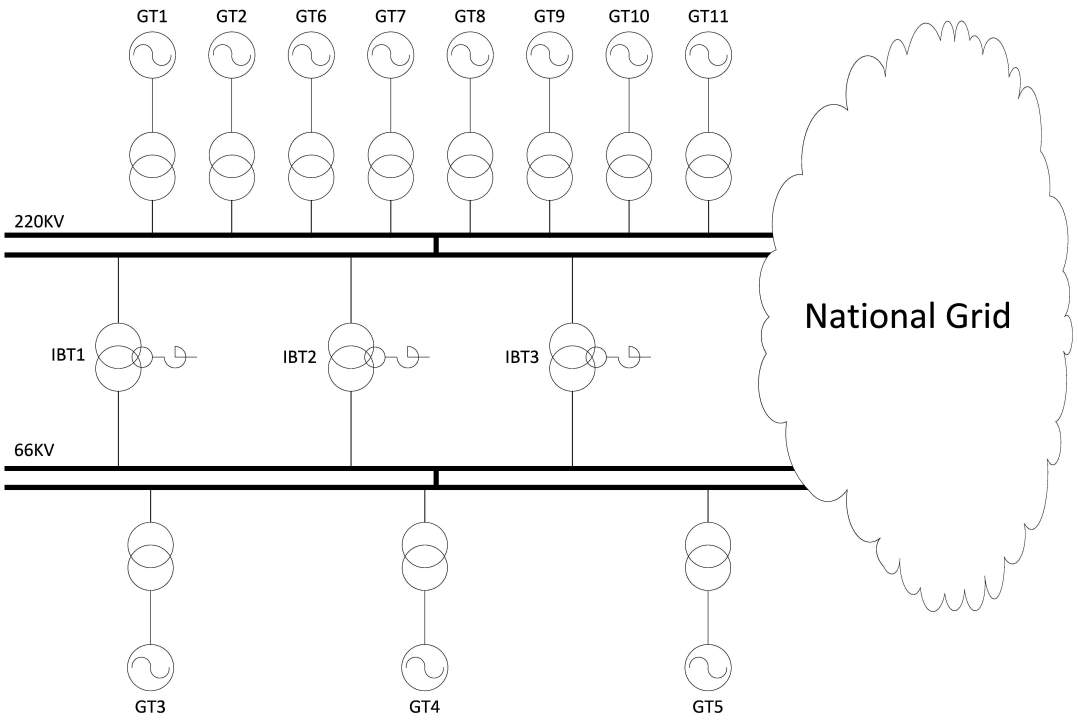


Figure 9.32: Power Station with Generators Connected to the National Grid

graphically presented in Figure 9.30a and 9.30b, respectively. The daily total operating cost for that electricity production is shown in Figure 9.30c. From the last sub-figure, it is obvious that there are two abnormal spikes. These two very high readings happened due to using a large amount of diesel as a fuel, which is a too costly option.

The electrical network of this power station (i.e., the two SCPPs) is shown in Figure 9.32. All the generating units are connected to a 220 kV transmission line, except GT3-5, of the first power plant, which are connected to a 66 kV transmission line. These two standard voltage levels are connected to each other through three inter-bus transformers (IBTs), and then connected to the national grid.

Now, suppose that the global OMF-ELD solver is used to find the optimal configuration[25] of both power plants to produce 11780 MW per a day at the lowest possible operating cost. Therefore, to economically meet that requirement, the xth

[25] i.e., the best xth element of the P_{NET} vector described in Algorithm 15.

Table 9.17: Results of OMF-ELD for the First and Second Scenarios

Scenario No.	Demand (MWd)	No. of Solutions	Worst Price ($)	Best Price ($)	Saving ($)	Date
Total Generation	11780	50	3.930705E+04	3.713548E+04	2.171569E+03	28072014
Net Generation	11780	57	4.033430E+04	3.793645E+04	2.397852E+03	29082013

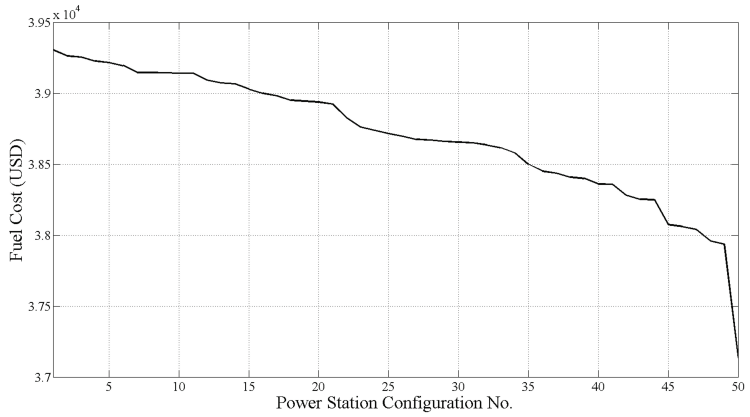
Table 9.18: Station Configuration for the First Scenario (Total Generation)

Parameters	UNIT	GT1	GT2	GT3	GT4	GT5	GT6	GT7	GT8	GT9	GT10	GT11	2MVA	5MVA	Total
Total Generation	kWd	1072266	0	1078944	0	1078736	1450000	1446000	1532000	1531000	1530000	1117000	0	0	11835946
Aux. Consumption	kWd	2520	0	2220	2640	2160	11250	4900	10330	1420	4810	3900	0	0	46150
Low Pressure Gas	m^3	277884	0	259829	262497	275836	NIL	NIL	NIL	NIL	NIL	NIL	NIL	NIL	1076046
High Pressure Gas	m^3	237886	0	222430	224713	236132	626004	615404	625578	0	613607	619995	NIL	NIL	4021749
Total Gas	m^3	515770	0	482259	487210	511968	626004	615404	625578	0	613607	619995	NIL	NIL	5097795
Diesel	m^3	0	0	0	0	0	NIL	NIL	NIL	NIL	NIL	NIL	0	0	0

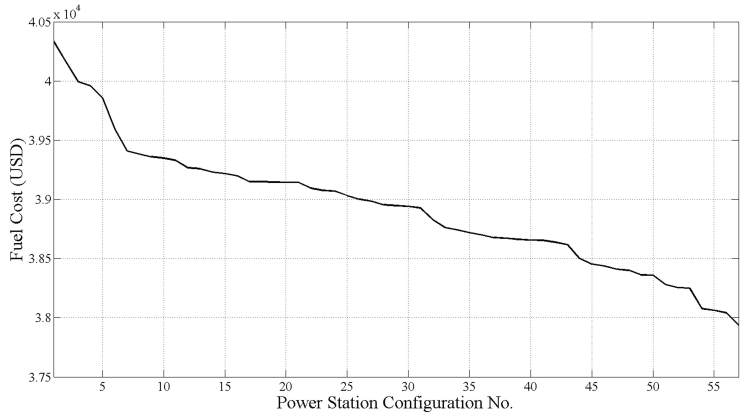
local OMF-ELD solver should be executed to search within the records stored in the operation logbook or the archiving server to estimate the best possible solution. Two scenarios are taken: finding the required power without and with subtracting the auxiliary power consumption. Such auxiliary loads are air compressors, LCRs, MCR, lightings, air conditioners, water and oil cooling towers, gas heaters, etc. The final results of both scenarios are shown in Table 9.17. As can be seen, the station can save more than 2,000 USD, daily. The configuration obtained from this local OMF-ELD solver happened in the 28th of July 2014 for the first scenario (i.e., without subtracting the station auxiliary consumption), and it happened in the 29th of August 2013 for the second scenario. The detailed information about the first and second configurations is shown in Tables 9.18 and 9.19, respectively. Also, the fitness curves of these two scenarios are respectively shown in Figures 9.33a and 9.33b.

Table 9.19: Station Configuration for the Second Scenario (Net Generation)

Parameters	UNIT	GT1	GT2	GT3	GT4	GT5	GT6	GT7	GT8	GT9	GT10	GT11	2MVA	5MVA	Total
Total Generation	kWd	1055450	0	1070957	1071544	1071120	1519300	1523200	1526700	0	1517000	1519000	0	0	11874271
Aux. Consumption	kWd	2520	0	2220	2640	2160	11250	4900	10330	1420	4810	3900	0	0	46150
Low Pressure Gas	m^3	277884	0	259829	262497	275836	NIL	NIL	NIL	NIL	NIL	NIL	NIL	NIL	1076046
High Pressure Gas	m^3	237886	0	222430	224713	236132	616394	609324	642592	620377	630049	461959	NIL	NIL	4501856
Total Gas	m^3	515770	0	482259	487210	511968	616394	609324	642592	620377	630049	461959	NIL	NIL	5577902
Diesel	m^3	0	0	0	0	0	NIL	NIL	NIL	NIL	NIL	NIL	0	0	0



(a)



(b)

Figure 9.33: Fitness Curves of OMF-ELD. (a) Scenario No.1 "Auxiliary Consumption is not Considered" and (b) Scenario No.2 "Auxiliary Consumption is Considered"

9.3.3 Discussion

From a practical point of view, many fatal issues are not covered within the objective function(s) of the classical ELD problem formulation, which might affect the solution quality. Some of these issues that should be considered are:

- The weather (including ambient temperature, humidity, air quality, etc) markedly affect the efficiency of thermal machines where the efficiency increases during winter and decreases during summer. This could be neglected if all the w units connected to the xth busbar are supposed to have a similar linear efficiency curve. But, in reality, this assumption is not valid.

- These units could be operated under high vibration, some non-working burners, some faulty thermocouples, errors on the opening of the fuel control valve, etc, which have some effects on the efficiency and the calculation of their controllers.

- The efficiency of each individual unit degrades with the time after returning the unit back from its minor or major overhaul.

These practical aspects can affect, with some tolerances, the solution quality of the ELD problem when it is solved by the known techniques presented in the literature. These aspects are considered as part of uncertainty, which could drift the solutions (obtained by the known conventional- and meta-heuristic-based optimization algorithms) away from the actual optimal point that is supposed to be found. On the opposite side, the proposed OMF-ELD technique can solve, or at least minimize, these effects; based on the size and type of the real data recorded in the operation logbook or the archiving server[26]. Some of the advantages of OMF-ELD can be addressed as follows:

- It does not require constructing an objective function or any of its parameters; $\{\alpha, \beta, \gamma, e, f\}$ in (3.3)-(3.14).

- It does not require satisfying any constraint since all the candidate solutions are practical, applicable and feasible.

- It does not require using any optimization algorithm, and hence it is a very fast technique.

- It is compatible and adaptable with all the known types of power plants.

- It does not require re-designing or re-programming the ELD solver if any new generating unit is added to the power plant.

However, each technique has its own disadvantages. Some of these disadvantages can be addressed as follows:

- The solution determined by this OMF-ELD technique is an estimate. Thus, the error is likely to exist. The amount of this deviation between the estimated

[26] i.e., it depends on the quantity and quality of the real dataset.

and exact optimal solutions depends on the size and quality of the real data recorded in the operation logbook or/and the archiving server as well as the variety of plant configurations.

- It cannot be used within new power plants where insufficient data is available yet. However, this limitation will not have any significant effect on the usefulness of this technique. The reason behind this is that when any new power plant is commissioned, then all its k generating units are supposed to be precisely modeled by their OEM, so no need to estimate the optimal plant configuration with the presence of the precise models programmed in the ELD package within its state-of-the-art EMS software. With the time, as the units degraded down, many configurations will be recorded.

- If the existing units are retrofitted[27], then the old configurations will be considered as infeasible solutions. This limitation could be ignored if the retrofit plan of the OEM is supposed to be done equally for all GTs, but this claim has not been proven yet.

The solution quality of the OMF-ELD solver could be enhanced if the planning department of the xth power station considers the following points:

- The planned overhaul of units are dispersed from each other to have a good diversity of station configurations, and hence covering the other parts of the practical search space where better configurations could be found.

- The replacement, updating, and upgrading costs of plants' equipment and systems are well monitored.

- All the daily crew cost, annual leaves, allowances, bonuses, overtime, call-outs, etc, are well recorded.

Add to that, linear and nonlinear interpolation methods could be involved here to predict new configurations that are located between some recorded configurations. To

[27]For example, ALSTOM replaced the old ABB conventional burners with its **EnVironmental** (**EV**) burners to increase the efficiency of GTs with low NOx emission rates [97].

Table 9.20: Some Predefined Points Used in (9.44)

j	1	2	3	4
P_j (MW)	40.7	43.5	48.1	49.7
$C_j\,(P_j)$ (\$)	622.7367	652.9764	702.9793	720.4659

explain how these interpolation methods work, the following pure quadratic equation - taken from [28, 337] - is proposed for the ith unit:

$$C_i\,(P_i) = 200 + 10P_i + 0.0095P_i^2 \tag{9.44}$$

As said before, the proposed OMF-ELD technique does not require using any of these equations. This mathematical model is shown here just to describe how the interpolation methods can be involved to enhance the overall performance of the OMF-ELD technique.

To clarify its mechanism, let's consider the four predefined points of $(P_{i,j}, C_{i,j})$ tabulated in Table 9.20. Now, suppose that the algorithm needs to estimate the fuel cost $C_i(P_i)$ of a non-recorded set-point at $P_{i,0} = 46.3$ MW. The following two interpolation methods are discussed just to give a general idea. The other more advanced methods could also be used in future work.

9.3.3.1 Classical Linear Interpolation:

This is the most simple **interpolation method**, which works based on a linearized line between the nearest left and right values around the point $(P_{i,0} = 46.3, C_{i,0} = ?)$. Based on the values given in Table 9.20:

$$\frac{P_{i,3} - P_{i,0}}{P_{i,3} - P_{i,2}} = \frac{C_{i,3} - \tilde{C}_{i,0}}{C_{i,3} - C_{i,2}} \tag{9.45}$$

$$\tilde{C}_{i,0} = C_{i,3} - \frac{(C_{i,3} - C_{i,2})\,(P_{i,3} - P_{i,0})}{P_{i,3} - P_{i,2}} = \$683.4129 \tag{9.46}$$

9.3.3.2 Lagrange Interpolating Polynomial:

From the literature, it is known that the fuel-cost curve of real thermal units can be fitted as a 2nd order (i.e., quadratic) or a 3rd order (i.e., cubic) polynomial regression

model. Thus, it is logical to shift from the previous simple linear interpolation process to a more suitable process called "**polynomial interpolation**". The **Lagrangian-based approach** is used as follows:

$$\tilde{C}_{i,0} = \sum_{j=1}^{n} \left[C_{i,j} \prod_{\substack{z=1 \\ z \neq j}}^{n} \left(\frac{P_{i,0} - P_{i,z}}{P_{i,j} - P_{i,z}} \right) \right] = \$683.3651 \tag{9.47}$$

where n is the number of points used in the interpolation process, which is equal to 4 as per given in Table 9.20.

Using (9.44) gives $C_{i,0} = \$683.3651$, which means that the absolute error is:

$$\text{AbsErr} = \left|\Delta C_{i,0}\right| = \left|C_{i,0} - \tilde{C}_{i,0}\right| = \left|683.3651 - \tilde{C}_{i,0}\right| \tag{9.48}$$

Substituting (9.46) and (9.47) in (9.48) for $\tilde{C}_{i,0}$ gives $|\Delta C_{i,0}| = \$0.0479$ if the classical linear interpolation method is used and $|\Delta C_{i,0}| = \$1.1369E^{-13}$ if the Lagrangian-based polynomial interpolation method is used.

Of course, the real readings of the ith unit do not necessarily follow the quadratic or cubic curve, but the preceding concept can be applied between very narrow real points to estimate new non-recorded configurations. This may effectively improve the overall performance of OMF-ELD and make it more flexible and practical to satisfy any power demand even those not recorded in the operation logbook or the archiving server.

9.3.3.3 Aluminum Smelters Power Plants - A Special Case:

Based on our background experience, OMF-ELD can be a very competitive technique in finding the optimal configurations of **aluminum smelters' power plants**. The reason behind this claim is that these power plants are considered as a special case where the only semi-fixed load is connected to a very short HVDC line (with subtracting the power consumption of auxiliaries and other loads) as shown in Figure 9.34. This load is simply multiple arrays of **electrolytic cells** called **pots**.

If the OMF-ELD technique is applied here, then a highly accurate solution could be obtained. The reason behind this is that the total output of these power plants is almost constant where the electrodes of **aluminum pot rooms** are energized with

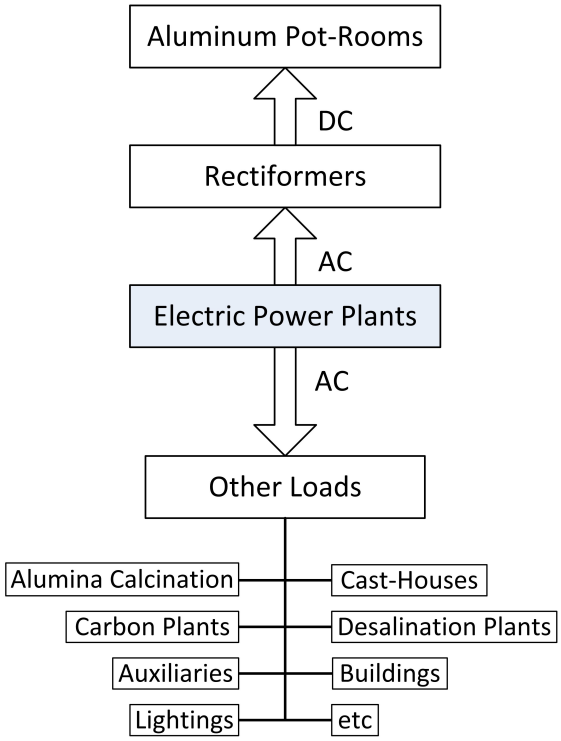


Figure 9.34: Simplified Diagram of the Main Loads Connected to Aluminium Smelters' Power Plants

a rectified electricity supplied from an array of special transformers called **rectifier-transformers** (**rectiformers**) for producing aluminum through an **electrolysis process**. Based on this, all the recorded configurations of smelters' power plants are located near each other. Thus, if these configurations are translated as hypothetical solutions, then all these collected candidate solutions will cover a large percentage of the practical and feasible parts of the search space because the aluminum production rate for any smelter is almost stable with different configurations of its power plants. Therefore, the associated error with the OMF-ELD technique could be effectively minimized. However, this interesting assumption has not been verified yet.

9.3.3.4 Brief Facts

This study highlighted many points regarding the proposed technique compared with other techniques presented in the literature. There are many facts that need to be clarified before concluding this section. We tried our best to address all the comments

raised by some experts, which are listed below:

- *Is the OMF-ELD technique applicable to deregulated and non-monopolistic electric power systems?*

  This point has been addressed during explaining the reason for using the local and global OMF-ELD stages. With monopoly power systems, both stages can be combined in one algorithm. But, because the modern power systems are highly deregulated and non-monopolistic, the local/global structure is proposed, which is graphically shown in Figure 9.26 where w power stations are connected to the grid.

- *What are the key points that distinguish this technique from artificial neural networks (ANNs)?*

  The new aspect here is about a technique that does not need to code a mathematical model (equations; objective function(s), equality constraints, inequality constraints, and side constraints), and it does not use any optimization algorithm. Yes, the alternative technique is to use artificial neural networks (ANNs). Such approach can be found in [140, 295, 391]. But, using neural networks will violate two important goals of this study: simplicity and being optimization-free. It is well known that the training phase of any neural network is simply an optimization stage, using BP algorithm or any other state-of-the-art global optimization algorithm. This means that ANNs are not optimization-free techniques. The second thing is about the implementation of ANNs. The proposed OMF-ELD technique can get rid of all the complicated models and weary calculations made by current modern optimization-based techniques. Employing AI will require additional knowledge and highly experienced data scientists, which violates the simplicity criterion. This technique takes care of conventional energy management systems that are built based on old SCADA and DCS systems; such as the old versions of ABB's ProControl DCS systems and GE's **Speedtronics control systems**. It has been seen that some power stations are operated by simple SCADA systems built based on Allen-Bradley's PLC containing **OPC server** and human-machine interface (HMI) units that are connected with old gas turbines (GTs). Further, some simple energy management systems do not contain a dispatcher module, and sometimes that module

could not be activated in modern EMS to reduce the total cost of some new power plants[28]. All these facts make implementing ANNs impractical option.

- *Are the solutions obtained by OMF-ELD global optimal?*
  The solution obtained by OMF-ELD is an estimate where the performance increases as the data quantity and quality increase. This fact has been said in different locations of the study. So, the question here: *What are the benefits?*

  1. Assume that a power plant does not have any strategy to reduce the operating cost (due to lack of experience, old EMS and automation systems, etc.), then what prevents that plant to get the best possible configurations of its units; easily by any person via some basic commercial and free software like MS Excel and **LibreOffice Calc**!
  2. This technique has a self-immunity against the real technical problem faced with multiple fuels where thermal units are operated based on a float ratio between two or more fuels. This phenomenon has been addressed in [11] and Chapter 7. Thus, implementing existing optimization-based techniques with this situation will not lead them to get optimal solutions, because real multi-fuel units can generate the same power with many possible ratios.
  3. This technique is supposed to be a very competitive technique in case it is applied to power plants of aluminum smelters.

- *What is the effect of big-data on the processing speed of OMF-ELD?*
  Regarding the **big-data** where **Hadoop** and **Spark** tools are used, all these technical things are not required if referring to the mechanism of the technique presented in the flowcharts and pseudocodes of the local and global dispatchers. That is, this technique does not search within the entire data set. Instead, the stored configurations of each xth power plant are labeled with their power outputs, and these values contain the settings of all the corresponding units. These entries are sorted based on the fuel consumption rates (i.e., their operating costs). Thus, if the global OMF-ELD stage asks the xth power station

[28]This cost includes the designing, commissioning, off-site/on-site training, and maintenance phases.

to provide P_{PS_x} MW, then the local OMF-ELD stage of that xth power station will look into the sorted configurations listed under that P_{PS_x} value. The first row will have the best configuration to supply P_{PS_x} MW at the cheapest known fuel price. That configuration can be swapped if any one of the units is unavailable, so the next row will be selected, and so on.

- *What is the main purpose of the global ELD stage?*
  The main purpose of this essential component of the OMF-ELD technique is briefly described via the flowchart shown in Figure 9.29 and the pseudocode given in Algorithm 15. The goal of the power system control is to guide all the w power stations to provide a specific amount of power from each individual power station. It acts as a global dispatcher that minimizes the total power losses in the network by optimally configuring the output of all the w power stations . Once the best net power P_{PS_x} is detected for each xth power station, the local w dispatchers of these power stations will start searching to find the best configurations to meet that power at the lowest possible operating cost. For example, suppose that there are 3 power stations and the demand is 300 MW. Now, suppose that the power system control finds, by Algorithm 15, the best configurations of these power stations, which are [70, 90, 40] MW. That is, the first power station must feed $P_{PS_1} = 70$ MW to the grid. Similar thing, $P_{PS_2} = 90$ MW and $P_{PS_3} = 40$ MW from the second and third power stations, respectively. Then, the local dispatcher (Algorithm 14) of the first power station will search for the best configuration of its generating units to provide a net power of 70 MW at the lowest operating cost. Similar thing for the local dispatchers of the second and third power stations.

- *Why there are no such experiments on Algorithm 15?*
  Yes, there is no numerical result for that part. The reason here is that only the power system control staff have access to this private information, and thus they are the only authorized entity that can conduct this experiment. However, the algorithm is straightforward, and it can be implemented easily.

- *How can newly added generating units provide many possible configurations even before starting to contribute to the grid?*
  This point means that the complexity in obtaining a precise mathematical model

of a power plant (containing multiple units) is solved because the expected performance from that plant is supposed to match the datasheets given by the corresponding original equipment manufacturer (OEM). Thus, many possible configurations can be generated from that plant even if it is just recently commissioned.

- *What is the purpose of presenting aluminum smelters' power stations in this study?*

  Actually, aluminum smelters have their own power stations. We are talking about 1 to 2 GW of power; or even more in some smelters. This huge amount of power is rectified through what are called rectiformers and then supplied to the pot-rooms to produce aluminum through an electrolytic process. The load here is almost constant with many different configurations of its power plants. Thus, applying OMF-ELD could provide highly accurate optimal solutions, because a big portion of the problem search space can be effectively explored by utilizing these rich configurations.

- *What is exactly the problem with existing ELD techniques?*

  This study covers many real technical problems that are not considered in current optimization-based techniques. Ignoring these problems will lead to getting non-optimal solutions. The technique proposed here bypasses all these restrictions because it is built based on real configurations detected from a real dataset.

- *Why existing ELD techniques will suffer from* (9.40)-(9.43)*?*

  These equations do not prevent the use of optimization algorithms. Instead, they make multiple barriers to them, so the solution expected to be obtained from them will take a long time. This is a common issue with any optimization algorithm containing equality constraints. The technical issue of the system active power balance equality constraint given in (3.16) is well known in the literature as a headache step; especially if the power loss formula, such as Kron's loss formula given in (3.18), is included. Thus, adding w new equality constraints will definitely complicate the entire process of any optimization algorithm. All these issues, associated with (3.16) and (9.40)-(9.43), are simply bypassed in this technique. It is a straightforward process where the best solution is estimated directly without referring to any iterative tool.

- *Suppose that the power system control center adopts some rational practice towards the minimization of operational costs, can we consider it as a competitive approach to the OMF-ELD technique?*

  It is very hard to provide a rational-practice-based solution that is better than that of OMF-ELD. In real-world ELD problems, each unit will have a huge number of configurations. This will end up with many thousands or millions of configurations for each power station as shown in Table 9.15. With w power stations, this mission, i.e. providing optimal scheduling by rational practice, will be almost impossible. Actually, this is the reason for adopting existing ELD techniques. Here, it can be said that the mission of the OMF-ELD technique is to compromise between the simplicity of the rational practice with the accuracy of existing optimization-based ELD solvers. Of course, talking about the accuracy of existing ELD solvers does not consider the effects of all the actual phenomena discussed in the study. Because neglecting these real technical problems could make the optimal solutions obtained by existing dispatchers infeasible.

- *How can we ensure that the interpolation phase will not generate high errors towards the actual configurations?*

  In terms of solution quality, OMF-ELD proves itself as an effective practical tool to minimize the operating cost of generating units. This claim has been validated by conducting some numerical experiments on 11 real gas turbines with different modes, settings, and fuel types. The technique is very efficient in determining the best possible solution among the stored data within a very short time compared with any existing technique. The reason is that it is an optimization/modeling-free technique. It can be called a filtration technique.

  Regarding applying the interpolation methods. This is a future extension that can be applied to OMF-ELD. Yes, the error could exist. But the study shows how to minimize that error in an effective way. This process can be summarized in two steps:

  1. Select very narrow configurations where the first one provides a power less than the value specified by the system control and the other configuration provides a power bigger than that value. As the delta between the two

configurations is small, the error is minimized.

2. It is known that the fuel-cost functions of thermal generating units are modeled by using a 2nd or 3rd order polynomial equation. Thus, the Lagrange interpolating polynomial method seems to provide more accurate results than the classical linear interpolation method.

- *How can the OMF-ELD technique convince neutral critics of its feasibility compared with other well-established techniques?*

  As an ideal electric power system, yes, existing optimization-based economic load dispatchers are the best approach. The problem appears when we jump from the modeling phase to the real-world phase. The study shows many practical issues that make these techniques very hard to be applied and hence their solutions are non-optimal and they could be infeasible as well. Even if all the power system components are assumed to be ideal without any draft or error and the mathematical model is highly precise with 100% identical to the real mechanical and electrical components, there are many obstacles force many power stations to not applying these techniques. Such limitations are: old DCS and SCADA, basic EMS, unskilled manpower, unlicensed software, old computing machines, etc. All these issues are solved by OMF-ELD.

  Also, when someone talks about the performance of any algorithm, he/she actually evaluates it based on four main criteria: accuracy, simplicity, processing speed, and project cost. Despite the first performance criterion, which is ambiguous due to the realization phase, the winner of the other three performance criteria definitely is the OMF-ELD technique. It is simple and fast, and it can be implemented in very basic software. Also, this technique can be easily executed without referring to highly skilled specialists.

9.4 Developing New Transformation-Based Linear Fuel-Cost Models to Compete with Quadratic and Cubic Models

The concavity of fuel-cost functions is the nature of thermal generating units. In Chapter 3, it has been seen how to model these functions by fitting different polynomial equations, such as the linear equation in (3.5), the quadratic equation in (3.7),

and the cubic equation in (3.9). Also, it has been said that the quadratic fuel-cost function is the most popular one because the linear regression model is weak to explain the variability and the cubic regression model is not significant if some statistical tests, such as the **t-test** and **F-test**, are applied. Thus, many studies use piecewise functions of multiple linear equations similar to that given in (3.11). The reason for using the last regression model is to make LP techniques applicable, and also to accelerate the computational speed of optimization algorithms [88]. That is, different linear equations are implemented together to linearize higher-order polynomial fuel-cost functions. Furthermore, the reason of using more than one linear equation is to increase the overall explanation level by maximizing both the coefficients of determination $\{R^2, R^2_{\text{adj}}, R^2_{\text{pred}}\}$ and the values of the t-test and F-test, and minimizing both the sample standard deviation (s) and the adjusted squared mean error (Adj MSE) or any other evaluation metric. However, using multiple "`if`-statements" to represent piecewise functions in any programming language also consumes a considerable CPU time and increases the complexity of the regression model.

This study proposes a new approach that can further accelerate the computational speed. This approach uses only one linear equation instead of a piecewise function of multiple linear equations approach. Moreover, the explanation level of this approach is very competitive to both quadratic and cubic equations. That is, with this new model the applicability, simplicity, speed, and accuracy criteria all are satisfied. The performance is validated based on a real dataset measured from one gas turbine (GT) of the SIEMENS power plant shown in Figure 9.31 during its start-up stage until reaching its baseload is fully committed to the grid.

9.4.1 Improving Non-Piecewise Linear Functions

Although the piecewise model given in (3.11) is very fast and it can be optimized by using LP algorithms, there are some few technical issues [17, 340]:

1. It is hard to determine the number of segments or zones Z_κ required for each ith unit.

2. Moreover, it is hard to determine the break-points between each two segments.

3. Programmatically, using nested `if`-statements consumes an additional CPU

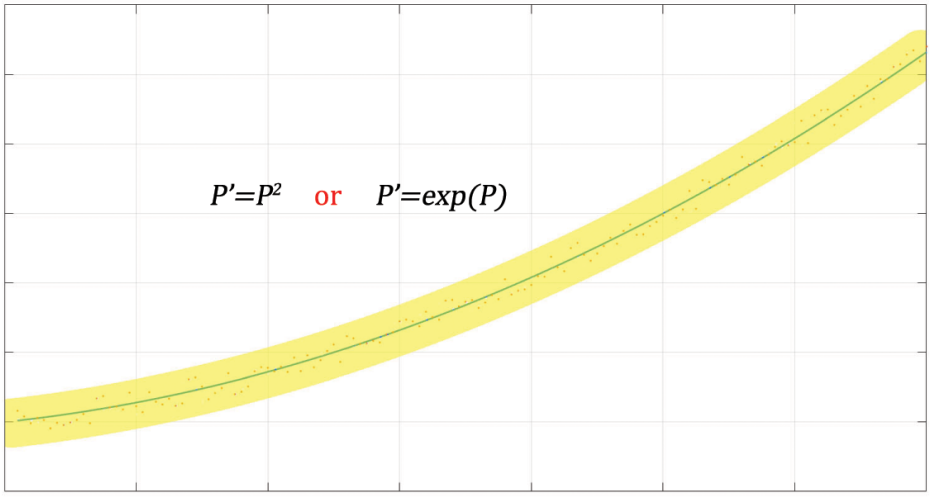


Figure 9.35: The Closest Prototype Regression Pattern to Thermal Fuel-Cost Curves

time compared with (3.5).

4. Also, it requires an additional effort from programmers to code (3.11) in optimization algorithms.

All these issues motivated us to look for a new flexible mechanism that combines the strengths of both the non-piecewise linear model and the higher-order models. This can be done by diagnosing the measures and then remedying them. One of these approaches is to use what is called the "transformation" technique. This approach can be done by transforming either the response variable (i.e., $\hat{C}_i$)[29], the predictor variable (i.e., P_i), or even both. By referring to the transformation guidance given in [214], the closest prototype regression pattern to thermal fuel-cost curves is the one shown in Figure 9.35.

This means that the original ith predictor P_i should be modified to $P_i^{'}$ before using it to regress the actual response variable C_i. Therefore, (3.5) becomes:

$$\hat{C}_i(P_i^{'}) = \beta_0 + \beta_1 P_i^{'} \tag{9.49}$$

It is important to note that both $\hat{C}_i(P_i)$ and $\hat{C}_i(P_i^{'})$ are just estimates of the actual fuel consumption C_i. Thus, $\hat{C}_i(P_i^{'})$ can be directly used to estimate the actual fuel

[29]Note that, in Chapter 3, the response variable was denoted by C_i. This symbol is just used for simplicity because the actual operating cost is estimated by the regression-based fuel-cost functions. Thus, the symbol $\hat{C}_i$ has to be used instead .

consumption C_i without transforming it to $\hat{C}_i(P_i)$. Yes, it is correct if the transformation is done on the response variable to have $\hat{C}_i'(P_i)$ instead of $\hat{C}_i(P_i')$, or if it is done on both $\hat{C}_i'(P_i')$.

Now, by considering the guidance shown in Figure 9.35, there are two possible ways to transform the predictor P_i of (3.5):

- If $P_i' = P_i^2$:

$$\hat{C}_i\left(P_i\right) = \beta_0 + \beta_1 P_i^2 \tag{9.50}$$

- If $P_i' = \exp\left(P_i\right)$:

$$\hat{C}_i\left(P_i\right) = \beta_0 + \beta_1 \exp\left(P_i\right) \tag{9.51}$$

However, based on our observation during regressing the real GT's I/O measurements, the actual scatter-plot between the active power and the fuel consumption is almost linear. Thus, the preceding two models (9.50) and (9.51) can be used just as a reference to build the following more general forms:

$$\hat{C}_i\left(P_i\right) = \beta_0 + \beta_1 P_i^x \tag{9.52}$$

$$\hat{C}_i\left(P_i\right) = \beta_0 + \beta_1 \exp\left(yP_i^x\right) \tag{9.53}$$

where $x \in (1,2)$ and $y > 0$, which can be determined through solving (9.52) and (9.53) directly using nonlinear regression approach.

9.4.2 Numerical Experiments and Discussion

To validate this regression approach, a real turbo-generator is used in these experiments. Table 7.2 shows the technical specifications of that machine. All the analysis are conducted after starting the unit from 0 MW up to its rated active power of 75 MW. The dataset consists of 185 samples measured from the "performance test data" page of the DCS automation system, which is shown before in Figure 7.4.

9.4.2.1 Classical Polynomial Regression Models

In this experiment, MINITAB ver.17.3.1 software is used to estimate the variability of the fuel consumption by regressing all the polynomial equations given in (3.5), (3.7), and (3.9). Table 9.21 shows some of the most important regression performance

Table 9.21: Performance Criteria of the Classical Polynomial Regression Models

Model	General			F-Test		Coefficient of Determination			Normality Test	
Type	DF	s	Adj MSE	F-Value	p-value	R^2	R^2_{adj}	R^2_{pred}	p-value	Pass
Eq.(3.9)	3	237.622	56464	31664.49	0.000	99.81%	99.81%	99.80%	< 0.005	no
Eq.(3.7)	2	237.303	56313	47624.07	0.000	99.81%	99.81%	99.80%	< 0.005	no
Eq.(3.5)	1	312.959	97943	54684.93	0.000	99.67%	99.66%	99.66%	0.059	yes

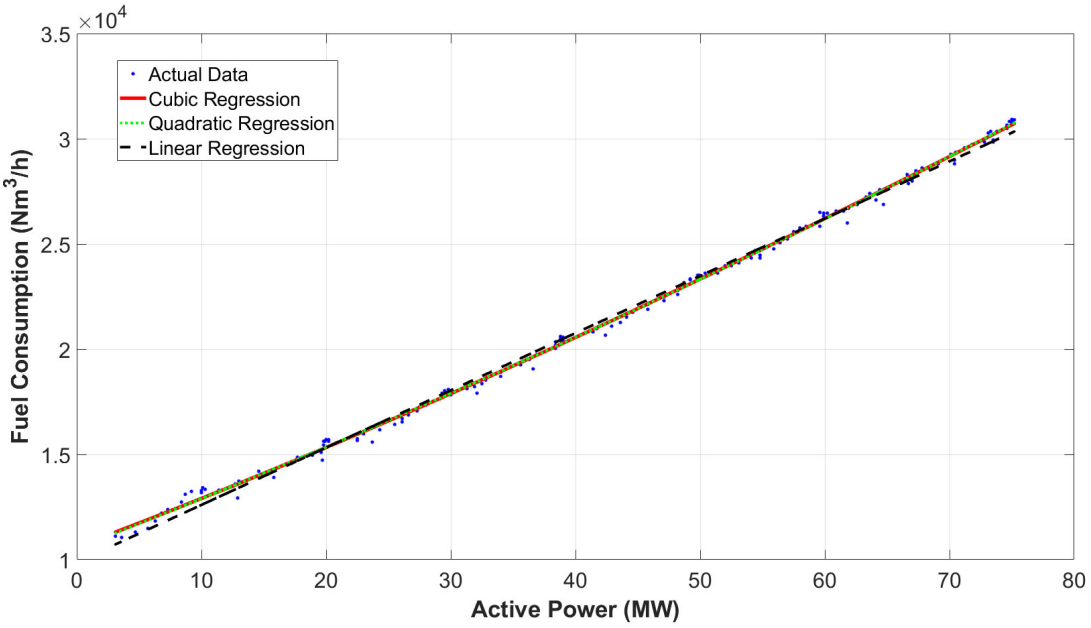


Figure 9.36: Fitting the Classical Cubic, Quadratic, and Linear Regression Models with the Actual Fuel Consumption Dataset

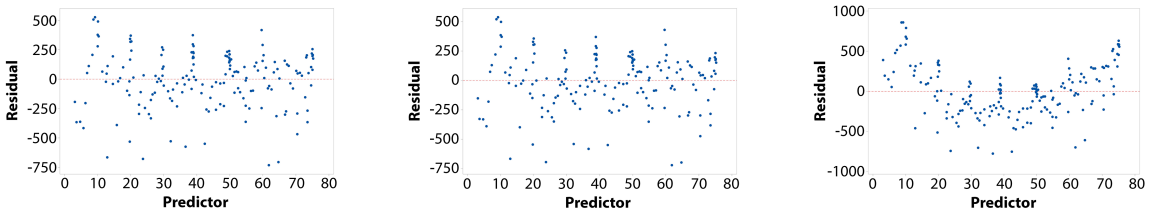
criteria calculated from each model. The plots of these classical regression models are shown in Figure 9.36. The numerical solutions are listed as follows:

$$\hat{C}(P) = 10611.7 + 222.149P + 0.73106P^2 - 0.0017107P^3 \tag{9.54}$$

$$\hat{C}(P) = 10549.6 + 229.458P + 0.520322P^2 \tag{9.55}$$

$$\hat{C}(P) = 9880.57 + 272.102P \tag{9.56}$$

From Table 9.21, although the coefficients of determinations $\{R^2,\ R^2_{\text{adj}},\ R^2_{\text{pred}}\}$ of these three classical polynomial regression models are almost equal, the sample standard deviation s and the adjusted mean squared error (Adj MSE) of the linear regression model are much larger than that of the cubic and quadratic models. However, the F-test value of (9.56) is better than those of (9.54) and (9.55). Also, (9.56)



(a) Cubic (b) Quadratic (c) Linear

Figure 9.37: Scatter-Plots Between the Residuals and the Predictor P of the Three Classical Polynomial Regression Models

passes the Anderson-Darling's normality test which states that the null hypothesis H_0 is rejected if the p-value is less than the significance level ($\alpha = 0.05$):

H_0 : *the residuals follow the normal distribution*, versus

H_1 : *the residuals do not follow the normal distribution*

Figure 9.37 shows the relationship between the residuals and the predictor of the preceding classical polynomial regression models. It is clear that the scatter-plot shown in Figure 9.37c has a specific pattern, which comes from the nonlinearity nature of the actual data. Thus, the residual points of the linear model depend on the predictor. This means that the linear model should be replaced by higher-order polynomial models, nonlinear models, or transforming its predictor or/and response.

9.4.2.2 Improving Linear Regression Model

Suppose that (9.52) is used for (9.49), then the predictor P must be transformed to ($P' = P^x$) before being used in the linear regression analysis:

$$\hat{C}(P) = \beta_0 + \beta_1 P^x = \beta_0 + \beta_1 P' \tag{9.57}$$

It is obvious that the optimal value of x can be found through an iterative process. But, before initiating the nonlinear regression analysis on (9.52), let's first solve (9.49)

Table 9.22: Performance Criteria of the Transformed Non-Piecewise Linear Regression Models

Model	General			F-Test		Coefficient of Determination			Normality Test	
Type	DF	s	Adj MSE	F-Value	p-value	R^2	R^2_{adj}	R^2_{pred}	p-value	Pass
Eq.(9.52) @ $x = 2$	1	1096.95	1203307	4282.97	0.000	95.90%	95.88%	95.79%	< 0.005	no
Eq.(9.52) @ $x = 1$	1	312.959	97943	54684.93	0.000	99.67%	99.66%	99.66%	0.059	yes
Eq.(9.52) @ $x = 0.975$	1	339.446	115224	46456.07	0.000	99.61%	99.61%	99.60%	0.006	no
Eq.(9.52) @ $x = 1.05$	1	269.732	72755	73679.96	0.000	99.75%	99.75%	99.75%	0.186	yes
Eq.(9.52) @ $x = 1.1$	1	243.885	59480	90166.05	0.000	99.80%	99.80%	99.79%	0.015	no
Eq.(9.52) @ $x = 1.15$	1	239.652	57433	93385.52	0.000	99.80%	99.80%	99.80%	< 0.005	no
Eq.(9.52) @ $x = 1.2$	1	256.824	65959	81291.24	0.000	99.78%	99.77%	99.77%	< 0.005	no

using the linear regression analysis for some selected values of x as follows:

$$x = 2 \quad : \quad \hat{C}(P) = 14489.1 + 3.16574P^2 \tag{9.58}$$

$$x = 1 \quad : \quad \hat{C}(P) = 9880.57 + 272.102P^1 \tag{9.59}$$

$$x = 0.975 \quad : \quad \hat{C}(P) = 9664.75 + 305.339P^{0.975} \tag{9.60}$$

$$x = 1.05 \quad : \quad \hat{C}(P) = 10284.5 + 216.323P^{1.05} \tag{9.61}$$

$$x = 1.1 \quad : \quad \hat{C}(P) = 10655.6 + 172.201P^{1.1} \tag{9.62}$$

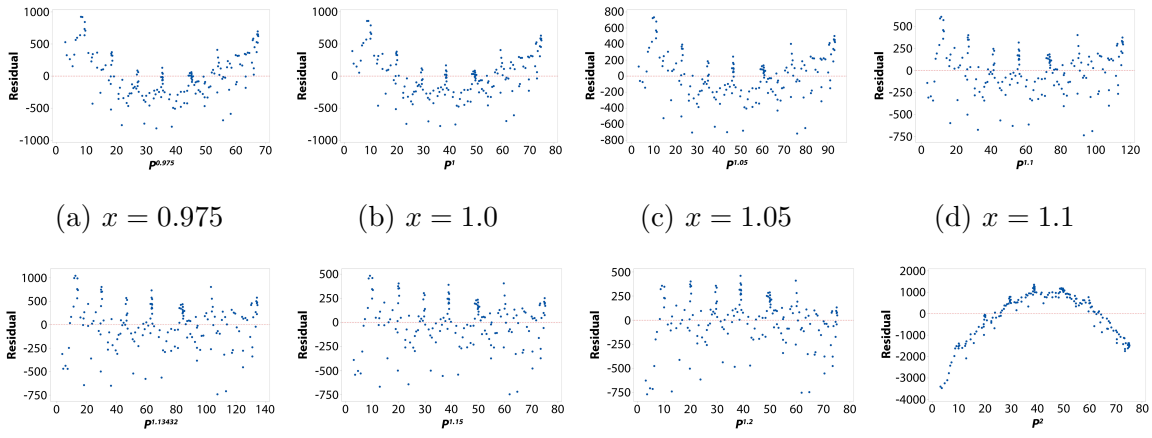
$$x = 1.15 \quad : \quad \hat{C}(P) = 10998.2 + 137.233P^{1.15} \tag{9.63}$$

$$x = 1.2 \quad : \quad \hat{C}(P) = 11315.7 + 109.475P^{1.2} \tag{9.64}$$

It is evident that (9.59) shows the same regression results of (9.56), because (9.52) becomes (3.5) when $x = 1$. Table 9.22 shows all the regression results of (9.58)-(9.64). Based on s and Adj MSE, it is obvious that the regression performance is ruined as $x \rightarrow 2$. Also, decreasing x below 1 does not help, which proves our assumption - when describing (9.52) - that the proper value of x should lie between 1 and 2. By continue increasing x by a step-size of 0.05, the regression performance is ruined again at $x = 1.2$. Therefore, the optimal value of x is near $x = 1.15$.

Now, it is the time to switch to the nonlinear regression analysis. The first step is to use the model given in (9.52) instead of (9.49). Thus, by iterating (9.52) using the values determined in (9.63) for $\{\beta_0, \beta_1, x\}$ as an initial guess or starting point, the numerical solution to this nonlinear regression model is determined as follows:

$$\hat{C}(P) = 10893.6 + 147.342P^{1.13432} \tag{9.65}$$



(a) $x = 0.975$ (b) $x = 1.0$ (c) $x = 1.05$ (d) $x = 1.1$

(e) $x^* = 1.13432$ (f) $x = 1.15$ (g) $x = 1.2$ (h) $x = 2.0$

Figure 9.38: Scatter-Plots Between the Residuals and the Predictors P^x of the Eight Transformed Non-Piecewise Linear Regression Models

Generally, the output result of nonlinear regression analysis does not show all the required information. Thus, it has to be re-switched again to the linear regression analysis. But, with substituting the optimal value of x (i.e., $x = 1.13432$) into (9.49). The final result of this linear regression model is found as follows:

$$\hat{C}(P) = 10893.6 + 147.341P^{1.13432} \tag{9.66}$$

It can be observed that (9.66) is exactly similar to (9.65), except a very slight change on β_1. Now, let's examine the relationship between the residuals of the preceding transformed linear regression models and their predictors, which are shown in Figure 9.38. It can be clearly seen that as x approaches its optimal value x^*, the specific patterns of the residual points vanish.

Based on all these facts, it can be said that (9.66) is highly competitive with the cubic and quadratic models given in (9.54) and (9.55). To prove this claim, Table 9.23 compares the performance of these three regression models. As can be clearly seen from that table, (9.66) can compete with (9.54) and (9.55) in terms of s, Adj MSE, R^2, R^2_{adj}, and R^2_{pred}. Moreover, (9.66) has the lowest degrees of freedom (DF) and much better F-Value than (9.54) and (9.55).

Therefore, (9.66) proves our claim that the transformed linear regression model given in (9.49) can be used in the place of the piecewise linear regression model given

Table 9.23: Performance Criteria of the Cubic, Quadratic, and Transformed Non-Piecewise Linear Regression Models

Model	**General**			**F-Test**		**Coefficient of Determination**			**Normality Test**	
Type	DF	s	Adj MSE	F-Value	p-value	R^2	R^2_{adj}	R^2_{pred}	p-value	Pass
Eq.(3.9)	3	237.622	56464	31664.49	0.000	99.81%	99.81%	99.80%	< 0.005	no
Eq.(3.7)	2	237.303	56313	47624.07	0.000	99.81%	99.81%	99.80%	< 0.005	no
Eq.(9.66)	1	238.551	56907	94251.22	0.000	99.81%	99.81%	99.80%	< 0.005	no

in (3.11). This approach can combine the strengthes of all the models (3.5), (3.7), (3.9), and (3.11). Thus, (9.49) can provide the following features:

- It is very fast model compared with (3.7), (3.9), and (3.11).
- It has the lowest DF; exactly similar to the classical linear regression model given in (3.5).
- It is a linear regression model, so the fuel-cost function can be easily optimized by many classical algorithms including LP[30].
- It can be easily programmed, and its code does not require any additional CPU time since there is no any `if`-statement involved here.
- Its performance is similar to (3.7) and (3.9).
- No need to determine how many segments are required for linearizing (3.7) or (3.9); as it happens with (3.11).
- No need to determine the break-point between every two segments, because there is only one segment here.

Based on all these pros, there is no hesitation in saying that (9.49) is much better than (3.11) in optimizing online and real-time electric power system operation problems. For example, the plot shown in Figure 9.39 on the logarithmic scale gives additional evidence to support the claim of its estimation quality. It shows that the fuel consumption estimated by (9.66) is almost identical to those of (9.54) and (9.55).

[30]This is very useful in hybrid optimization algorithms where the classical optimizers are used to exploit the solutions obtained by the meta-heuristic optimizers. Such algorithms have been seen in Chapter 2.

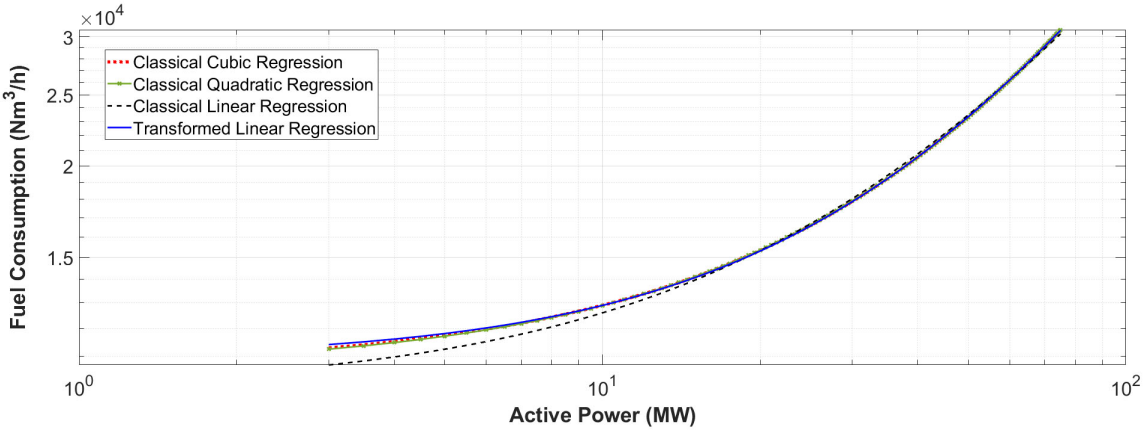


Figure 9.39: Logarithmic Plots of the Estimated Fuel Consumption Using the Classical and Transformed Regression Models

9.4.2.3 Realizing Operating Costs

It has been seen how to estimate the fuel consumption using (3.5), (3.7), (3.9), (3.11), and (9.49). However, to find the countable operating cost, the preceding models need to be further modified from a volumetric flow-rate (i.e., normal cubic meter per hour — Nm^3/h; where "normal" means: 0°C and 1 atm) to \$/h. Here, three possible billing approaches can be considered as follows [16]:

1. Billing no.1:

$$F_i\left(P_i\right) = \xi\ \hat{C}_i\left(P_i\right) \tag{9.67}$$

where ξ is the fuel's price rate in $\$/m^3$.

2. Billing no.2:

$$F_i\left(P_i\right) = \phi + \varphi\left[\xi\ \hat{C}_i\left(P_i\right)\right] \tag{9.68}$$

where ϕ is the initial cost in \$/h and φ is the operational pricing gain "unitless".

3. Billing no.3:

$$F_i\left(P_i\right) = \phi + \varphi\left[\xi\ \hat{C}_i\left(P_i\right)\right] + \psi\left[\xi\ \hat{C}_i\left(P_i\right)\right]^2 \tag{9.69}$$

where ψ is the second-order pricing coefficient in h/\$.

The other possible approaches are by changing the heat-rate unit (for example, the British thermal unit per hour, Btu/h) to \$/h, which take the same steps described in (9.67)-(9.69). More details about calculating these billings are given in [16].

9.5 Diagnosing Fuel Pumps, Power Transducers, CTs, and PTs via Fuel-Power Function and 2oo3 Voting Logic

Modern electric power systems are very advanced and their components are highly interconnected between each other. On each part of these components, many instrument devices are mounted to measure different variables and then transmit them to the automation center through DCS or SCADA system. These data are monitored and utilized to control the entire automation. Also, part of these data are sent to the corresponding EMS, so many electric power studies can be conducted. Such studies are PF analysis, contingency analysis, system security, fault analysis, ORC, ELD, UC, OPF, etc.

Any online process measurement is exposed to some errors [109]. These errors could happen due to an inherent accuracy problem of instrument devices or an operation failure in any internal part (the primary sensor, transducer, signal conditioner, or transmitter), transmission protocols, status of transmission mediums, modules of automation's control panels, modules of sub-systems provided by some third-party companies[31], communication between SCADA/DCS and EMS, **cyber-attacks** on SCADA, DCS, or/and EMS, etc.

Based on this, an state estimation (SE) stage is implemented to filtrate any highly erroneous measurements or bad data before it is used for on-line system security assessment and enhancement, contingency analysis, real-time PF analysis, automatic generation control (AGC), ELD, UC, etc. That is, the main purpose of SE is to produce the best estimate of the system state by diagnosing the measurements, detecting and identifying discordant values, and filtrating redundant data and incorrect measurements out before conducting any dependent analysis in EMS. Thus, through SE, reliable state estimates can be produced [109, 155].

One of these main real-time data, that needs to be correctly measured and directly utilized in EMS, is the power output generated from units committed to the grid. For example, this crucial measurement is used as an independent variable in what is called the fuel-cost function. The sum of all individual fuel-cost functions is taken as an objective function when solving ELD, UC, and OPF. The main power

[31] Such as microSCADA, **remote terminal units** (**RTUs**), **programmable logic controllers** (**PLCs**), individual protection systems, etc.

output measurement comes from what is called a power transducer module. Also, the preceding power output can be indirectly determined by measuring the terminal voltage and amperage through a potential transformer (PT) and current transformer (CT), respectively.

This study tries to hit multiple birds with one stone. Firstly, it borrows the well-known fuel-cost function from power system operation for utilizing it in state estimation to provide an indirect backup measurement of power generated by thermal units. Secondly, it borrows the well-known "**2 out of 3** (**2oo3**)" voting logic from instrumentation and control (C&I) engineering to compare the signals received in both energy management and automation systems; i.e., EMS and SCADA/DCS. This approach can ensure precise information is shared between power system operation, protection, state estimation, control, and automation; which has a significant impact on the overall performance.

9.5.1 Transposing Fuel-Cost Functions

It can be clearly seen that the fuel-cost functions given in Chapter 3 represent the estimated operating cost of thermal generating units, which can be expressed as functions of active or reactive power. Let's focus on the first independent variable; i.e., active power. The classical way to re-arrange or transpose the preceding functions for the ith unit is to make P_i the subject analytically. For example, if (3.5) is used, then the transposed version can be derived as follows [36]:

$$\hat{P}_i\left(C_i\right) = \gamma_{i,0} + \gamma_{i,1} C_i \tag{9.70}$$

where $\hat{P}_i$ is the estimated active power of the ith unit and C_i is the actual fuel consumption measured from the flow transmitter mounted on the fuel pipeline of the ith unit. The two coefficients can be determined as follows:

$$\gamma_{i,0} = \frac{-\beta_{i,0}}{\beta_{i,1}} \tag{9.71}$$

$$\gamma_{i,1} = \frac{1}{\beta_{i,1}} \tag{9.72}$$

For (3.7), the following quadratic formula can be used to estimate P_i from C_i:

$$\hat{P}_i\left(C_i\right) = \frac{-\beta_{i,1} \pm \sqrt{\beta_{i,1}^2 - 4\beta_{i,2}\left(\beta_{i,0} - C_i\right)}}{2\beta_{i,2}} \tag{9.73}$$

Similarly for the cubic function given in (3.9) where different analytical methods can be applied.

9.5.1.1 Issues Associated with Analytical Approaches

The main issue that should be seriously taken into account is that the original models are created by a regression analysis where uncertainty always exists. Thus, re-arranging (3.5) to have (9.70) is not preferred, because the coefficients $\gamma_{i,0}$ and $\gamma_{i,1}$ obtained by an analytical approach are not optimal. In other words, the response $\hat{P}_i$ is not the best-fitted curve. Thus, there is a possibility to find better models by fitting P_i as the subject of the model instead of C_i. The other issue is that the higher-order polynomial models give more than one solution, so a wise selection is needed to avoid unforgivable errors.

9.5.1.2 How to Solve These Issues

The original fuel-cost functions given in (3.3), (3.5), (3.7), and (3.9) are practically constructed through linear regression analysis. Thus, the re-arrangement process should also be carried out through another regression analysis. Thus, instead of using the non-optimal coefficients $\gamma_{i,0}$ and $\gamma_{i,1}$, given in (9.70), the optimal coefficients $\xi_{i,0}$ and $\xi_{i,1}$ should be obtained by solving the following linear regression model:

$$\hat{P}_i\left(C_i\right) = \xi_{i,0} + \xi_{i,1} C_i \tag{9.74}$$

The same thing can be applied to higher-order polynomial models. For example, the inefficient model given in (9.73) is replaced with the following straight-forward 2nd order polynomial model:

$$\hat{P}_i\left(C_i\right) = \xi_{i,0} + \xi_{i,1} C_i + \xi_{i,2} C_i^2 \tag{9.75}$$

and even (3.9), which can be easily re-arranged as follows:

$$\hat{P}_i\left(C_i\right) = \xi_{i,0} + \xi_{i,1} C_i + \xi_{i,2} C_i^2 + \xi_{i,3} C_i^3 \tag{9.76}$$

9.5.2 Numerical Experiments and Discussion

To support the claim raised in the last section regarding the analytical- and regression-based approaches in estimating active power, the real turbo-generator given in Table 7.2 is used in these experiments. The raw dataset consists of 185 samples. The outliers are filtered out before regressing the dataset. The cleaned version can be found in Appendix R.

9.5.2.1 Regressing Fuel-Cost Models

Because some observations are filtered out, so (9.54)-(9.56) are not valid anymore. For this updated dataset, MATLAB R2017b Curve Fitting Toolbox is used instead of MINITAB. Here, the subject is $\hat{C}$, so the numerical solutions to (3.3) are:

- For $d = 3$:

$$\hat{C}(P) = 10515.792 + 231.817\ P + 0.489979\ P^2 + 0.000216527\ P^3 \quad (9.77)$$

- For $d = 2$:

$$\hat{C}(P) = 10526.202 + 230.777\ P + 0.517772\ P^2 \quad (9.78)$$

- For $d = 1$:

$$\hat{C}(P) = 9727.487 + 275.503\ P \quad (9.79)$$

Now, let's find $\hat{P}$ analytically. To transform the linear model given in (9.79), the coefficients of (9.70) are calculated by (9.71)-(9.72) as follows:

- $\gamma_0 = -\beta_0/\beta_1 = -9727.487/275.503 = -35.3081$
- $\gamma_1 = 1/\beta_1 = 1/275.503 = 0.00362973$

Thus, the numerical solution to (9.70) is:

$$\hat{P}(C) = -35.3081 + 0.00362973\ C \quad (9.80)$$

The problem with this model will be seen later. Now, let's jump to the quadratic model. The analytical solution of its transposed version given in (9.73) is:

$$\hat{P}(C) = \frac{-230.78 \pm \sqrt{53258.14 - 2.07\,(10526.2 - C)}}{1.0355} \quad (9.81)$$

For (9.77), it can be directly obtained by MATLAB using `roots` command as follows:

$$\hat{P}(C) = \texttt{roots}\left(\left[\begin{array}{cccc} 0.000216527 & 0.489979 & 231.817 & (10515.792 - C) \end{array}\right]\right) \quad (9.82)$$

where the third root is the correct one, because the first root is -ve and the second root is much bigger than $P^{\max}$.

9.5.2.2 Regressing Fuel-Power Models

It has been seen how to analytically re-arrange fuel-cost functions to make P the subject instead of C. To prove that the curves of these models are not the best-fitted curves, the following regression analysis is conducted to find the optimal coefficients of (9.74)-(9.76):

- For $d = 3$:

$$\hat{P}(C) = -0.001994\ C + 2.8516 \times 10^{-07}\ C^2 - 4.6154 \times 10^{-12}\ C^3 \quad (9.83)$$

- For $d = 2$:

$$\hat{P}(C) = -46.308 + 0.0046974\ C - 2.4565 \times 10^{-8}\ C^2 \quad (9.84)$$

- For $d = 1$:

$$\hat{P}(C) = -35.11 + 0.0036207\ C \quad (9.85)$$

By comparing (9.85) with (9.80), it can be seen that there are some little differences in their coefficients. These deviations represent the errors associated with (9.80) because the optimal model is represented by (9.85). This can be proved by calculating the mean-squared error (MSE) of both approaches. MSE of (9.80) is 0.794851 while (9.85) gives 0.792885, which is less and thus the model is more accurate. The same observation could be seen by comparing (9.84) with (9.81), and (9.83) with (9.82). Therefore, (9.83)-(9.85) should be selected. Because the outliers are removed from the quadratic regression model, so (9.84) performs better than (9.83) and (9.85). Thus, (9.84) is used to estimate the active power produced from the generator side based on the actual fuel consumed by the turbine side; which is an opposite principle of the fuel-cost functions used in ELD studies.

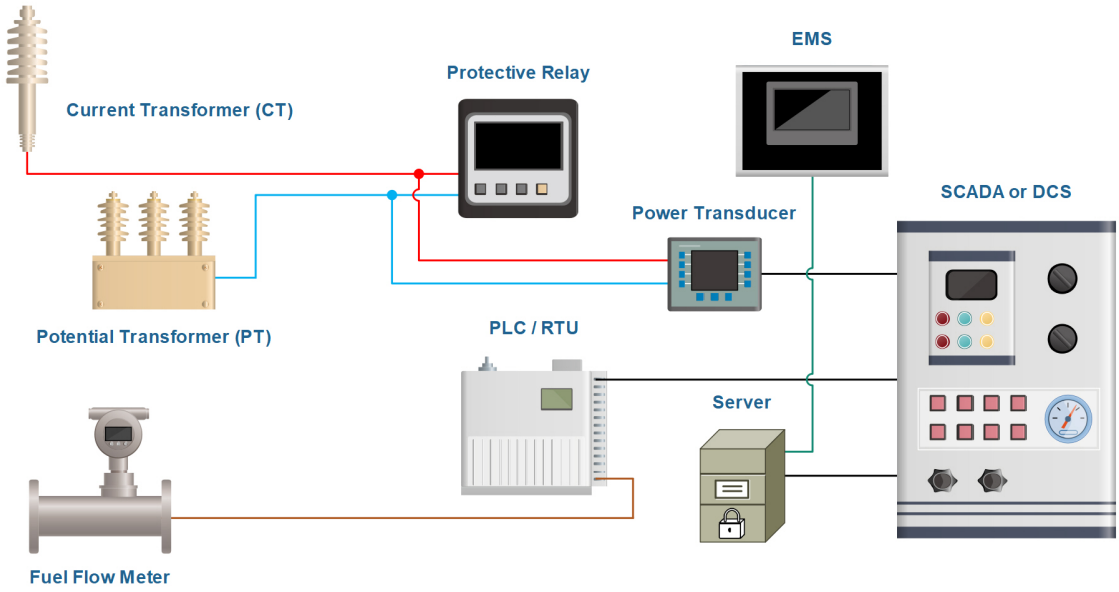


Figure 9.40: Illustrated Diagram to Show How to Estimate Power Readings from Different Instrument and Electric Devices

9.5.3 2oo3 Voting Principle

In electric power industries, there is an instrument device called a "power transducer". This device is used to calculate the power produced by generators. It is done by utilizing the real-time current and voltage readings measured by step-down CT and PT, respectively. Also, these measurements are fed into some protective relays to provide multiple lines of defense against different fault types. Furthermore, the fuels (such as diesel and natural gas) consumed by thermal units are measured and monitored, because the fuel reading is used to calculate the operating cost of thermal units. All these three vital measurement stages are depicted in Figure 9.40. In C&I engineering, there is a well-known logic called a "2 out of 3" voting logic (sometimes called 2oo3 voting principle). This logic is used for some critical measurements by installing three instrument devices and then taking an action based on at least 2 identical readings. The same principle can be applied here by voting on the three measurements shown in Figure 9.40. This can hit multiple birds with one stone. For example:

- It can act as a global checker to ensure that the current, voltage, power, and fuel flow-rate readings are correct.

- It can provide an additional layer of protection to protective relays by diagnosing the health status of their CTs and PTs.

- It can ensure that the estimated operating cost is accurate and thus the estimated value becomes more dependable than before.

- It can merge between different research areas, such as protective relays in optimal operation, fuel-cost functions of ELD to enhance protection, fuel flow meters as backup power transducers or as monitoring units of protective devices to check their health status, etc.

Some of the above benefits are further explained through the flowchart shown in Figure 9.41.

9.6 Possible Approaches to Trade Non-Electric Energy Sources in the Next Generation Smart Grids

Any electric power system consists of three main principal parts, called: 1. power generation, 2. power transmission, and 3. power distribution. In modern power systems, each one of these three major parts is built based on highly advanced technologies with the latest state-of-the-art techniques to ensure achieving all the tasks and instructions correctly and efficiently. These powerful features can be seen in hardware (static machines, rotating machines, towers, utility equipment and components, instrument devices, communication devices and protocols, computing machines, etc), software (programming languages, programs, algorithms, codes, computing systems, etc), and even firmware (such as those implemented in numerical relays and other intelligent electronic devices "IEDs"). All these arms are monitored, controlled, optimized, and protected through EMS. Of course, the automation part belongs to either DCS or SCADA system.

With the recent revolutions in sensors, digital systems, telecommunication, and **internet of things** (**IoT**), modern electric power systems are now able to do a two-way communication. Thus, the utility can send and receive information from/to consumers; and vice versa. Also, all the sensors mounted on different equipment of generation, transmission/sub-transmission, and distribution systems are smart with

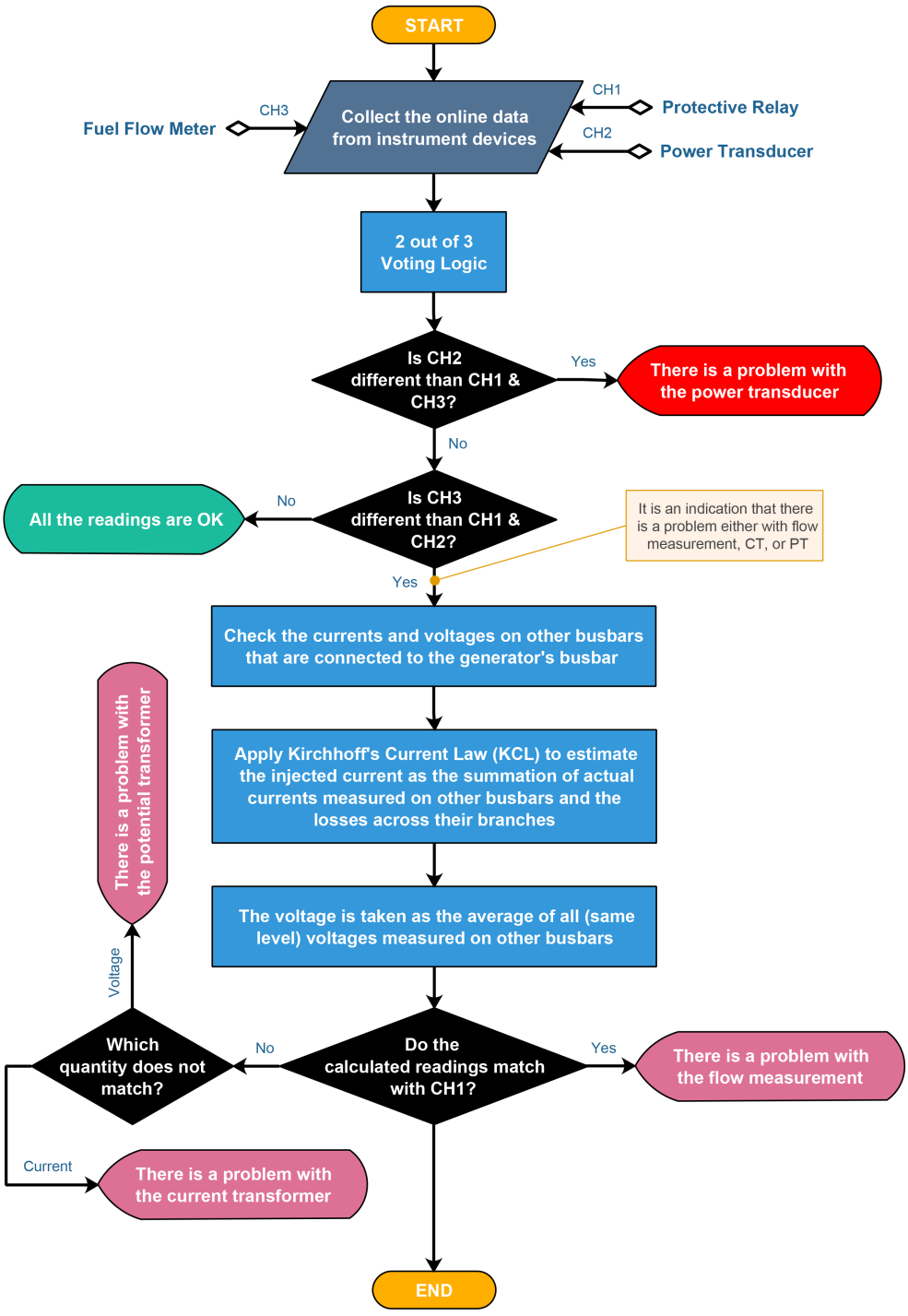


Figure 9.41: Flowchart of the Mechanism Proposed to Diagnose the Health Status of Instrument and Electric Devices

the self-calibration feature. The output signals initiated by these sensors are standardized through signal conditioning elements (that are built in their transmitters or transceivers). Some of these transmitters are remotely connected with their local/-central control units by using **RF** signals, or through some other standard protocols (such as HART, Foundation Fieldbus, and Profibus communications). All these capabilities make the preceding modern electric power systems very intelligent, and this is why they are called smart grids [18].

Actually, there is no specific definition of the term "smart grid" in the literature. For example, a blind grid can be turned into a smart grid in case its conventional electromechanical meters are replaced with smart ones. Also, when electromagnetic, solid-state, and hardware-based digital protective relays are replaced with numerical relays, the grid becomes partially smart. The reason behind this is that the cutting edge numerical relays, especially general-purpose types, can process the two fundamental signals (i.e., current and voltage) in any way based on their internal algorithms. They are equipped with two-way communication ports and their operations can be synchronized through time pulses received from **GPS** antennas "optional"; and this is why they are frequently called IEDs. The grid becomes fully smart if the whole operations are carried out automatically through computers and algorithms to ensure that everything are monitored, measured, archived, controlled, optimized, and safely operated. Different technologies and techniques can be seen when someone read about smart grids and their **advanced metering infrastructure** (**AMI**), such as IED, **smart meters**, **smart homes**, **smart appliances**, **active network management** (**ANM**), **automatic voltage control** (**AVC**), **dynamic line rating** (**DLR**), **phasor measurement unit** (**PMU**), **dynamic demand** (**DD**), **reactive power compensation**, **energy storage elements**, **electric vehicles** (**EVs**), **virtual power plants** (**VPPs**), **microgeneration**, **distributed energy resources** (**DER**), distributed generation (DG), and **distribution management system** (**DMS**) [223, 384].

With these highly advanced features, local energy trading between two or more entities becomes real and applicable. Thus, the entity that has an extra/surplus energy (due to no one at home, at night, good wind speed, clear sky, etc) can sell it to others who have some deficits [309].

Fortunately, power grids can accept energy generated by any electric-based renewable energy sources, such as wind turbines, **photovoltaic** (**PV**) **cells**, **biomass-based generators**, **solar steam turbines**, hydro turbines, **tidal turbines**, **wave turbines**, etc. But, unfortunately, this is not valid for non-electric-based renewable energy sources that are used in [20]:

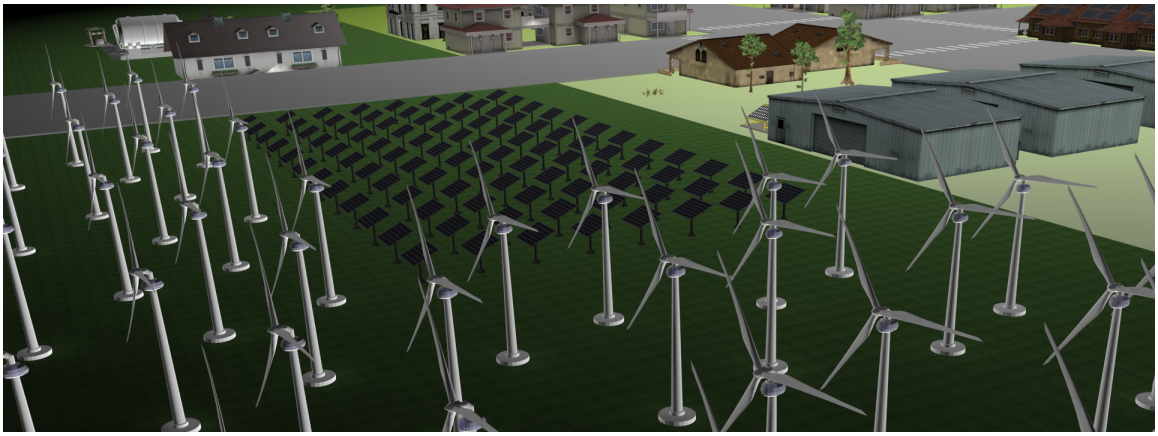
- Cooking: methane gas supplied from **biogas plants**.
- Transportation: vehicles operated by **biofuel** (**biodiesel** and **bioethanol**).
- Heating:
  - Hot air: stoves and furnaces operated by **firewood**.
  - Hot water: boilers and heat-exchangers operated by firewood, **solar thermal collectors**, **biogas**, **geothermal**, etc.
  - Hot oil: this special oil is not used as a fuel, instead it acts here as a heat reservoir (or buffer) because its specific heat capacity and boiling point are relatively high. Most of the **oil-filled heaters** or **radiators** available in the market are operated by electricity. However, it is possible to achieve that kind of efficient heating systems by involving renewable gaseous/liquid/solid fuels to heat that reservoir.

With this wide range of applications and benefits of using smart grids, still, there are some deficiencies due to the lack of efficiently coordinating non-electric forms of energy. First of all, let's see the virtual district illustrated in Figure 9.42. This virtual district gives two examples of the preceding forms of energy. For instance, Figure 9.42b shows wind and solar PV farms, and Figure 9.42c shows a solar thermal collector and two types of biomass (**logs** behind the wall and a biogas plant).

Of course, there are some other types of solar thermal collectors used to generate superheated steam for steam turbines (STs). Also, most of the types of biomass can be indirectly used to produce electricity; such as biogas electric generators and the next generation biofuel combustion technology for gas turbines (GTs) — as the one presented by the **National Research Council Canada** (**NRCC**) [271].

This study focuses on these renewable sources when they are used in their primitive non-electric forms of energy. During the winter season, especially in cold countries,

(a) Overall 3D Scene of a Virtual District



(b) Electric Energy Sources

(c) Non-Electric Energy Sources

Figure 9.42: 3D Illustrations of a Virtual District where Different Electric and Non-Electric Renewable Energy Sources are Harnessed

many heating systems are built based on water that is commonly heated up using these non-electric forms of energy.

Based on the literature, there are many attempts to maximize the benefits of using these **non-electric forms of energy**. These studies can be categorized into two streams: 1. designing innovative techniques to increase their harvesting efficiency or/and 2. utilizing the capabilities of smart grids to maximize the usage by finding their optimal operating intervals and forecasting the users' usage profiles. Such studies are those presented in [76, 119, 144, 157, 371]. However, I did not find any study that covers a strategy to do local energy trading using these non-electric energy forms. That is, this study tries to boost the smartness of the present grids by finding a way to trade these non-electric energy forms directly between entities without transforming them into electricity.

9.6.1 Typical Residential Heat Profile

Before proceeding into the topic, it is important to answer this vital question: *Is it significant for entities, in smart and* ***micro-grids****, to trade and exchange these non-electric forms of energy between each other; directly without transforming them into an electric energy form?* To be able to answer this essential question, it is crucial to analyze real data gathered from a typical user. For this mission, Figure 9.43 represents a profile of a typical customer in Nova Scotia, Canada. The readings were hourly recorded for the entire year of 2012. This data is taken from Nova Scotia

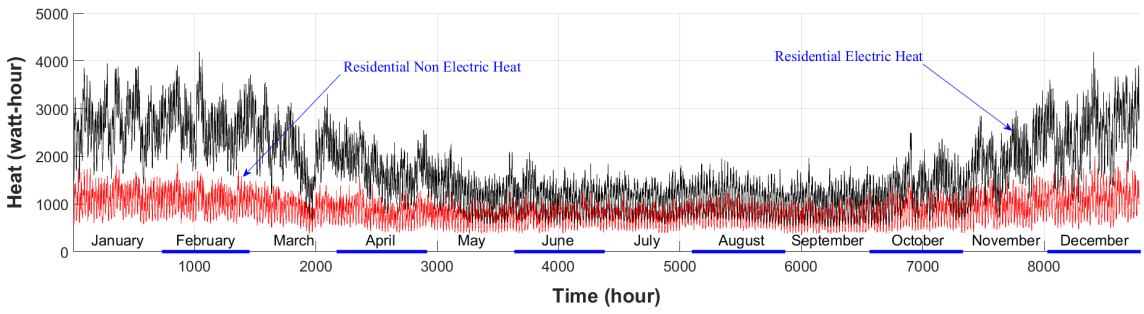


Figure 9.43: Actual Comparison Between Electric and Non-Electric Heat Consumed by a Typical Nova Scotian Resident. the Records Were Measured Every Hour for the Whole Year of 2012

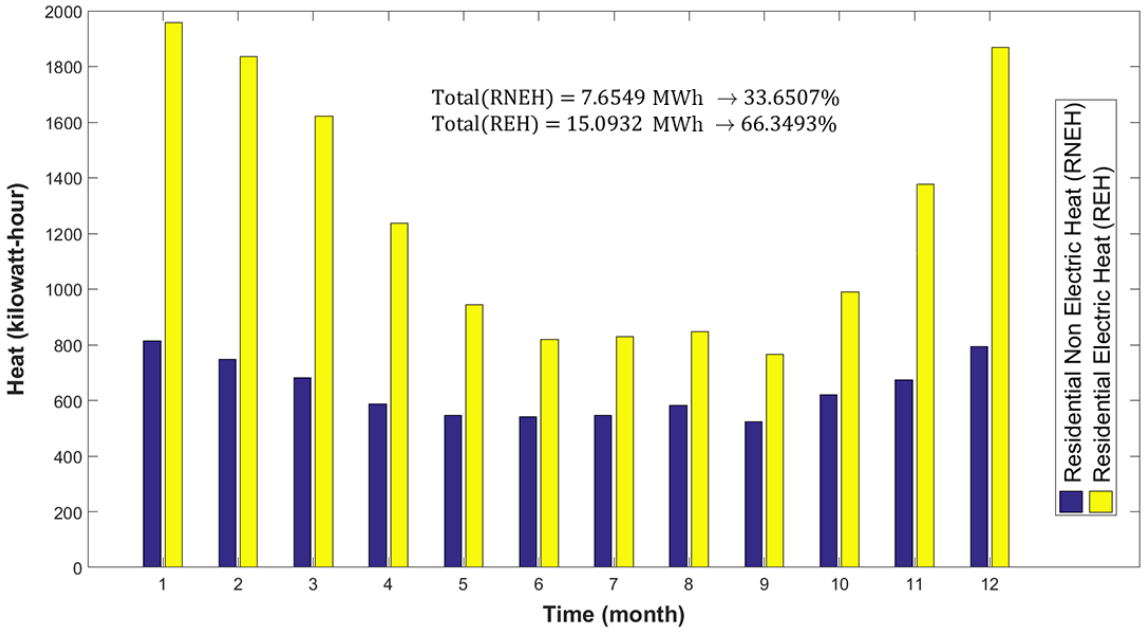


Figure 9.44: Monthly Residential Heat Consumption of Electric and Non-Electric Energy Sources

Power Inc. [285]. Although the amount of the residential electric heat is bigger, the portion of the residential the non-electric heat is also considerable. To have a more clear view, Figure 9.44 shows a monthly consumption of both the residential electric and non-electric heat.

These two plots reveal a fact that more than 33% of heat comes from non-electric energy forms, which is very big and highly significant. Thus, there is a problem when these forms of energy are not smartly coordinated with electric energy in smart grids. This fact gives a solid conclusion that the existing smart grids have an inherent deficiency in trading non-electric energy forms between local entities.

9.6.2 Smart Hot Water Pipelines Energy Trading Strategy

This is our first attempt to trade non-electric forms of energy between local entities, which is published as a self-contained study in [18]. This strategy is built based on a real fact that the main usage of these non-electric energy forms is to produce hot water for heating purposes. For example, Figure 9.42c shows three different types of energy sources. The first one is the **solar water heater**, which can produce hot

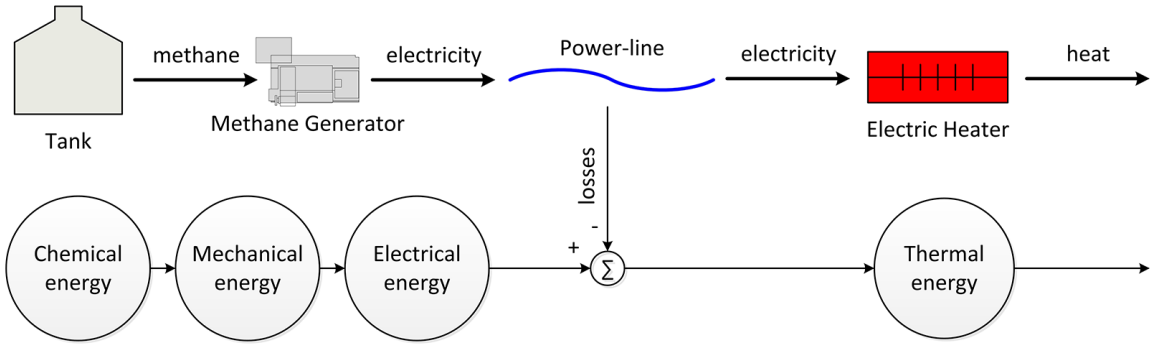


Figure 9.45: Direct Energy Transfer of a Biogas Plant Between Two Entities.

water directly from the sun. The second one is the wood fuel (i.e., the logs behind the wall), which is one type of biomass[32]. The third type of energy sources that can be seen from that figure is the biogas unit where the digester produces the methane gas and then collected in the gas storage tank.

For the first two types (i.e., the solar water heater and logs), it is hard to transform their heat into electricity. Even if it is possible, the overall efficiency is very important to decide whether these energy transformations are feasible or not. The biogas, instead, is more flexible where the methane gas can be either directly burnt to have hot water or to produce electricity through using a methane gas generator.

Thus, one of the practical alternatives to trade the heat produced by solar water heaters and logs is by transferring that thermal energy through pipelines connected between local entities. That is, to cover the missing link[33], the local trading strategy of future smart grids should have both powerlines and pipelines.

For the biogas plant shown in Figure 9.42c, its energy can be directly traded in the classical smart grids if a gas generator is integrated with it. This approach is graphically described in Figure 9.45.

However, from the preceding diagram, it is clear that there are different energy transformations. If the electrical losses are not considered, then the overall efficiency $\eta_{\text{overall}}^{\text{lossless}}$ of this heating approach can be calculated as follows:

$$\eta_{\text{overall}}^{\text{lossless}} = \eta_{c\to m} \cdot \eta_{m\to e} \cdot \eta_{e\to t} = \frac{t_{o/p}}{c_{o/p}} \tag{9.86}$$

[32]It has to be said that the wood is still the largest type of biomass used today. For example, wood stoves and wood furnaces in many heating systems.

[33]i.e., smartly coordinating the 33.65% of heat as seen in the typical customer profile in Figure 9.44.

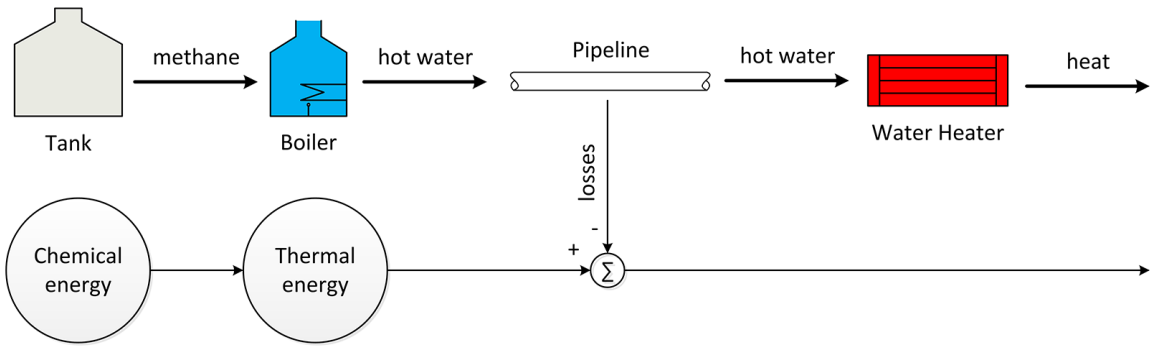


Figure 9.46: Indirect Energy Transfer of a Biogas Plant Between Two Entities

where $t_{o/p}$ and $c_{o/p}$ are respectively the outputs of the thermal and chemical energy. $\eta_{c\to m}$ is the energy efficiency of the chemical to mechanical transformation process. Similar thing for $\eta_{m\to e}$ and $\eta_{e\to t}$, where the subscripts e and t stand for the electrical and thermal energy, respectively. These three transformation efficiencies can be calculated as follows:

$$\eta_{c\to m} = \frac{m_{o/p}}{c_{o/p}} \tag{9.87}$$

$$\eta_{m\to e} = \frac{e_{o/p}}{m_{o/p}} \tag{9.88}$$

$$\eta_{e\to t} = \frac{t_{o/p}}{e_{o/p}} \tag{9.89}$$

where $m_{o/p}$ and $e_{o/p}$ are respectively the outputs of the mechanical and electrical energy.

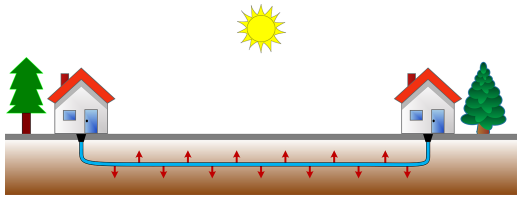
If the electrical losses e_L are not neglected and $\eta_{e\to t}$ is constant for both $e_{o/p}$ and $\tilde{e}_{o/p} = e_{o/p} - e_L$, then (9.86) should be modified to be as follows:

$$\eta^{\text{lossy}}_{\text{overall}} = \eta^{\text{lossless}}_{\text{overall}} \left(\frac{e_{o/p}}{e_{o/p} + e_L} \right) \tag{9.90}$$

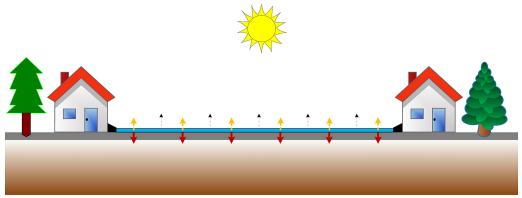
On the opposite side, if the methane gas generated from the digester is directly burnt to heat water in a boiler, then there is only one energy transformation as described in Figure 9.46.

If there are no thermal losses in the pipeline, then the overall efficiency $\eta^{\text{lossless}}_{\text{overall}}$ of this heating approach can be calculated as follows:

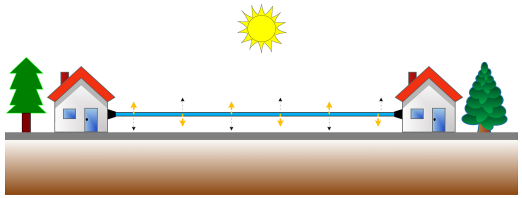
$$\eta^{\text{lossless}}_{\text{overall}} = \eta_{c\to t} = \frac{t_{o/p}}{c_{o/p}} \tag{9.91}$$



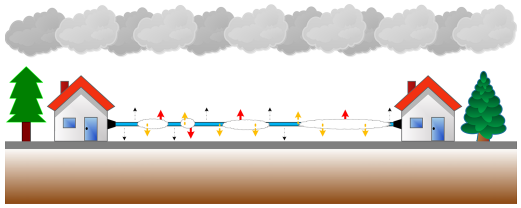
(a) Scenario No.1



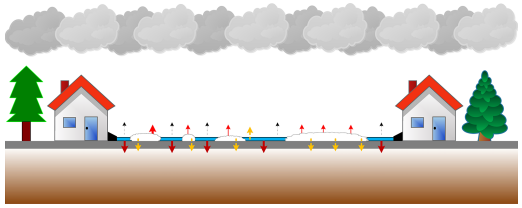
(b) Scenario No.2



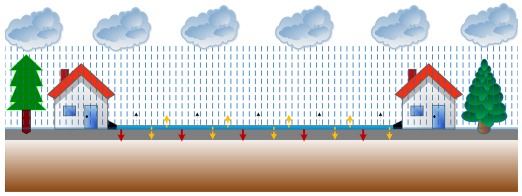
(c) Scenario No.3



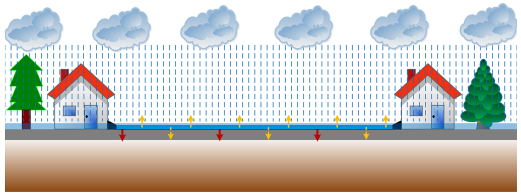
(d) Scenario No.4



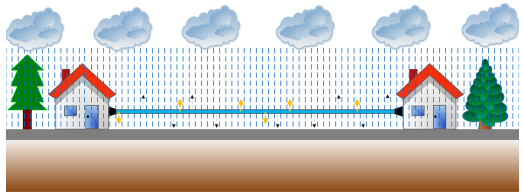
(e) Scenario No.5



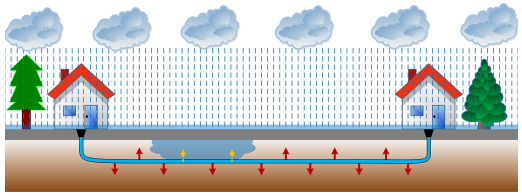
(f) Scenario No.6



(g) Scenario No.7



(h) Scenario No.8



(i) Scenario No.9

Figure 9.47: Nine Possible Heat Transfer Scenarios Across a Hot Water Pipeline Connected Between the xth and yth Entities. the Arrows Describe the Types of Heat Transfer on the Outer Surface: 1. Conduction: Thick/Red Arrows, 2. Convection: Thick/Dashed/Orange Arrows, and 3. Radiation: Thin/Dashed/Black Arrows

Table 9.24: Main Electric and Non-Electric Process Variables

Non-Electric Variables			Electric Variables		
Physical Quantity	**Symbol**	**SI Unit**	**Physical Quantity**	**Symbol**	**SI Unit**
Pressure	$\mathcal{P}$	Pa	Potential	U	V
Volumetric Flowrate	$\dot{v}$	m^3/s	Current	I	A
Min. Temperature	$\breve{T}^{\min}$	°C	Nominal Frequency	$f_\circ$	Hz
Heat[a]	q	J	Energy	E	J
Diameter	D	m	Impedance	Z	Ω

[a]It can be calculated using the values of $\breve{T}$, $\mathcal{P}$, $\dot{v}$, and D.

where $\eta_{c\to t}$ is the energy efficiency of the chemical to thermal transformation process.

If the thermal losses t_L are not neglected, then (9.91) should be modified to be as follows:

$$\eta^{\text{lossy}}_{\text{overall}} = \eta^{\text{lossless}}_{\text{overall}} - \frac{t_L}{c_{o/p}} \tag{9.92}$$

Therefore, it is important to know which energy form should be locally traded between the entity x and the entity y. Based on the transformation efficiencies and lines' losses, sometimes the approach given in Figure 9.46 becomes cheaper and sometimes not. The problem should be optimized during finding the cheapest option; that is, a buyer-seller optimization problem.

For entities that have solar water heaters and wood furnaces, the approach illustrated in Figure 9.46 could be adopted.

9.6.2.1 Heat Transfer Across Pipelines

It has been seen that the buyer could change his/her decision based on e_L and t_L shown in (9.90) and (9.92), respectively. For e_L, the literature is crowded with different approaches to calculate its value; which is a combination of active power losses P_L and reactive power losses Q_L. The problem here is with the second term, which is t_L. The pipeline installation and the weather condition both are very important variables that need to be taken into consideration to correctly calculate t_L. For example, Figure 9.47 shows nine possible scenarios where the three heat transfer types (conduction, convection, and radiation) could be faced together during calculating t_L. Based on the design, the pipeline could be installed under the ground (as in

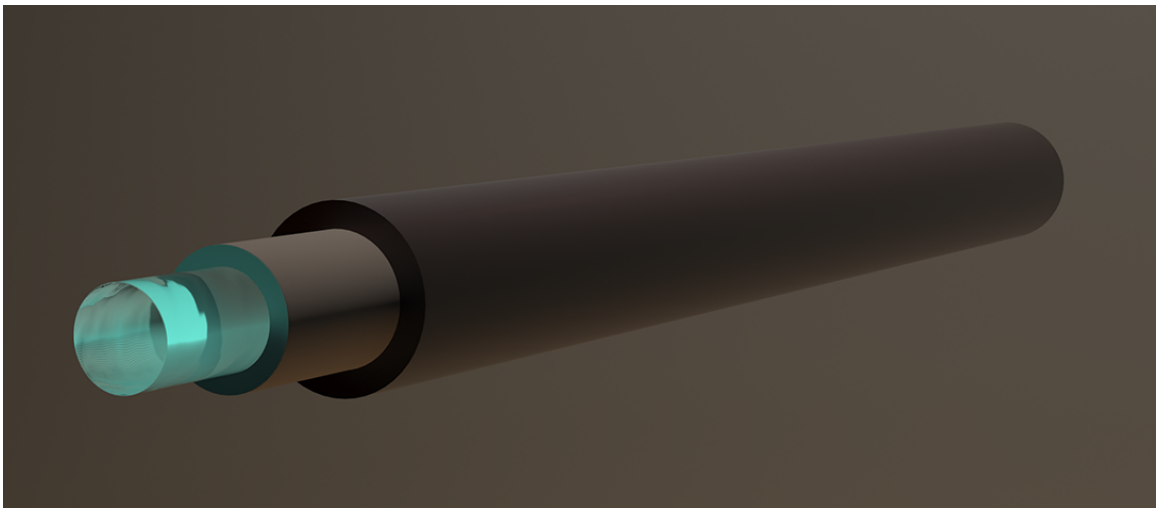


Figure 9.48: Insulated Hot Water Pipeline

Figures 9.47a and 9.47i), on the ground surface (as in Figures 9.47b, 9.47e-9.47g), or hanged-up above the ground surface (as in Figures 9.47c, 9.47d, and 9.47h). Based on the weather condition, it could be a sunny day (as in Figures 9.47a-9.47c), a snowy day (as in Figures 9.47d and 9.47e), or a rainy day (as in Figures 9.47f-9.47i). This is the beauty of involving the features available in the smart grids to make these blind hot water pipelines smart too. This can be achieved by utilizing all the digital signals available in the network; especially those devices that are integrated with the IoT features. Therefore, the distance between the xth and yth entities and the surrounding weather conditions can be determined. The other important process variables can be provided by installing new instrument devices. Such variables are listed in Table 9.24, which are compared with the equivalent electric variables.

For each scenario, the heat transfer calculation could be completely modified if the pipeline is insulated by some materials as illustrated in Figure 9.48.

9.6.2.2 Numerical Experiment and Further Discussion

It is important to call some equations from the heat transfer topic to address the effects of installation and weather parameters. Some of the main equations used in the literature are [143]:

- ***Conduction heat transfer:***

  This type of heat transfer is carried out using Fourier's law of conduction as

follows:

$$q = -kA\frac{\Delta\breve{T}}{\Delta x} \tag{9.93}$$

where k is called the thermal conductivity (in W/m.K), A is the area of the body exposed to the heat (in m^2), $\Delta\breve{T}$ is the temperature difference (in K) - also called the driving force - , and Δx is the thickness (in m).

- ***Convection heat transfer:***

  This type of heat transfer is carried out using **Newton's law of cooling** as follows:

$$q = hA\left(\breve{T}_s - \breve{T}_\infty\right) \tag{9.94}$$

  where $\breve{T}_s$ and $\breve{T}_\infty$ are respectively called the surface and average surrounding temperatures (in K), and h is called the heat transfer coefficient (in $W/m^2.K$).

- ***Radiation heat transfer:***

  This type of heat transfer is carried out using **Stefan-Boltzmann's law** of the power radiated from a **black body** as follows:

$$q = \varepsilon\ \sigma\left(\breve{T}_s^4 - \breve{T}_\infty^4\right) \tag{9.95}$$

  where ε is called the emissivity of the grey body ($0 \leqslant \varepsilon \leqslant 1$), σ is called **Stefan-Boltzmann's constant** (which is equal to 5.670367×10^{-8} $W/m^2.K^4$).

Now, let's simulate the heat losses across the hot water pipeline shown in Figure 9.47c:

Numerical Experiment

Suppose that the xth entity heats the water up using a methane gas generated from a biogas plant. Assume that the temperature set point of the boiler is adjusted at 82°C and the inside surface of the pipe has a convective coefficient of $h_i = 1000$ Btu/hr.ft^2.°F (i.e., $h_i = 5678.263$ $W/m^2.K$). The pipe is insulated with 1.5 inches of insulation on the outside surface with the logging outside convective coefficient of $h_o = 2$ Btu/hr.ft^2.°F (i.e., $h_o = 11.35653$ $W/m^2.K$) and the radiation heat transfer is neglected.

Except the "**critical thickness of insulation**" phenomenon, it is clear that $q \propto k$. For water pipelines, the following commercial materials are available in the market: Copper ($k = 386$ W/m.K), Aluminum ($k = 215$ W/m.K), Stainless Steel ($k = 45$ W/m.K), **PE-RT** ($k = 0.46$ W/m.K), and **PEX** ($k = 0.41$ W/m.K). Although the copper material is very expensive, it also shows the highest thermal conductivity k; which is highly not recommended for saving energy. Based on this, PEX is selected for this problem. However, the PEX tube itself is commercially available with different dimensions for both the inner diameter (ID) and the outer diameter (OD). In USA, the following PEX tubing standards are used [306]:

- 3/8": ID = 0.36" and OD = 0.5"
- 1/2": ID = 0.485" and OD = 0.625"
- 5/8": ID = 0.584" and OD = 0.75"
- 3/4": ID = 0.681" and OD = 0.875"
- 1": ID = 0.875" and OD = 1.125"

To find the best PEX tubing standard that has the lowest heat loss, the preceding standard dimensions are simulated for different distances between the entities x and y with assuming the weather temperature is $\breve{T}_\infty = 0°$C. Figure 9.49 shows that the best standard dimension for the PEX tube is 3/8". The maximum heat losses can be reduced from above 5.27 kW down to around 2.55 kW when the distance between the xth and yth entities is 80 meters.

After selecting the best material and standard dimension for the pipeline, the next goal is to select the best thermal insulator. In this experiment, the following materials are simulated: polyurethane foam, calcium silicate, mineral wool, silica aerogel, phenolic foam, asbestos[34], fiberglass, polyethylene, silicone rubber. Figure 9.50 shows the heat losses across different lengths of pipelines when the preceding thermal insulators are applied individually. From the last figure, it is obvious that the best insulation material is silica aerogel. It can further reduce the heat losses from 2.55 kW down to just 366.7 W.

[34]This insulation material is banned in around 58 countries.

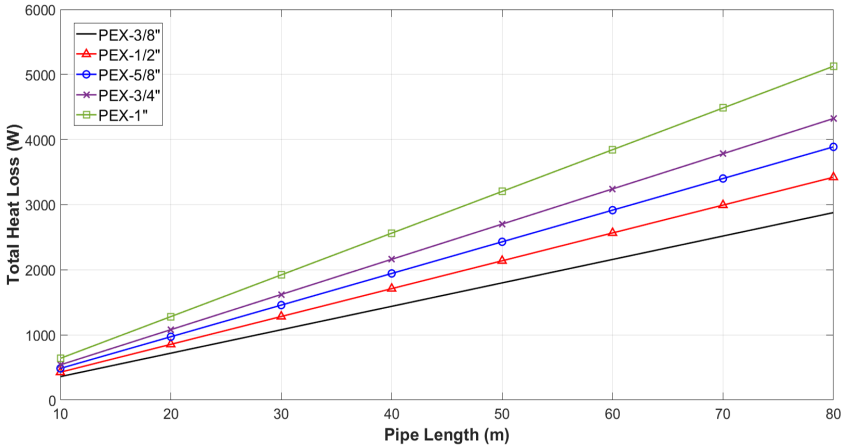


Figure 9.49: Effect of the PEX Tubing Standard on the Total Heat Losses of a Hot Water Pipeline at Different Lengths

Now, let's see the effect of the weather temperature $\breve{T}_\infty$ on the total heat losses at different distances between the two entities. Assume that the heating system of the yth entity (i.e., the buyer) is turned ON when $\breve{T}_\infty \in [-10, 20]^\circ$C. The effect of $\breve{T}_\infty$ on the convection heat transfer of the insulator's outside surface is shown in Figure 9.51.

From Figure 9.51, it can be observed that for each fixed-length L the heat loss across the pipeline increases as $\breve{T}_\infty$ decreases. In the worst case, the difference is 134.2 watts; which happens when jumping from $\breve{T}_\infty = 20^\circ$C down to $\breve{T}_\infty = -10^\circ$C at $L = 80$ m. Thus, $\breve{T}_\infty$ is also an important variable when involving these non-electric forms of energy in future smart grids. However, the most important variables that need to be seriously taken into account are: 1. the piping installation, 2. the piping standard dimension, 3. the insulation material, and 4. the weather condition; as shown before in Figures 9.47 and 9.50.

Discussion

It has been seen how to practically trade hot waters between residents in smart and micro-grids. However, many open points need to be answered. Such points are listed below:

1. It has to be said that the preceding problem is just a very simple scenario. The other eight scenarios, shown in Figure 9.47, could also be faced in cold countries. Moreover, there are many other scenarios could be seen in more realized designs.

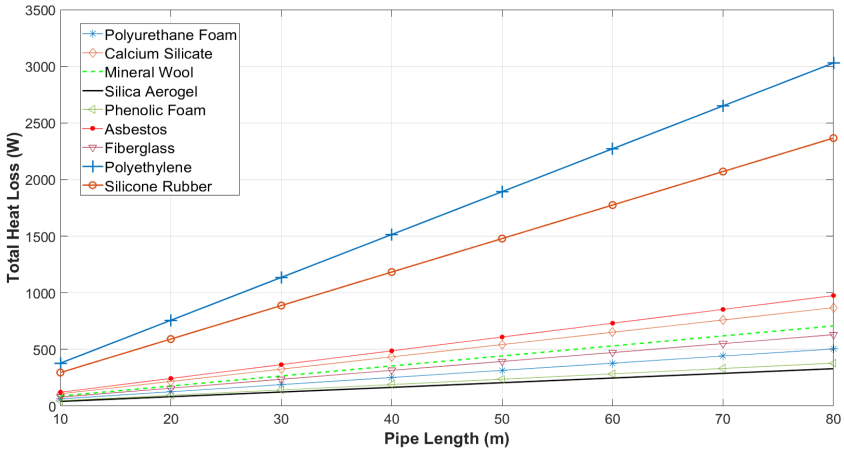


Figure 9.50: Effect of the Thermal Insulation Material on the Total Heat Losses of a Hot Water Pipeline at Different Lengths

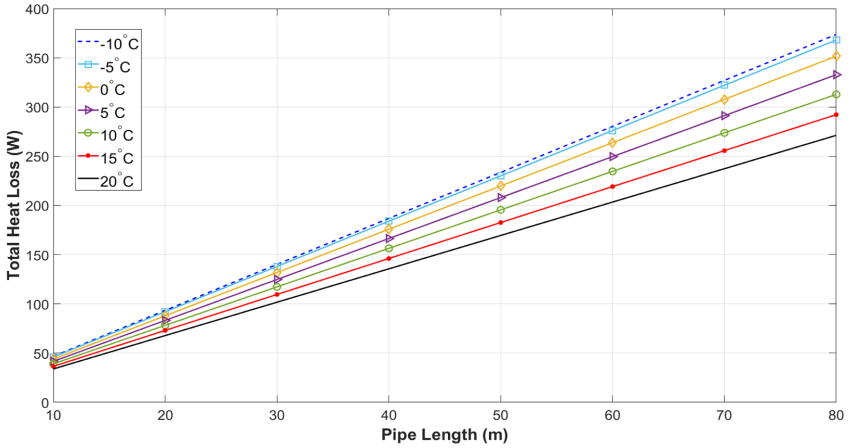


Figure 9.51: Effect of the Average Surrounding Temperature on the Total Heat Losses of a Hot Water Pipeline at Different Lengths

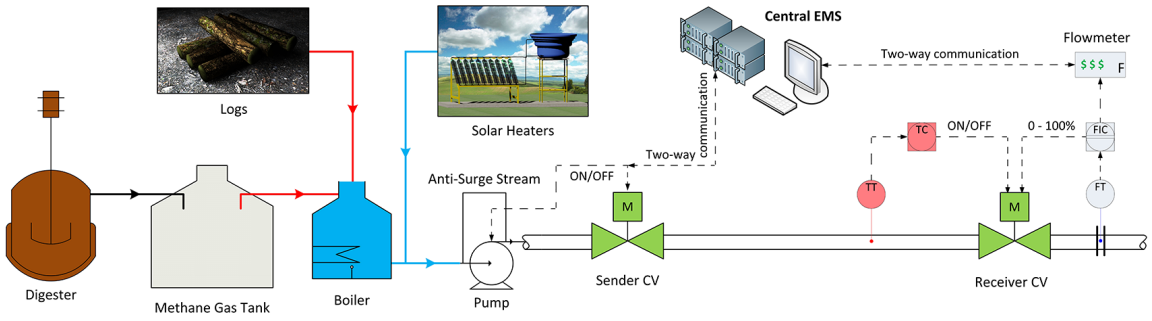


Figure 9.52: Simplified Piping and Instrumentation Diagram (P&ID) to Illustrate the Process of the Smart Hot Water Pipelines Energy Trading Strategy

For example: 1. inclined, partially vertical, and partially horizontal pipelines, 2. partially above/on/under the ground installation, 3. non-uniform geometry of pipelines, 4. different piping materials with multi-layer insulation approach, 5. phase transition due to melting phenomenon of accumulated ice/snow, etc.

2. The heat transfer due to the radiation part should also be considered. Thus, the body properties (material, paint type, color, glossiness, texture, roughness, etc) all should be addressed to select the best ε of the grey body so that the losses q can be reduced.

3. It may require to use **water pumps** (**WPs**) to transfer hot water between entities. Thus, the preceding heat transfer analysis should be modified to include this special case. Also, the installation, operating, and maintenance costs of these WPs should be taken into account during finding the optimal trading approach.

4. Based on the last point, the **fluid transport** topic should be involved to calculate the pressure drop due to the piping geometries and fittings (such as elbows, T-joints, pipe diameters, pipe lengths, pipe elevations, size of WPs, etc).

5. In the preceding example, the trading performance is evaluated by comparing the heat losses across a pipeline and the electrical losses across a powerline. To have a precise judgment, the efficiency of all the energy transformations shown in Figures 9.45 and 9.46 should also be considered to calculate the total energy losses of each alternative.

6. All the pipelines illustrated in the preceding diagrams have a one-way direction. But, in real-world applications, there are actually two pipelines; one is an outlet stream that carries the hot water from the xth entity (i.e., the seller) to the yth entity (i.e., the buyer), and the other is an inlet stream that returns the water from the yth entity after utilizing its heat. The reason behind this is that the buyer wants the heat and not the water (which is the carrier of that heat). Thus, the water after being cooled down should be returned back again. Here, two possible options could be applied in the proposed local energy trading strategy:

    - Returning that cool water to its provider freely, or

- Pumping it to some **common storage tanks** (**CSTs**) installed in the network, which can be re-pumped later to other providers.

7. For the seller side, the profit can be markedly maximized by compromising between the installation cost (i.e., the cost of the piping and insulation materials plus their maintenance) and the reduction in the heat losses.

8. For the buyer side, the profit can be maximized by selecting the best available energy source(s). *Should the buyer select an electric or/and a non-electric energy form? If it is non-electric energy form, then from which entity? Is it from the one who provides hot water generated by a biogas plant or from others who can satisfy the same energy requirement by using* ***solar heaters****, firewood, biofuel plants, geothermal heating/cooling systems, etc?*

9. The other selection criteria should also be considered, such as safety, environment, greenness, and aesthetics.

10. To successfully transfer thermal energy through hot water pipelines, the entity belongs to the buyer-side must be equipped with a water heater. Otherwise, he/she cannot utilize that thermal energy; and vice versa for electric energy where the buyer cannot use it if he/she does not have an electric heater.

11. By referring to the last point and Figures 9.45 and 9.46, the efficiency of both electric and water heaters should also be accounted in deciding the best energy trading approach.

12. The last point is about the criterion that accepts transferring the heat from the xth entity to the yth entity. In the classical smart grids, the AC power should be synchronized to 50/60Hz before being able to trade it locally in the grid, while in this proposed smart grid, the temperature on the receiving-end point of hot water pipelines should be equal to or bigger than $\breve{T}^{\min}$; as highlighted in Table 9.24.

Figure 9.52 summarizes many of the above points. It also shows how to monitor, control, and protect the proposed smart hot water pipelines using the **piping and instrumentation drawing** (**P&ID**) approach. Although this new local energy

trading approach can integrate many non-electric energy forms into the next generation smart grids, it is an uncompleted solution. There are some drawbacks associated with this concept; i.e. smart hot water pipelines. Such drawbacks are:

1. This concept is exclusively designed for some specific heating systems. Thus, the other applications, such as cooking and transportation, still are blind.

2. The heat losses across pipelines depend on many criteria, such as installation, material(s), insulation(s), geometry, water temperature, water velocity, free or forced by WPs, weather condition (temperature, wind direction, wind velocity, sunny/rainy/snowy, etc).

3. Add to the last point, even if the best design and weather condition are met and satisfied, the heat losses across pipelines are proportional to the distance between entities. Therefore, the benefits expected from this local energy trading strategy are limited, because the providers are not able to sell their hot waters to entities located far away from them.

4. The seller should stay in his/her entity to utilize the heat generated from the boiler, and then send the residual to other entities. Otherwise, an independent piping unit should be installed to provide auto-selective control.

5. The thermal energy itself cannot be stored for a long time if there is no anyone wants to buy it.

9.6.3 Other Innovative Concepts to Trade Energy Harvested from Non-Electric Sources

From the last subsection, it has been seen that there are many factors must be considered during selecting between electric and water heaters. To maximize the selection weight of non-electric energy sources, there is a need to conduct more research to find other possible alternatives that could be used to transfer non-electric energy between entities. That is, based on the factors affecting the heat losses across pipelines and the inherent limitations of the preceding trading approach (i.e., the hot water concept), it can be said that the selection weight can be maximized by reducing the losses and the usage limitations. Some of these alternatives are listed below:

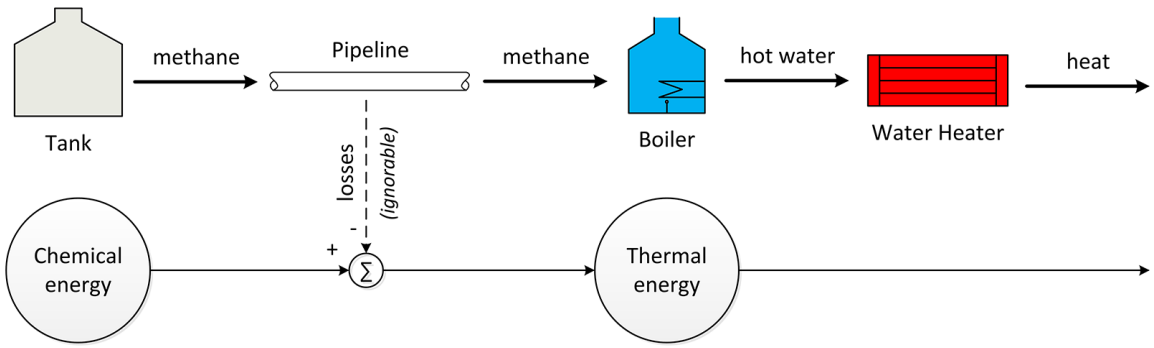


Figure 9.53: Modified Indirect Energy Transfer of a Biogas Plant Between Two Entities

9.6.3.1 Biogas Plants

The third novel alternative is to shift the "chemical to thermal energy transformation" to the buyer-side. The configuration diagram of this new approach is shown in Figure 9.53 diagram. That is, instead of heating the water by using a boiler installed on the seller-side, the methane gas generated from the digester can be directly sent to the buyer so he/she can then boil his/her water or utilize that gas in any other purposes. This new concept has many advantages. For example:

- Non-significant or ignorable losses in its flow-rate. Thus, a local energy trading can be established even with a long distance between the xth and yth entities.
- Consumers can adjust the gas amount based on their usage.
- Methane gas can be used in cooking and heating systems.
- Buyers can re-sell the remaining or unused methane gas to other entities, or they can act as distributors.
- Sellers and buyers can use storage tanks to keep the remaining or unused methane gas for future uses.

The losses mentioned above mainly come from the condensation phenomenon where a very small portion of the transported methane gas is drained through some condensate traps. This issue can be avoided, or at least minimized, by sticking on the pipeline operational curve; which also opens the door for integrating the thermodynamic topic. However, because the liquid condensate is flammable, so it can also be utilized and reused again.

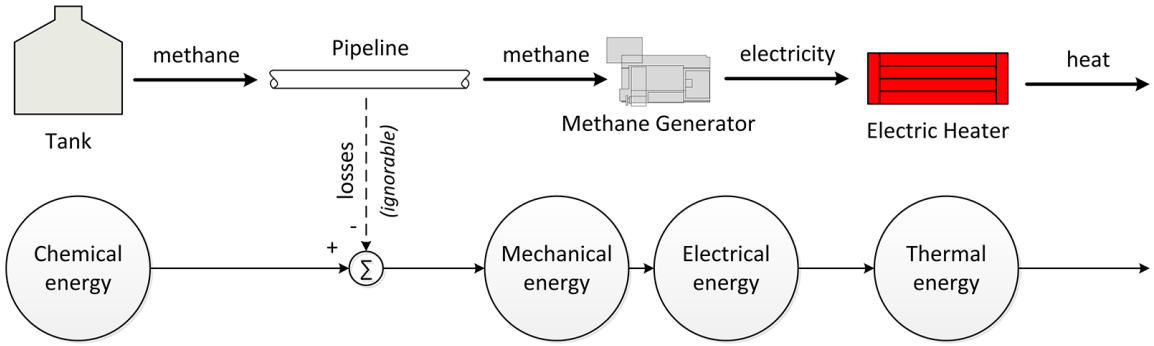


Figure 9.54: Modified Direct Energy Transfer of a Biogas Plant Between Two Entities

On the opposite side, there are some disadvantages associated with this approach. For example:

- Transporting flammable gas requires extra precautions.
- Burning quality of any fuel gas is affected by its preheating temperature.

The first point can be eliminated, or minimized to its lowest level, by installing underground gas pipelines, preventing them be highly pressurized, and avoiding any source of heat near them. The second one is mainly faced in power industries where heating plants are used to preheat and filtrate fuel gases, coming from wells, before being sent to the combustion chambers of gas turbines (GTs) [106, 385]. For domestic applications, where a local energy trading between entities take a place, it can be said that this argument (i.e., the effect of temperature on the burning quality) is still vague, and thus it needs more investigations. Also, the last point triggers some ideas about employing a portion of hot water, produced by any renewable source, to preheat the methane gas before being burnt. However, this process should be carried out based on the profit and loss criteria. Add to that, underground gas pipeline installations could be a good choice if a **trade-off solution** is made based on the depth versus installation and maintenance costs.

The same concept can be applied to modify the approach shown in Figure 9.45. Therefore, the xth entity is responsible to provide a specific amount of methane gas to the yth entity. Thus, the electricity required to operate the buyer's electric heater or any other appliances can be produced through a methane gas generator installed on the buyer-side. This new configuration diagram is shown in Figure 9.54.

9.6.3.2 Solar Thermal Collectors

In this case, the owner does not have any other choice but to use the concept shown in Figure 9.46. The production could be sent directly to the buyer or used to preheat methane gas in underground pipelines of Figure 9.53. Therefore, more research should be conducted on heat transfer and fluid transport topics. The simple diagram shown in Figure 9.52 is also valid for this approach. All the points listed in the last subsection should also be addressed here, so more accurate decisions can be taken during selecting the best approach to trade energy locally between entities.

9.6.3.3 Wood Fuel

Wood fuel, which is also known as **fuelwood**, can be split into different types, such as **pellets**, **charcoal**, **sheets**, firewood, **chips**, and **sawdust**. The burning quality and price play an important role in deciding the best type to be used. Wood fuel, which is listed as solid biomass, occupies the biggest portion of all the known biomass renewable sources used by human. The applications of wood fuel cover heating, cooking, and sometimes electricity production by using steam turbines (STs); or even gas turbines (GTs) via what is called **liquid wood** [3]. Also, the **wood gas**, which is a **syngas fuel**, can be used in stoves, furnaces, and vehicles. For indoor heating systems, the wood fuel can be used in stoves and fireplaces, while campfires and bonfires can be used outdoor, and furnaces can be used in both indoor and outdoor. In general, let's just focus on logs to see how they can be locally traded between entities. The same concept can be applied to any other type of fuelwood.

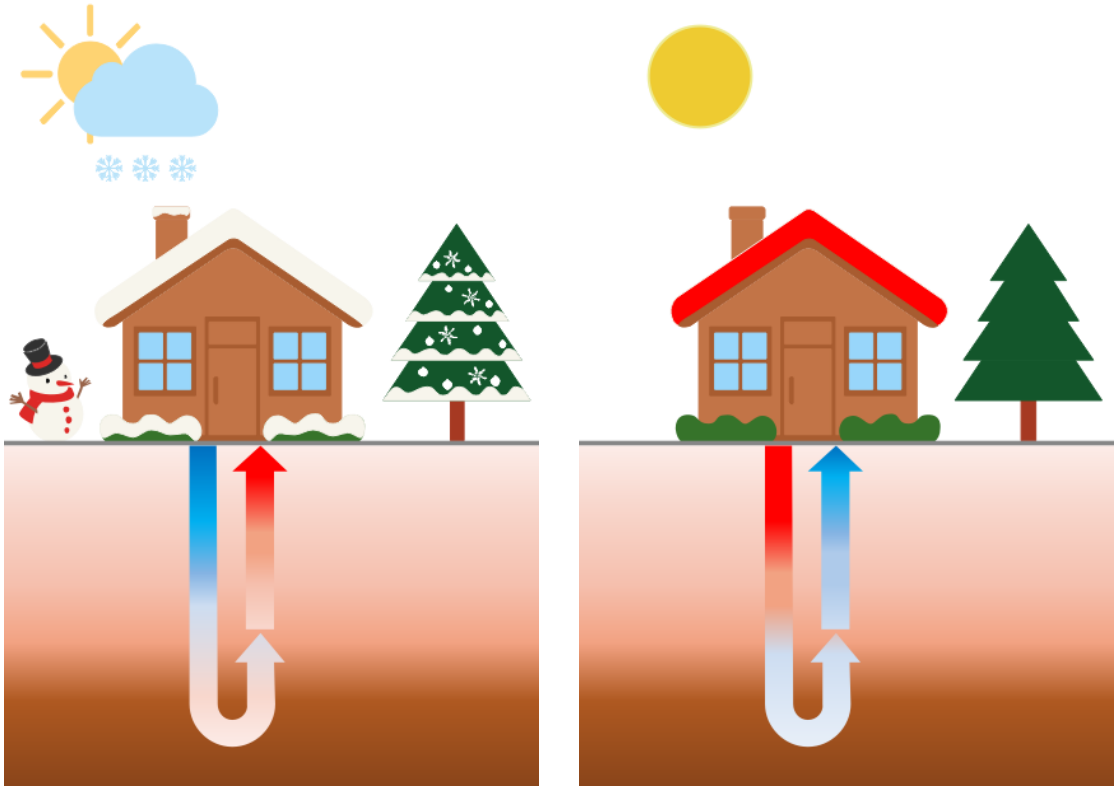
As seen before, logs can be used as a heating fuel in all the fluid-based heaters. If the former approach given in Figure 9.46 is used, then there is no other choice but to use the design described in Figure 9.52. The inherent limitations of this energy trading approach appear when the buyers have air heaters or if they want that heat for cooking.

To solve all these limitations plus the losses across the pipelines of Figure 9.46 and Figure 9.52, a patch process shown in Figure 9.55 can be used instead. This approach is very useful for lumberjacks because it allows them to sell their logs or exchange them with other energy forms. For example, they can sell their surplus products during the winter season and then buy the required electricity during the summer

Figure 9.55: Illustrated Strategy to Trade Fire Logs Indirectly via a Patch Process That Is Automatically Organized by Some Public Logs Warehouses

season to cool their houses and power their appliances. This indirect sell/buy trading strategy can be replaced with a computerized-based automatic exchange process. Here, the seller will get a credit for any fire logs he/she gave to the nearest **public logs warehouse** (**PLW**). Then, for each unit of electricity he/she consumed, specific money will be directly deducted from his/her account. This innovative approach is illustrated in Figure 9.55. It has many prospective applications because the consumer buys the raw materials (i.e., the logs) instead of restricting or depending on the energy transformation shown in Figure 9.46. Thus, the buyer can use these logs for heating and cooking no matter indoor or outdoor.

The purpose of the smart grid is to act as an organization and energy management center. From the smart meters installed in the yth entity, it can detect its energy needs. Then, based on the equipment installed and appliances used in that entity, the smart grid can guide the property owner to consume a specific amount of logs available in the zth PLW. Of course, the whole process is controlled based on some performance criteria that will be discussed later.



(a) Winter Season (b) Summer Season

Figure 9.56: Geothermal Heating and Air Conditioning Systems

9.6.3.4 Geothermal Heating and Air Conditioning Systems

The principle behind this renewable source is that the underground temperature stays consistent all year along. Suppose, a heat-exchanger is installed in a trench and water is recycled via a **geothermal heat pump** (**GHP**). Then, the preceding phenomenon can be utilized as a heating system during the winter season and as an air conditioning system during the summer season. These two heat transfer processes are graphically explained in Figure 9.56. It has to be said that the U.S. **Environmental Protection Agency** (**EPA**) considers this type of renewable energy as "*the most energy-efficient, environmentally clean, and cost-effective space conditioning systems available today*" [81]. Traditionally, geothermal energy is used to bypass electric energy [382], while this study tries to coordinate all the commercial electric and non-electric energy sources into the next generation smart grid.

If the xth entity is reinforced with this geothermal energy system, then only

water pipelines can be used as a trading medium with other entities. However, the property owner can hit two birds with one stone. That is, hot water can be supplied to the yth entity during the winter season, and cool water can be supplied during the summer season. But, before doing that, there is an open question about the most efficient way to trade this kind of non-electric energy forms. *Should the seller send the hot/cool water directly to the buyer or after transforming it to a hot/cool air through an indoor heat-exchanger and a duct system? Does the distance between the xth and yth entities play an important role in selecting this alternative? What if the buyer wants hot water instead? What about the installation/operating/maintenance costs of GHP?* Thus, there is a need to conduct more studies on these crucial points by joining electrical, electronics, instruments and control, mechanical, chemical, and civil engineering disciplines.

9.6.3.5 Other Energy Sources

There are some other possible non-electric energy sources can be seen in the smart grid. For example, biogas/biofuel plants (such as biodiesel, **biobutanol**, **biomethanol**, bioethanol, and **synthesis gas**), marine energy (such as marine current, thermal, osmotic, wave, and tidal), and solar furnaces. Most of these renewable sources are exclusively integrated into some big entities, like power companies and industries. However, they are listed here just to complete the list and to mention that all the known non-electric energy forms can be traded; either between local houses in micro-grids or between local houses and other big entities in smart grids.

9.6.4 Further Discussion

To maximize the benefits of trading non-electric energy forms, it is important to check all the possible alternatives before doing that mission. It has been seen that many alternatives can be used to trade the preceding energy forms in the next generation smart and micro-grids. Thus, the best trading approach can be selected based on a single or multi-objective function. Such objectives are price/cost, efficiency, greenness, aesthetics, and safety. Also, the "source availability" criterion plays an important role in that selection process. Thus, the approaches shown in Figures 9.45, 9.46, 9.53, and Figure 9.54 could be offered with very cheap prices, but the buyer is forced to

go to the nearest PLW because the heating system installed in his/her property is a wood fireplace.

To check the feasibility of these new trading concepts and whether they are viable and applicable in the near future, then different engineering disciplines must be involved to examine their strengths, weaknesses, and possible applications as well as their sensitivity and vulnerability to different design coefficients, parameters, and variables.

9.6.4.1 Off-Grid Non-Electric-Based Networks

The other interesting application is about the possibility to establish **off-grid networks** that can be energized by pure non-electric energy sources. This feature can be easily applied in micro-grids or even partially between neighbors. Thus, homes that rely entirely on electricity can be heated by non-electric renewable sources provided from nearby homes equipped with such sources.

9.7 Using Linear Heat Sensors as Very Cheap Alternative to Existing Electric Power Protective Devices

The common devices used in power system protection are [15, 24, 74, 123, 167, 292]:

1. Non-electrical relays (such as: bimetallic, Buchholz, and pressure relief relays);
2. Electrical relays (such as: distance, overcurrent, differential, over/under voltage, over/under frequency, reverse power, and overflux relays);
3. Reclosers;
4. Sectionalizers; and
5. Fuses.

This study focuses on the last protective type; i.e., using fuses. The question raised here is about the possibility to replace these compact protective devices with a simpler device borrowed from **fire and gas** (**F&G**) **systems**. The goal of this study is to merge one of the techniques used in F&G systems into electric power system protection. The technique implemented here is based on detecting fires by using **linear heat sensors** (**LHS**). This study is an attempt to make LHS applicable to protect electric power components.

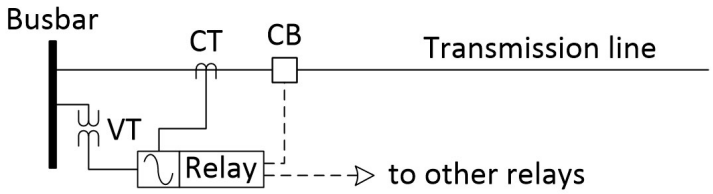


Figure 9.57: Illustrated 1ϕ Protection System

9.7.1 Fuses and Thermal Relays

Electric protective relays are widely used in power systems. However, there are other devices can be used to protect some specific electric components. Such devices are Buchholz relays, pressure relief relays, fuses, and bimetallic relays. The first two protective devices have limited applications, and they are extensively described in [315]. The last two protective devices are explained below.

9.7.1.1 Fuses

Fuses are considered as the oldest protective devices that are still used nowadays [292]. This is because of their simple construction and basic working principle. To clarify this point, let's see the major elements of a protective relay shown in Figure 9.57 [24]. If a short-circuit fault occurs on that transmission line, then the fundamental current and voltage signals are respectively processed through a current transformer (CT) and a potential transformer (PT). In fact, there are two stages to step down these current and voltage quantities. The first one or the huge stepping down part is done through the preceding main CT and PT, while the second stage is done through auxiliary CT and PT located inside the protective relay. Moreover, some issues have to be taken into consideration, such as decaying DC components, harmonics, and CT saturation. If a numerical relay is used, then it is programmed with a special algorithm to compare the incoming signals and detect any possible fault. Finally, a trip signal is sent from that protective relay to its circuit breaker (CB) to interrupt the abnormal flow of the short-circuit current.

Instead, all the preceding essential stages (sensing, comparison/detection, and interruption) are combined into one compact device called a fuse. The most distinct properties of this protective device are its speed and price compared with all the

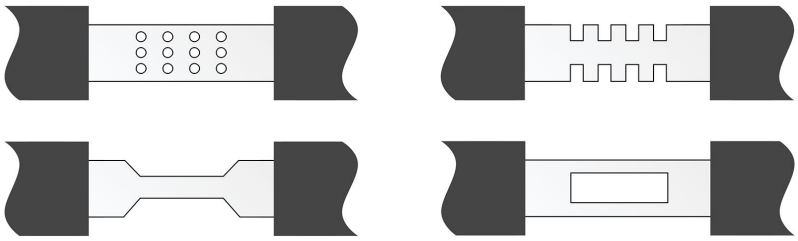


Figure 9.58: Different Weak Link Regions of Fuses' Internal Structure

known types of relays. Also, fuses do not need to use any CTs, PTs, or even CBs to do their jobs.

The working principle of fuses is very simple. If an electrical current flows through a conductive material, a portion of that energy is dissipated as thermal energy. The amount of that energy is proportional to the capacity and duration of the flowing electrons. By deforming a part of that material in a way that the energy absorption process is concentrated in a weak link (i.e., its area is less than that of the end regions), then that link will be melted and vaporized once the magnitude and duration of the flowing current exceed the threshold level. This phenomenon happens during shunt faults where the fault currents steeply increase due to low impedances of their short paths. Figure 9.58 shows four possible shapes of the previously described weak link [251].

The main disadvantages of these unique protective devices are their uncontrollable operating time and the irreversible interrupting action. Thus, if the fuse weak link is melted, then the whole fuse unit needs to be replaced with a new one. Some good theoretical and practical descriptions about fuses can be found in [105, 164, 251].

As a summary, fuses are considered as very fast non-directional and non-adjustable overcurrent protective devices. There are many types of fuses available in the market, which can be classified based on the operating time, voltage and insulation level, type of system, maximum fault current level, and load current. Different classes are used in the market to describe the application(s) of each type, such as **RK1**, **RK5**, **CC**, **T**, **K**, **G**, **J**, **L**, and **R** [105, 147]. Also, some technical problems could be faced with fuses, especially when coordinating them with other protective devices. Some of these problems are reported in [118, 150]. There are some practical guidelines and recommendations need to be followed during installing such devices [52, 105, 147].

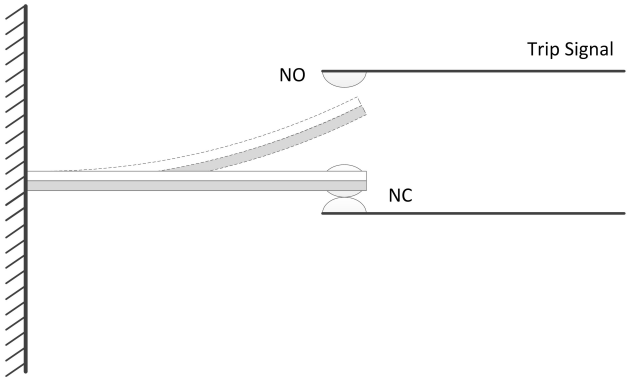


Figure 9.59: Bimetallic Relay

9.7.1.2 Bimetallic Relays

These devices work on a very basic physical phenomenon where the body expansion increases proportionally with the applied temperature. If two different materials with different expansion rates are combined together[35], then if these two flat strips are heated above their normal temperature they will bend one way, and in the opposite direction if they are cooled below their normal temperature. Thus, if the right-ends of these two strips are equipped with some contacts as shown in Figure 9.59, then a trip signal can be initiated by the normally-open (NO) contact once the thermal energy exceeds the allowable limit and duration.

The main problem associated with this type of protective devices is its slow response, and thus bimetallic relays are primarily used to protect motors against long overloading condition [292]. This fatal drawback restricts their applications in electric power systems. Thus, for generation and transmission, bimetallic relays cannot be used because a quick isolation process is required. Based on this, these protective devices cannot be considered as proper candidates and viable alternatives to fuses.

The operational philosophy of other overcurrent protective devices, such as reclosers and sectionalizers, can be found in [46, 74, 105, 147, 167, 315].

9.7.2 Linear Heat Sensors

It has been seen different techniques that are used to protect power system components. Also, it has been seen why bimetallic relays cannot be used as a practical

[35] They can be attached by welding, brazing, or riveting.

replacement of fuses, because of their slow time response. Fuses themselves have some disadvantages. It is well-known, in the literature and industries, that these compact devices are cheaper than protective relays. The word "cheaper" does not mean that the fuses are not expensive. To be more precise, a non-ignorable annual budget is required to purchase and replace such fuses in - almost - daily routine jobs of all power industries. For example, the fusible element of some types are made from silver and some of them are electronic fuses [105, 150]. Also, a significant amount of time is required for their maintenance, and only experienced technicians can replace these devices after isolating all live electric components connected to them.

All the above practical points motivate us to search for other possible ways to protect power system components with competitive price, performance, and features. The journey started by understanding the basis of heat. This physical quantity can be produced by different transformations of energy. One of these energy transformations occurs when an electric current flows through a conductor; i.e., an electrical to thermal energy transformation process. This physical phenomenon is the backbone of many overcurrent protective devices. The other type is called a chemical to thermal energy transformation process, which can be achieved by the mean of fire. With the presence of oxygen (or any other oxidizer), the preceding energy transformation is a continuous process where the fuel is burned if there is enough heat, and vice versa. This process is explained by what is called the **fire triangle**[36], which is shown in Figure 9.60. This triangle tries to say that most of the fires are formed by these fundamental ingredients; fuel, oxidizing agent, and heat. Thus, with the presence of oxygen, fires can be avoided by breaking one of the remaining two arms; i.e., either removing the fuel or keeping the site very cool. That is, when a fire is ignited it will start producing heat until the entire oxygen or fuel is consumed. All these points are covered in fire and gas (F&G) systems. F&G is an active branch of instrumentation and control (C&I) engineering. It deals with all the aspects related to sensors, detectors, special wirings, monitoring systems, fighting systems, and alarm systems of both fires and gases.

Therefore, this chemical to thermal energy transformation process inspired us to think about the possibility to borrow the techniques used in F&G systems and

[36]It is also known as the **combustion triangle** and the **fire diamond**.

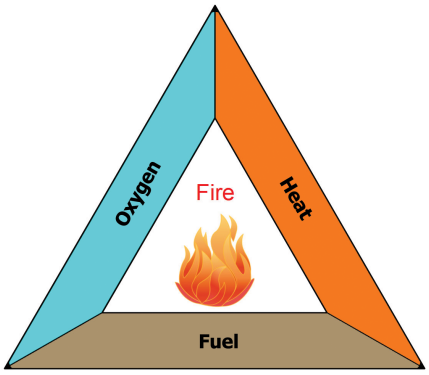


Figure 9.60: Fire or Combustion Triangle

employing them into electric power system protection. The point shared between these two different disciplines is the last phase of energy transformations, which is the heat. The first discipline prevents the heat from being generated. This could be done by fighting fires, smokes, and flammable gases, and keeping the temperature at a low level. The other discipline protects the power systems from the excessive heat generated by faults.

One of the simplest and most effective heat-based techniques used in F&G systems is called a linear heat sensor (LHS). It is also known as a **linear heat detection (LHD) cable**, a **linear detection wire**, or just a **linear heat**. The working principle of LHS is explained in Figure 9.61. That is, if a fire is ignited near LHS, then the heat generated by that fire will push the thermal sensitive **polymers** of LHS to break down, which causes a short circuit between the two internal conductors. These two conductors are used as alarm contacts, so when they touch each other a fire alarm signal will be initiated.

The logical question that should be raised here is: *What prevents the use of this*

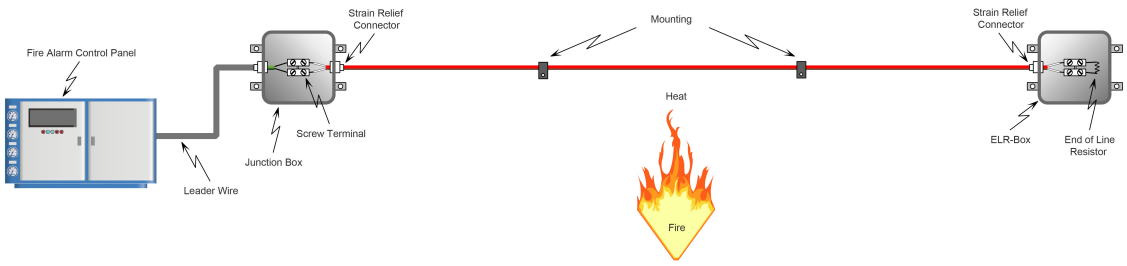
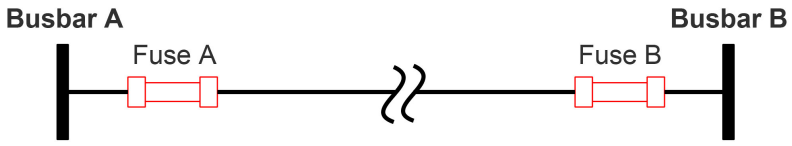
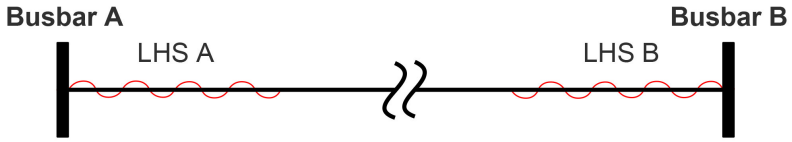


Figure 9.61: Illustrated Working Principle of Linear Heat Sensors Used in F&G Systems



(a) Traditional Protection Using Fuses



(b) Proposed Protection Using LHS

Figure 9.62: Comparison Between the Working Principles of Fuses and LHS

technique to assist with power system protection applications? It is very cheap, simple, small, and does not require any special precaution during installation and maintenance phases. The proposed concept can be graphically explained in Figure 9.62. The fuses A and B are inserted as a medium between the branch and the busbars. Thus, they can detect any short-circuit fault by sensing and comparing the heat generated by the flowing current. Then, they isolate that fault by letting their weak links to melt and vaporize. On the opposite side, the two LHS work on the same principle of fuses, but they are attached on the surface of the line. Thus, when there is a fault on the line, LHS can detect it externally without a permanent interruption of the powerline as it happens with fuses.

Currently, LHS or LHD cables are used in the following F&G applications [322]:

- Cable trenches and tunnels.
- Outdoor areas and parking lots.
- Fuel stations and storage areas.
- On the top of industrial machinery.
- All other areas where installing conventional fire detectors is not possible due to harsh weather and air pollution.

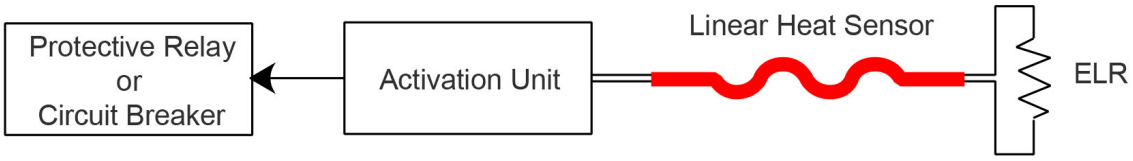


Figure 9.63: Closed-Loop Approach for LHS

Thus, it has been wondering why such cheap and effective sensors are not employed in electric power system applications! Even the first application, the purpose of installing LHS in cable trenches and tunnels is to prevent fires not to prevent electric faults.

9.7.2.1 Protecting Power Components Against Faults via LHS

In Figure 9.62, it has been seen how to replace fuses with LHS. However, to be able to do that, there are two possible approaches:

Closed-Loop Approach

In this approach, the two internal conductors (i.e., the contacts) of LHS are terminated by a resistor called the **end of line resistor** (**ELR**); as seen before in Figure 9.61. If this approach is selected, then its main stages can be graphically described in Figure 9.63.

Thus, a separate activation unit is required to translate the short-circuit action of LHS contacts to a trip signal. A very simple and basic activation circuit is shown in Figure 9.64. However, a more advanced electronic circuit should be used for high-speed switching.

The main drawbacks of this approach are: 1. higher installation and maintenance costs, 2. external power unit, 3. extra space due to increase in size, and 4. more processing time.

Open-Loop Approach

To overcome all the technical issues associated with the preceding approach, the open-loop approach shown in Figure 9.65 is used instead.

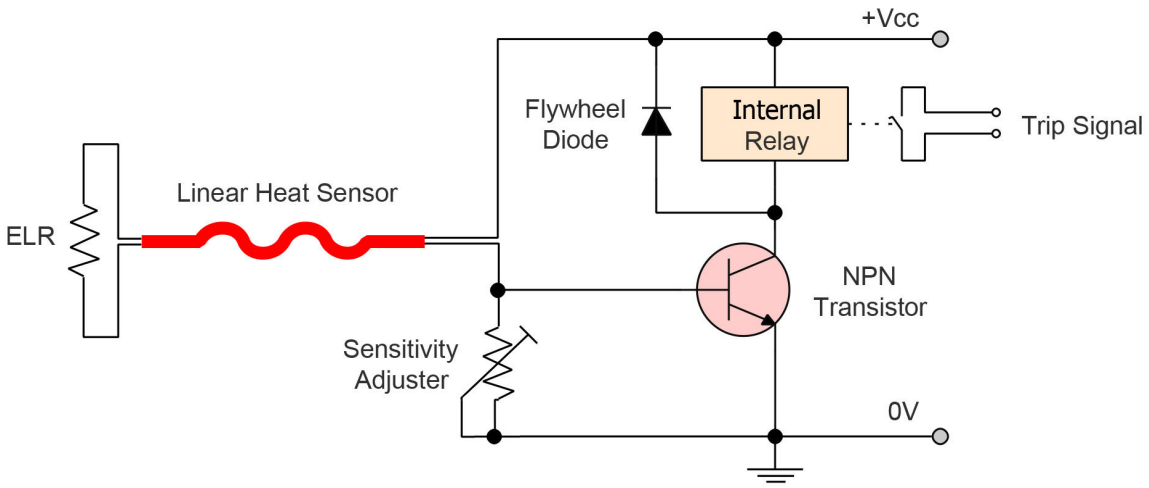


Figure 9.64: Basic Activation Circuit for LHS

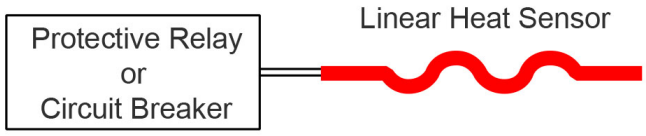


Figure 9.65: Open-Loop Approach for LHS

That is, the two conductors of LHS are used as contacts of relays or CBs. To accelerate the isolation speed, LHS should be directly connected to CBs.

9.7.2.2 Advantages of Using LHS

There are many advantages when using LHS, such as:

- LHS can be used for transmission, sub-transmission, and distribution lines; as well as other applications.
- They are easy to be installed and replaced; even on live powerlines.
- The protected components are not permanently isolated from the network as with fuses, because the interruption stage belongs to CBs.
- LHS are very cheap, thin, light, compact, and small elements.
- Multiple LHS can also be installed on the same spot to provide a backup protection[37].

[37] i.e., applying the 2oo3 voting logic. Please, refer to the second last subsection.

Table 9.25: Technical Specifications and Commercial Price List of Two Industrial Power Fuses

Product Attribute	**Fuse 1**	**Fuse 2**
Manufacturer:	Eaton	Littelfuse
Fuse Type:	Time Delay / Slow Blow	
Product:	Class L	
Model Number:	KRP-C-6000SP	KLPC6000X
Body Style:	Cartridge Blade Fuses	
Mounting Style:	Holder	
Termination Style:	Knife Blade	
Indicator Style:	Without Indicator	
Brand:	Bussmann / Eaton	Littelfuse
Length:	10 3/4 in	
Width:	7 1/8 in	
Unit Weight:	15.241 kg	
Unit Price[a]:	9,710.14 CAD	5,710.87 CAD

[a]They are taken from Mouser Electronics, Inc. website [mouser.com] - September 4, 2019.

- Similar to fuses, LHS can be designed with different fault currents and durations.
- The length of LHS can be very small, or it can be laid up to 10,000 feet (3,045 meters) per zone.
- LHS can be directly connected to CBs to accelerate the isolation speed.
- Also, they can be connected to protective relays to provide more information about the occurred fault to the corresponding energy management system (EMS).

Fuses are the cheapest available protective devices. There is no doubt about that. However, Table 9.25 gives a general picture of some technical specifications and commercial price list of two industrial power fuses that come with ratings of 600 VAC and 6 kA and an interrupting rating of 300 kA.

Table 9.26: Technical Specifications and Commercial Price List of Three Linear Heat Detection Cables

Product Attribute	**LHS 1**	**LHS No.2**	**LHS 3**
Manufacturer:		Eurofyre	
Brand:		FyreLine	
System Compatibility:		Digital	
Model Number:	EF-TH68-100	EF-TH88N-100	EF-TH88B-100
Cable Composition[a]:	Type I	Type II	Type III
Detection Temperature:	{68,88,105} °C	{88,105} °C	{88,105} °C
Alarm Temperature:		+45 °C to +185 °C	
Ambient Temperature:		-45 °C to +125 °C	
RFI or EMC:		Not Affected	
Cable Length:		100 meter	
Price[b]:	688.35 CAD	825.87 CAD	2,385.47 CAD

[a]Type I: standard outer sheath, Type II: enhanced corrosion and abrasion resistant, and Type III: grade 316 stainless steel.

[b]Taken from Discount Fire Supplies website [discountfiresupplies.co.uk], including value-added tax (VAT) - Sept 4, 2019.

The last row of Table 9.25 gives convincing evidence that the preceding prices can not be tolerated even though they are considered rather cheap compared with other power protective devices. The frustrating thing here is when there is a malfunction where the whole fuse unit needs to be changed. On the opposite side, Table 9.26 shows the price list of three different LHS or LHD cables.

It can be clearly seen that the LHS/LHD cables are very cheap. Moreover, small pieces can be used, which means that tens of LHS can be made from that 100-meter length cable. For example: 100 LHS of 1-meter length, which means that the prices listed in Table 9.26 can be reduced by $\times 100$ times! In F&G systems, three main options can be provided to replace the standard **PVC** outer jacket of LHS [328]:

- **Nylon** - to increase its protection against **UV**; which is mainly used for harsh industrial environments.

- **Polypropylene** - to increase its protection against harsh and caustic environments.
- **Guidewire** - 12 AWG stainless steel wire to support long LHD cables.

9.7.3 Further Discussion

In spite of all the facts stated before regarding LHS, still, they cannot be used in their existing forms because they are designed exclusively for F&G systems. Thus, to be able to use LHS as a protective device in power systems, then the following points could help in making it feasible and viable:

- **Interrupting current** can be satisfied by selecting the proper polymer type.
- **Detection speed** can be controlled by choosing the optimal thickness of the polymer.
- A "2 out of 3 (2oo3)" voting logic could be adopted by installing three small LHS on the same spot. Thus, the trip signal is initiated only if, at least, two LHS detect that fault. This strategy is very useful to ignore trip signals coming from malfunctioning LHS, and thus the system dependability can be enhanced.

The preceding three important points give the opportunity to conduct more research on this new promising area.

9.8 Locating Faults in Mesh Networks by Distance-Time Characteristic Curves Designed for Directional Overcurrent Relays Using Interpolation- and Regression-Based Models

The three principal parts of electric power systems[38] are exposed to different types of faults. They can be seen as passive (such as over-loading, over-voltage, power swing, and under-frequency) or active (i.e., real) faults. In general, the real faults are categorized into two broad types called open-circuit (or series) faults and short-circuit (or shunt) faults. Thus, for active short-circuit faults, the current is passed to the ground. This phenomenon happens if any one of the five possibilities shown in Figure 9.66 exists. In practice, the phase-to-ground fault is the most probable type,

[38] i.e., generation, transmission, and distribution.

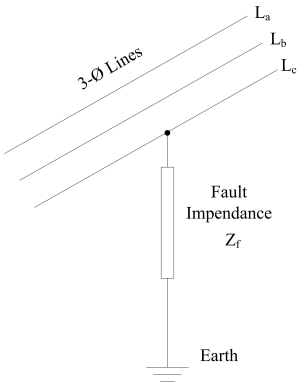
whereas the three-phase (3ϕ) fault is the most severe type as listed in Table 9.27 [65, 74]. These five types are classified into two groups or classifications as follows:

1. **Symmetrical** (or **balanced**) faults: in this classification, the three phases have the same voltage and current sinusoidal waves that are shifted by 120° between each other. Thus, the self and mutual impedances of these three lines are also equal. The last two possibilities given in Figure 9.66 (i.e., 3ϕ faults) are symmetrical.
2. **Asymmetrical** (or **unbalanced**) faults: the remaining possibilities given in Fig. 9.66 are classified as unbalanced faults. In these fault types, the phases are affected unequally, and thus the magnitudes and angles of their voltages, currents, and impedances are not equal anymore.

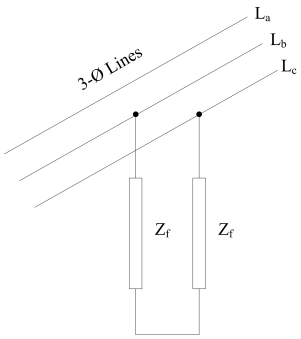
Moreover, these symmetrical and asymmetrical faults could happen in underground cables, overhead lines, **joint-nodes**[39], series-compensated lines, generators, transformers, **switchgear**, PTs, CTs, etc [74, 158, 329]. If any one of these faults occurred, then its duration will decide whether it is a **transient fault** (also called a **temporary fault**) or a **persistent fault** (also called a **permanent fault**). In many cases, the transient faults occur in overhead transmission/sub-transmission lines, while the faults occur in underground cables are likely to be persistent. In practice, an automatic recloser is used to return any temporary faulty line back to service. Table 9.28 shows the probability of faults occurrence on different elements of electric power systems [292]. It shows that 50% of faults occur in overhead lines, which is one of the reasons why solving ORC problems becomes a very hot topic nowadays. This study covers the following stream: mesh networks → active → shunt → symmetrical and asymmetrical → persistent faults.

Modern electric power systems are equipped with advanced tools to process all the preceding faults through several stages to maintain reliable and stable operation. These stages are summarized as: 1. **fault detection**, 2. **fault classification**, 3. **fault location**, 4. **fault containment**, and 5. **fault recovery** [92, 375]. This study mainly focuses on the first three stages where the last two stages are parts of power system stability and control [46, 147, 213]. The relation between the first three stages is graphically described in Figure 9.67 [92].

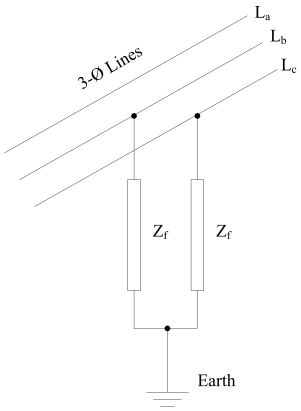
[39] They are mixed of overhead lines and underground cables.



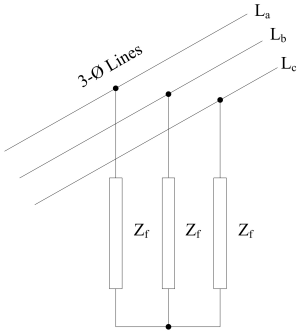
(a) L-G fault



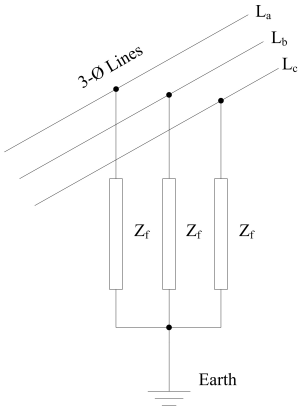
(b) L-L fault



(c) L-L-G fault



(d) L-L-L fault



(e) L-L-L-G fault

Figure 9.66: Five Possible Conditions of Active Short-Circuit Faults. The Letters L and G Stand for "Line" and "Ground", Respectively

Table 9.27: Probability of Faults Occurrence Corresponding to Their Types [65, 74]

Fault Type	Probability (%)	Severity Level
Single phase-to-ground	70 - 80	Lowest
Phase-to-phase	10 - 8	
Phase-to-phase-to-ground	10 - 17	
Three-phase	2 - 3	Highest

Table 9.28: Probability of Faults Occurrence on Different Power Elements [292]

Power System Element	Probability of Occurrence (%)
Overhead lines	50
Underground cables	9
Transformers	10
Generators	7
Switchgears	12
CT, PT, etc	12

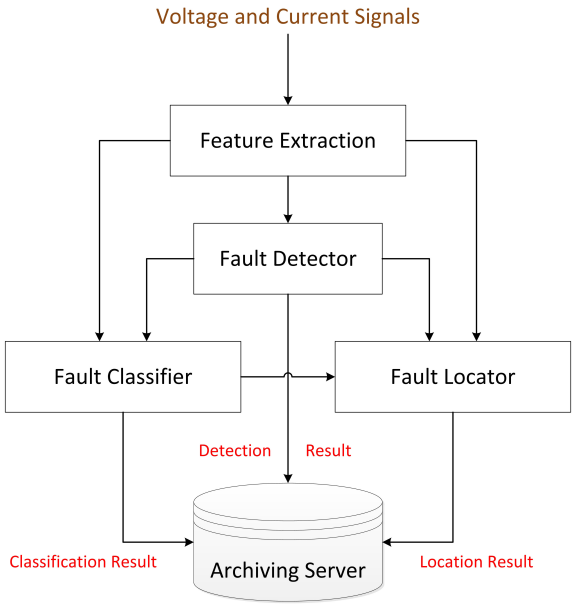


Figure 9.67: Simplified Framework for Fault Detection, Classification, and Location

The third stage, i.e. fault location, is the core of this study. It can be defined as a process to precisely locate faults in electric power systems. This process can be achieved by implementing many algorithms where the fault location function itself can be implemented as stand-alone fault locators, **digital fault recorders** (**DFRs**), or even just as post-fault analysis programs inside the state-of-the-art numerical protective relays [329]. Nowadays, the market contains different types of fault locators where each one of them has its own capabilities and limitations. Such devices are: **Murray loop bridge**, **A-Frame**, **time domain reflectometer** (**TDR**), **wheatstone bridge**, **high voltage** "**Thumpers**", **integrated test sets**, etc [361]. Although modern protective relays can be upgraded to act as fault locators[40], it has to be remembered that the main purpose of protective relays is to protect electric power components against any fault as fast and selective as possible. That is, protective relays are responsible to carry-out the first stage of Figure 9.67; i.e. fault detection. To clarify this essential point, Table 9.29 lists the main differences between fault locators and protective relays.

Powerlines (transmission, sub-transmission, and distribution lines) come with two or more terminals. Based on the number of terminals, powerlines can be protected by: 1. **one-end**, 2. **two-end**, or 3. **multi-end algorithms** [92, 261, 329]. The algorithms built based on the first approach are very simple and they do not require to use any communication link since the information received at the far-end terminal is not utilized. This is the reason why the fault location function can be embedded in modern numerical relays if the one-end approach is used. If the two-end algorithms are used to locate faults on faulty lines, then the information received from both ends (i.e., near-end and far-end terminals) should be processed. This job can be accomplished by providing a communication link between the two terminals. A GPS antenna can also be installed to provide a time reference pulse to synchronize both relays, which is very useful in some applications. Therefore, the algorithms of the two-end approach consume more computation time than those of the first approach. With multi-terminal lines, the multi-end algorithms must be used to locate faults, which are more complicated than the preceding two approaches [113].

[40]Because both devices have the same hardware.

Table 9.29: Main Differences Between Protective Relays and Fault Locators [329]

Fault Locators	**Protective Relays**
larger data window	smaller data window
more accurate	less accurate
slower	faster

The main goal of this study is to design a new two-end algorithm to locate faults in mesh networks based on the capabilities of modern numerical protective relays. This can be accomplished by utilizing the features that can be provided by modern EMS. For example, protection engineers and power operators can perform different operations, such as changing relays' settings and retrieving their online data remotely through some common communication protocols used today in numerical relays. Such protocols are **IEC 608750-5**, Modbus, **MMS/UCA2**, **Courier**, and **DNP** [147]. Therefore, this feature can be activated to import some important data from the two-end relays of a faulty line. Such data are the operating times, short-circuit currents and fault type, which can be effectively utilized to estimate the location of that fault.

For that mission, optimally coordinated inverse-time directional overcurrent relays are used. The reason for that selection has been extensively discussed in Chapter 4. Thus, to continue reading this section, it is important to refer to that chapter to know the mathematical modeling behind ORC problems.

As a summary, this technique utilizes the online data received from numerical DOCRs installed on both ends of each line. Different approaches have been tried to reach the final highly precise and accurate models. These approaches are classical linear interpolations, **logarithmic-based nonlinear interpolations**, polynomial regressions, and nonlinear regressions. To solve many technical issues associated with the preceding approaches, a new nonlinear regression approach is proposed in this study and it is called a distance-time characteristic curve (DTCC), which comes with two versions.

The following points list the main achievements done in this study:

1. Merging between the ORC and fault location topics to build an integrated algorithm that can estimate the location of faults by utilizing the information retrieved from both end numerical DOCRs.

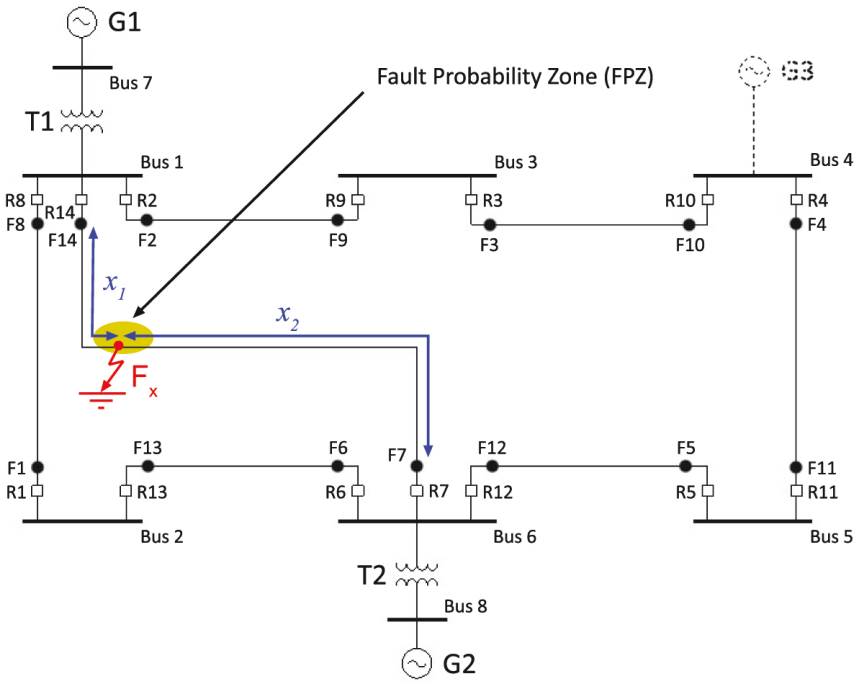


Figure 9.68: Single Line Diagram of the IEEE 8-Bus Test System

2. Although the correct settings of TMS and PS of all DOCRs are obtained by using an optimization algorithm, this task is predefined and it belongs to the ORC stage. Thus, the proposed fault locator is an optimization-free technique.

3. For the nonlinear regression part, the coefficients are optimally tuned only once during building the regression model. This process is done off-line before the fault locator is put into service.

4. Although the approach of the DTCC nonlinear regression can provide great models, there are also four other attempts to approximate the actual behavior of each two-end relays. Therefore, the study triggers many points and open the door wide to conduct more researches on this topic.

The performance of this novel fault locator is evaluated by applying it to estimate fault locations of a faulty line of the IEEE 8-bus test system. The mechanism of the technique is described in detail with considering different scenarios and fault types.

9.8.1 Mechanism of the Proposed Fault Location Algorithm

Chapter 4 covers the fundamental steps behind the mechanism of this fault location algorithm. Now, it is time to describe how it works. Figure 9.68 shows the IEEE 8-bus test system, which is one of the most popular ORC test systems presented in the literature [25]. Although it consists of 8 buses and 9 branches, only the buses and branches located after the two transformers are taken into account during solving its ORC problem [40,51,280]. The network contains 7 branches. Because each branch is protected by two DOCRs, so the total is 14 DOCRs. Based on (4.2), the dimension of this ORC problem is:

$$n = 2\varrho = 4ß = 2(14) = 4(7) = 28$$

The bold dot denoted beside each relay represents the near-end fault point, which is considered as the far-end point of the relay installed on the other terminal of the line. To clarify this point on the IEEE 8-bus test system, let's take bus 1 and bus 6 as bus x and bus y of Figure 4.2. By analogy, R_x and R_y are respectively R_{14} and R_7. Also, F_x and F_y are respectively F_{14} and F_7. Thus, R_{14} sees F_{14} as the near-end fault and R_7 sees it as the far-end fault, and vice versa for F_7.

Now, assume that a shunt fault F_x occurs on the line between bus 1 and bus 6 (i.e., L_{16}). Thus, the protective relays R_7 and R_{14} see F_x as an in-zone fault, so they act as primary relays for that fault. The backup relays R_1, R_5, R_9, and R_{13} see F_x as an out-zone fault, where R_5 and R_{13} operate when R_7 fails and R_1 and R_9 operate when R_{14} fails. From L_{16}, the length x_1 represents the distance between the near-end fault F_{14} and the actual fault F_x when it is seen by R_{14}, and the length x_2 represents the distance between the near-end fault F_7 and the actual fault F_x when it is seen by R_7. Also, it can be said that the length x_1 is the distance between F_x and the far-end fault F_{14} when it is seen by R_7, and the length x_2 is the distance between F_x and the far-end fault F_7 when it is seen by R_{14}. However, we will stick with the first definition. Thus, the total length of the line L_{16} can be approximated as follows [10]:

$$\text{length}(L_{16}) = l_{L_{16}} \approx x_1 + x_2 \tag{9.96}$$

From (9.96), it is clear that the sum of x_1 and x_2 measured by the two-end relays does not necessarily be the same length of L_{16}. There are many sources of uncertainty

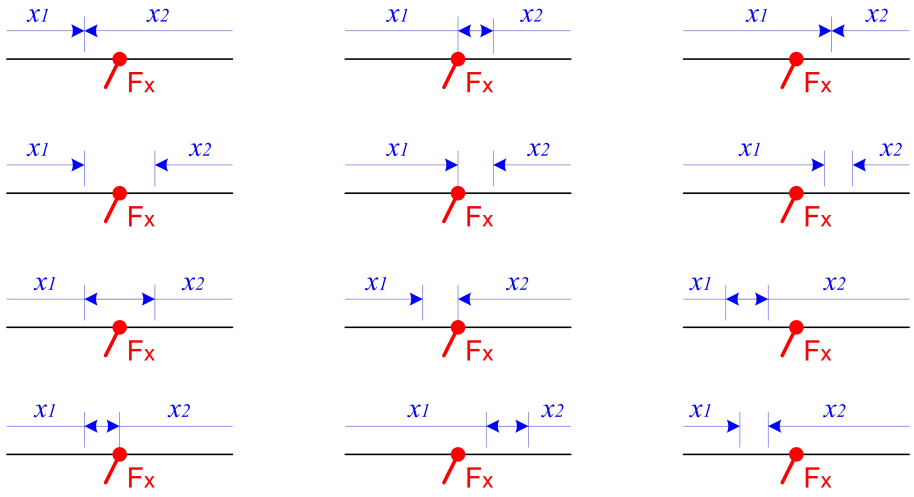


Figure 9.69: Twelve Possible Fault Probability Zones

behind this phenomenon, such as the inherent errors of CTs, PTs, and relays [46,339]. Twelve possible **fault probability zones** (**FPZs**) can be faced during calculating the distance between the actual fault location and its estimate where the destination of x_2 will not meet the destination of x_1 at F_x. These zones are graphically shown in Figure 9.69. By returning to Figure 9.68, these zones are created because of the errors associated with R_7 and R_{14}. Therefore, it is useful to take the average value of these two estimates to reduce the overall error as follows:

$$\bar{x}_{\text{bus 1}} = \frac{x_1 + (l_{L_{16}} - x_2)}{2} \tag{9.97}$$

$$\bar{x}_{\text{bus 6}} = \frac{x_2 + (l_{L_{16}} - x_1)}{2} \tag{9.98}$$

where $\bar{x}_{\text{bus 1}}$ is the distance estimated from bus 1 to F_x, and $\bar{x}_{\text{bus 6}}$ is the distance estimated from bus 6 to F_x. The estimates x_1 and x_2 are calculated from R_{14} and R_7, respectively. In this study, bus 1 is selected as a reference, so (9.97) should be used. For the sake of simplicity, the notation $\bar{x}_{\text{bus 1}}$ is replaced with $\bar{x}$ and the term $(l_{L_{16}} - x_2)$ is replaced with $\tilde{x}_2$, so the average estimate given in (9.97) becomes:

$$\bar{x} = \frac{x_1 + \tilde{x}_2}{2} \tag{9.99}$$

The goal is to find a direct relationship between the location of F_x and the operating times of the two-end relays, which are here R_7 and R_{14}. The operating time of the two-end relays can be calculated using either (4.4) or (4.5) based on the standard

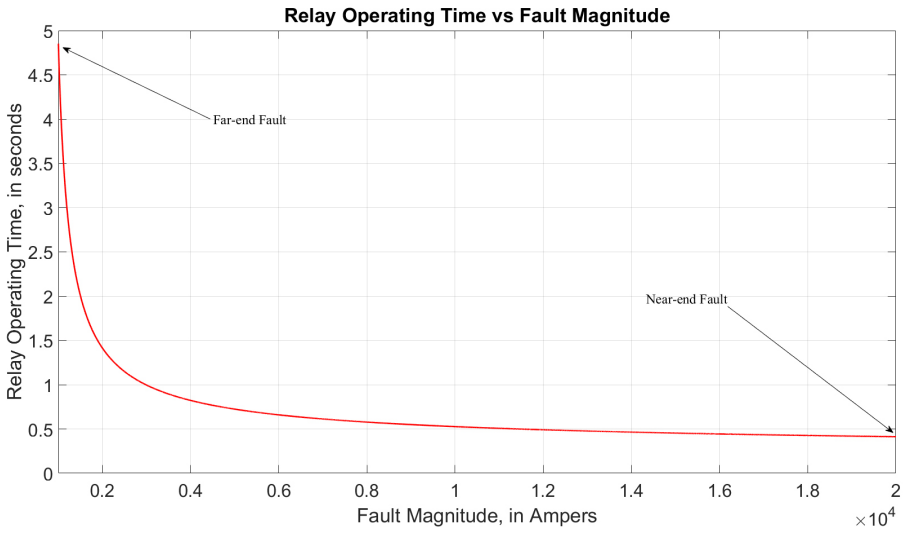


Figure 9.70: IEC IDMT-Based TCCC for Different Fault Magnitude Values

TCCC used; as tabulated in Table 4.2 and shown in Figure 4.7. These equations can determine the operating time of the ith relay based on the fault type and its value. Figure 9.70 shows how the IEC standard inverse TCCC behaves at different fault magnitudes. Thus, the operating times of these two relays, i.e. R_7 and R_{14}, are inversely proportional to the fault magnitude. This magnitude depends on the fault type, and the operating time depends on the values of $\{TMS, PS, \alpha, \beta, \gamma\}$.

By focusing on the bold dots shown in Figure 9.68, let's say that the lowest and highest possible operating times of R_7 are $T_{R_7}^{\min} = T_{R_7}^{F_7}$ and $T_{R_7}^{\max} = T_{R_{14}}^{F_{14}}$, respectively. For the other end relay, which is here R_{14}, its lowest and highest possible operating times are then $T_{R_{14}}^{\min} = T_{R_{14}}^{F_{14}}$ and $T_{R_{14}}^{\max} = T_{R_{14}}^{F_7}$, respectively. Modern numerical protective relays can detect the time, value, and type of any fault. After that, they can store and send all these data automatically through the preceding protocols. For example, **AREVA MiCOM P12x OCRs** can detect all the five possible fault types and identify them by the code F80 [55]. Thus, the operating time T_{R_i} is a function of the fault magnitude I_{R_i,F_x} seen by the ith relay, where the value of I_{R_i,F_x} and the type of F_x can be provided too. Also, it has been seen that the fault magnitude I_{R_i,F_x} is a function of the fault location x. This relation can be mathematically expressed as follows:

$$I_{R_i,F_x} = I_{R_i}(x) \tag{9.100}$$

By substituting (9.100) in (4.4) or (4.5), the operating time of the ith relay can then be indirectly represented as follows:

$$T_{R_i}\left(I_{R_i}\left(x\right)\right) \equiv T_{R_i}\left(x\right) \tag{9.101}$$

The final step is to find the inverse function, so the location can be estimated as follows:

$$T_{R_i}\left(x\right) \Rightarrow x\left(T_{R_i}\right) \tag{9.102}$$

Based on this, a relationship can be made between the operating times of the two-end DOCRs and the location of the fault F_x. However, this method is not simple, because it is nonlinear and there are many sources of error due to the calculation, communication, and final data processing of the first three stages shown in Figure 9.67. Also, there are other sources of error due to some uncertainties that could be faced on actual electric networks. The next paragraphs describe the five approaches used in this study to estimate fault locations.

9.8.1.1 Approach No.1: Classical Linear Interpolation

The linear interpolation method estimates the new jth point x_j by constructing a linear curve between two predetermined points. This approach is the simplest one. However, there is a significant error in its estimation. The reason behind this error comes from the nonlinearity nature of the relationship between the fault location x and the operating times of the corresponding two-end relays. The classical linear interpolation can be carried out for the ith relay using the following equation:

$$x_j = x_j^{\min} + \left(x_j^{\max} - x_j^{\min}\right)\left[\frac{T_{R_i}^{F_x} - T_{R_i}^{\min}}{T_{R_i}^{\max} - T_{R_i}^{\min}}\right] \tag{9.103}$$

where $x_j^{\min}$ and $x_j^{\max}$ are respectively the minimum and maximum distances measured from the jth terminal.

The estimate x_j is computed based on an assumption that all the points between $x_j^{\min}$ and $x_j^{\max}$ form a straight line, which is not correct for any ITOCR. In (9.103), the reference point is $x_j^{\min}$ for estimating the new point x_j. Thus, if $x_j^{\max}$ is taken as the reference point, then the alternative equation that can also be used is:

$$x_j = x_j^{\max} - \left(x_j^{\max} - x_j^{\min}\right)\left[\frac{T_{R_i}^{\max} - T_{R_i}^{F_x}}{T_{R_i}^{\max} - T_{R_i}^{\min}}\right] \tag{9.104}$$

In this study, (9.103) is used to estimate x_j. For the operating time of the other end relay, there is one tricky point that should be considered. In Figure 4.2, the operating time of R_x increases from the minimum value at x to the maximum value at y, while the operating time of R_y decreases as the fault location moves from x to y. Thus, to avoid using (9.97), the estimate of the other end relay can be directly computed as follows:

$$\tilde{x}_j = x_j^{\min} + \left(x_j^{\max} - x_j^{\min}\right) \left[\frac{T_{R_i}^{F_x} - T_{R_i}^{\max}}{T_{R_i}^{\min} - T_{R_i}^{\max}}\right] \tag{9.105}$$

Thus, there is no need to subtract the estimate from the total length. That is, (9.99) is involved here for the other end relay.

It has to be said that (9.103) should be carried out for each two-end relays. The final estimate of the exact location is then computed by using either (9.97) or (9.98); based on which bus is selected as a reference. If (9.97) is selected, then (9.99) is preferred.

The main problem associated with (9.103) is the accuracy. The amount of error generated by this static approach can be significantly minimized by using a more advanced dynamic linear interpolation. Thus, the closest lower and upper predefined points should be used in the position of the global minimum and maximum points of the static linear interpolation. This can be achieved by using the following equation:

$$\tilde{x}_j = x_j^{\text{clb}} + \left(x_j^{\text{cub}} - x_j^{\text{clb}}\right) \left[\frac{T_{R_i}^{F_x} - T_{R_i}^{\text{clb}}}{T_{R_i}^{\text{cub}} - T_{R_i}^{\text{clb}}}\right] \tag{9.106}$$

where x_j^{clb} and x_j^{cub} are respectively the closest lower and upper bounds or points predefined near the location of the actual point x. These closest points can be determined by mapping their closest lower and upper operating times $T_{R_i}^{\text{clb}}$ and $T_{R_i}^{\text{cub}}$ measured from R_i.

9.8.1.2 Approach No.2: Logarithmic/Nonlinear Interpolation

This approach can be implemented to minimize the errors produced by the preceding approach. The magic ingredient applied here is that the relationship between x and $T_{R_i}^{F_x}$ is supposed to follow a logarithmic shape. Because that assumption is not correct, based on (4.4) and (4.5), so the static version of this approach also suffers from the

preceding accuracy problem faced with the classical linear interpolation approach. To apply it, the expression given in (9.103) is modified to be as follows:

$$x_j = x_j^{\min} + \left(x_j^{\max} - x_j^{\min}\right) \cdot \left[\frac{\log T_{R_i}^{F_x} - \log T_{R_i}^{\min}}{\log T_{R_i}^{\max} - \log T_{R_i}^{\min}}\right] \tag{9.107}$$

Similar to the preceding approach, the static equation given in (9.107) can be replaced with the following dynamic version:

$$x_j = x_j^{\text{clb}} + \left(x_j^{\text{cub}} - x_j^{\text{clb}}\right) \cdot \left[\frac{\log T_{R_i}^{F_x} - \log T_{R_i}^{\text{clb}}}{\log T_{R_i}^{\text{cub}} - \log T_{R_i}^{\text{clb}}}\right] \tag{9.108}$$

Also, to apply (9.99), the estimate of the other end relay should be modified to be:

$$\tilde{x}_j = x_j^{\text{clb}} + \left(x_j^{\text{cub}} - x_j^{\text{clb}}\right) \cdot \left[\frac{\log T_{R_i}^{F_x} - \log T_{R_i}^{\text{cub}}}{\log T_{R_i}^{\text{clb}} - \log T_{R_i}^{\text{cub}}}\right] \tag{9.109}$$

All these versions of the interpolation-based approaches are further explained through the pseudocode given in Algorithm 16.

Algorithm 16 Pseudcode of the Proposed Interpolation-Based Fault Locators

Require: Vectors of predetermined fault locations and operating times for each fault type

Require: Actual operating times supplied from both relays of a faulty line during the fault occurrence

Solve the ORC problem using any n-dimensional optimization algorithm. This stage will include load flow analysis, P/B relay pairs identification, and short-circuit analysis
Determine the fault type based on the data supplied from the two-end relays of the preceding faulty line
if the classical linear interpolation approach is used **then**
    if the static version is preferred **then**
        Apply (9.103)
    else
        Apply (9.106)
    end if
else
    if the static version is preferred **then**
        Apply (9.107)
    else
        Apply (9.108)
    end if
end if
Apply (9.97) to estimate the fault location

9.8.1.3 Approach No.3: Polynomial Regression

The good thing of the interpolation-based approaches is that they only need to know the operating time of both end relays when the fault F_x happens at the lower and upper bounds. On the opposite side, the proposed regression-based approaches need a vector of predefined operating times measured at different fault locations. This means that it is required to conduct a short-circuit analysis at different points along

each line. This process should be accomplished before energizing any line. Therefore, when this vector of short-circuit currents is substituted in (4.4) or (4.5) for calculating the corresponding operating times of the two-end relays, the response vector can be obtained. Based on (9.100)-(9.102), the predictor must be the vector of the operating times of the ith relay and not the vector of the short-circuit currents, and the response must be the vector of the preceding predefined fault locations. These input/output vectors can then be used to construct linear and nonlinear regression models. Because each branch has two relays, so two regression models are required. Based on this, any xth fault location can be predicted by just supplying the actual operating times measured from the two-end relays. Then, the average estimate $\bar{x}$ of the actual fault location x can be calculated using either (9.97) or (9.99).

The first regression-based approach is built by using two polynomial equations for both end relays. These equations could be set as linear, quadratic, cubic, or even higher-order polynomial equations. In this study, different orders are used. A general model, with order d, can be expressed as follows:

$$x_j = \theta_{0,i} + \theta_{1,i} T_{R_i} + \theta_{2,i} T_{R_i}^2 + \theta_{3,i} T_{R_i}^3 + \cdots + \theta_{d,i} T_{R_i}^d \tag{9.110}$$

These $(d+1)$ theta coefficients can be obtained by using any regression software or package, such as **SAS**, **R**, **SPSS**, MINITAB, MATLAB, etc.

9.8.1.4 Approach No.4: Asymptotic Regression

The **asymptotic regression model** can be mathematically expressed in the following general form:

$$x_j = \theta_{0,i} + \theta_{1,i} \exp\left(\theta_{2,i} T_{R_i}\right) \tag{9.111}$$

As can be clearly seen, this is a nonlinear regression model. Therefore, the optimal values of these three theta coefficients can be obtained by solving (9.111) numerically via using optimization algorithms. Such algorithms are **Gauss-Newton** (**GN**), **Gradient Descent** (**GD**), and **Levenberg-Marquardt** (**LM**).

9.8.1.5 Approach No.5: DTCC-Based Regression

This is the most advanced regression-based approach proposed in this study. The main goal behind it is to design a new nonlinear model that can act as a transposed

function of $f_i(x)$ shown in (9.102); where f_i is the operating time received from the ith relay. To clarify this point, let's recall the ANSI/IEEE TCCC model, given in (4.5) since it is a more general model than (4.4):

$$T_{R_i} = TMS_i \times \left[\frac{\beta_i}{\left(\frac{I_{R_i}}{PS_i}\right)^{\alpha_i} - 1} + \gamma_i \right] \tag{4.5}$$

The next step is to transpose the preceding time-current characteristic curve (TCCC) to our proposed **current-time characteristic curve** (**CTCC**) as follows:

$$I_{R_i} = PS_i \times \left[\frac{\beta_i \; TMS_i}{T_{R_i} - \gamma_i \; TMS_i} + 1 \right]^{1/\alpha_i} \tag{9.112}$$

Now, to apply the relation shown in (9.100) between the fault location x and its short-circuit current I_{R_i}, let's assume that a function $g(x)$ is used:

$$I_{R_i} = g_i(x) \tag{9.113}$$

For both end relays, the relation given in (9.113) is proportional to the first end relay and inversely proportional to the second end relay. For example, in Figure 4.2, if the fault F_m approaches the fault F_x, the short-circuit current I_{R_x} will increase and the short-circuit current I_{R_y} will decrease, and vice versa if F_m approaches F_y. These two behaviors can be mathematically explained as follows:

$$F_m \to F_x \quad : \quad \begin{cases} I_{R_x} \propto F_m \\ I_{R_y} \propto \frac{1}{F_m} \end{cases} \tag{9.114}$$

$$F_m \to F_y \quad : \quad \begin{cases} I_{R_x} \propto \frac{1}{F_m} \\ I_{R_y} \propto F_m \end{cases} \tag{9.115}$$

Some graphs and tables will cover these two opposite relations in the next numerical experiment with more detailed information.

Finally, after agreeing with the relation given in (9.113), a regression model can be designed to predict the fault location x by supplying the operating time received from the ith relay. This step has been successfully done by proposing a new transposed curve from CTCC to a **distance-time characteristic curve** (**DTCC**). To do that, the terms $\{PS_i, \beta_i \; TMS_i, -\gamma_i \; TMS_i, 1, 1/\alpha_i\}$ of (9.112) are replaced with

theta regression coefficients $\{\theta_{0,i}, \theta_{1,i}, \theta_{2,i}, \theta_{3,i}, \theta_{4,i}\}$ to have the following new nonlinear regression model:

$$x_j = \theta_{0,i} \times \left[\frac{\theta_{1,i}}{T_{R_i} + \theta_{2,i}} + \theta_{3,i}\right]^{\theta_{4,i}} \tag{9.116}$$

Since the whole process is done through a regression analysis, a less accurate version of (9.116) can be designed by taking $\theta_{4,i}$ equal to 1 and then removing the brackets as follows:

$$x_j = \theta_{0,i} + \frac{\theta_{1,i}}{T_{R_i} + \theta_{2,i}} \tag{9.117}$$

But, be careful, the theta coefficients of (9.117) are different than that of (9.116). Also, they are different than that of (9.110) and (9.111). That is, each regression model has its own coefficients. This is why DTCC can be simplified from (9.116) to (9.117) by selecting proper initial coefficients during fitting their nonlinear curves.

These four regression-based approaches are further explained through the pseudo-code given in Algorithm 17.

9.8.2 Numerical Experiment and Performance Comparison

In the preceding paragraphs, five different approaches have been proposed to locate faults in mesh networks by utilizing the operating times received from the two-end relays of a faulty branch. Now, it is time to evaluate the performance of each approach numerically. For this mission, the IEEE 8-bus test system shown in Figure 9.68 is used again. The full information about this test system is given in [25]. Assume that the branch between bus 1 and bus 6 is considered again. That is, L_{16} is the faulty line, and thus R_7 and R_{14} are the two-end primary relays. Thus, to apply the proposed fault location technique, shunt faults are created on several points distributed along L_{16}. Then, the short-circuit currents produced by these faults are measured by the preceding relays. In this experiment, two fault types are studied. The first one is a 3ϕ balanced fault, which is shown in Figures 9.66d and 9.66e. The second type is a single line-to-ground unbalanced fault, which is shown in Figure 9.66a. A zero impedance (0 Ω) is taken for both fault types. Also, bus 1 is chosen as a reference node, so (9.97) must be used instead of (9.98). In this experiment, $\tilde{x}_2$ is selected, so (9.97) is replaced with (9.99).

Algorithm 17 Pseudcode of the Proposed Regression-Based Fault Locators

Require: Vector of gradient operating times calculated for each type of faults bounded between $T_{R_i}^{\min}$ and $T_{R_i}^{\max}$ of the ith relay

Require: Actual operating times supplied from both relays of a faulty line during the fault occurrence

Solve the ORC problem using any n-dimensional optimization algorithm. This stage will include load flow analysis, P/B relay pairs identification, and short-circuit analysis
Determine the fault type based on the data supplied from the two-end relays of the preceding faulty line
if the polynomial regression approach is used **then**
    Regress (9.110) for both x_1 and $\tilde{x}_2$
else
    Set the optimization parameters (`max iterations`, `convergence tolerance`, `algorithm type`, `starting point`, etc)
    if the asymptotic regression approach is used **then**
        Regress (9.111) for both x_1 and $\tilde{x}_2$
    else
        Regress (9.116) or (9.117) for both x_1 and x_2
    end if
end if
Apply (9.99) to estimate the fault location

9.8.2.1 Locating 3ϕ Balanced Faults

To be able to apply this fault location technique, the ORC problem should be solved first. For this part, the ORC settings given in [13] are used where the close-in short-circuit currents seen by all P/B pairs are shown in Table 6.12. The optimal values of TMS and PS of all these 14 DOCRs are obtained through the BBO-LP algorithm described in Chapter 2, while the optimal ORC solution to the short-circuit currents given in Table 6.12 is presented in [13] and tabulated in Table 8.1. It has to be said that these TMS and PS values are optimized using the IEC standard inverse TCCC.

Substituting these optimal settings and standard coefficients in (4.4) for R_7 and R_{14} yields:

$$T_{R_7} = \frac{0.14(0.1)}{\left[\frac{\left(\frac{5}{800}\right)I_{R_7}}{2.5}\right]^{0.02} - 1} = \frac{0.014}{(0.0025I_{R_7})^{0.02} - 1} \tag{9.118}$$

$$T_{R_{14}} = \frac{0.14(0.1)}{\left[\frac{\left(\frac{5}{800}\right)I_{R_{14}}}{2.0}\right]^{0.02} - 1} = \frac{0.014}{(0.003125I_{R_{14}})^{0.02} - 1} \tag{9.119}$$

It is clear that the operating times of R_7 and R_{14} are functions of I_{R_7} and $I_{R_{14}}$, respectively. Now, by referring to (9.113), the preceding two equations can be re-expressed as functions of the fault location x:

$$T_{R_7} = \frac{0.14(0.1)}{\left[\frac{\left(\frac{5}{800}\right)g_7(x)}{2.5}\right]^{0.02} - 1} = \frac{0.014}{[0.0025g_7(x)]^{0.02} - 1} \tag{9.120}$$

$$T_{R_{14}} = \frac{0.14(0.1)}{\left[\frac{\left(\frac{5}{800}\right)g_{14}(x)}{2.0}\right]^{0.02} - 1} = \frac{0.014}{[0.003125g_{14}(x)]^{0.02} - 1} \tag{9.121}$$

Therefore, the only remaining step is to have a vector of operating times for each relay. This can be done by creating multiple faults at different points of L_{16}, so T_{R_7} and $T_{R_{14}}$ can be computed based on I_{R_7} and $I_{R_{14}}$, respectively. Thus, to have a mirror relation (i.e., an inverse function), the vector of predefined operating times measured from each relay should act as the predictor (i.e., input variable), and the vector of the predefined fault points should act as the response (i.e., output variable) in such regression models. To accomplish this task, it is essential to know the values of I_{R_7} and $I_{R_{14}}$ calculated for each fault point. Table 9.30 shows a gradient of 3ϕ faults seen by the two-end relays. If I_{R_7} and $I_{R_{14}}$ are respectively substituted in (9.118) and (9.119), then the operating time or the predictor of each relay can be determined. With this essential information, the preceding five interpolation- and regression-based approaches can now be applied to estimate the location of faults in mesh networks.

Linear Interpolation Models

To apply this approach, the values of $x_j^{\min}$, $x_j^{\max}$, $T_{R_i}^{\min}$, and $T_{R_i}^{\max}$ tabulated in Table 9.30 are substituted in (9.103) for R_{14} and in (9.105) for R_7. Thus, the location of

Table 9.30: Operating Times of the Two-End Relays Calculated Based on 3ϕ Faults Created at Different Points of L_{16} of the IEEE 8-Bus Test System (IDMT)

Fault	**Bus 1: R_{14}**		**Bus 6: R_7**	
Location	**Short-Circuit**	**Operating**	**Short-Circuit**	**Operating**
x (%)	**Current (A)**	**Time (s)**	**Current (A)**	**Time (s)**
1	1291.33	0.49479	907.51	0.84747
5	1255.12	0.50523	933.78	0.81871
10	1210.68	0.51910	967.10	0.78592
15	1167.45	0.53388	1001.01	0.75613
20	1125.37	0.54967	1035.53	0.72893
25	1084.38	0.56659	1070.66	0.70400
30	1044.45	0.58478	1106.43	0.68103
35	1005.51	0.60442	1142.86	0.65980
40	967.56	0.62568	1179.99	0.64010
45	930.55	0.64879	1217.84	0.62175
50	894.47	0.67402	1256.47	0.60460
55	859.33	0.70165	1295.90	0.58852
60	825.11	0.73205	1336.20	0.57340
65	791.84	0.76562	1377.41	0.55914
70	759.56	0.80281	1419.59	0.54566
75	728.30	0.84419	1462.80	0.53289
80	698.14	0.89034	1507.12	0.52074
85	669.15	0.94193	1552.64	0.50916
90	641.52	0.99946	1599.28	0.49814
95	615.30	1.06370	1647.32	0.48758
99	595.48	1.12014	1686.75	0.47945

F_x can be estimated by taking the average of the following two estimates using (9.99):

$$\begin{aligned} x_1 &= 1 + (99-1)\left[\frac{T_{R_{14}}^{F_x} - 0.49479}{1.12014 - 0.49479}\right] \\ &= 156.71224 T_{R_{14}}^{F_x} - 76.53965 \end{aligned} \tag{9.122}$$

$$\begin{aligned} \tilde{x}_2 &= 1 + (99-1)\left[\frac{T_{R_7}^{F_x} - 0.84747}{0.47945 - 0.84747}\right] \\ &= 226.67268 - 266.28988 T_{R_7}^{F_x} \end{aligned} \tag{9.123}$$

For example, suppose that the actual location of an unknown 3ϕ fault (that needs to be located) is at 7% away from bus 1 (i.e., 93% from bus 6). Then, the short-circuit currents seen by the two-end relays are ($I_{R_{14}} = 1237.20$ A) and ($I_{R_7} = 947.04$ A). Substituting these values in their respective equations (9.118) and (9.119) yields ($T_{R_{14}} = 0.51067$ s) and ($T_{R_7} = 0.80520$ s). If these two operating times are respectively substituted in (9.122) and (9.123), then the fault location x can be estimated as ($x_1 = 3.48859\%$) and ($\tilde{x}_2 = 12.25607\%$). Therefore, using (9.99), the average estimate is ($\bar{x} = \frac{3.48859+12.25607}{2} = 7.87233\%$). It is clear that the error produced by each individual relay can be effectively reduced by taking their average estimate $\bar{x}$.

If the dynamic version given in (9.106) is adopted here to estimate the same location of the preceding 3ϕ fault, then the closest lower and upper operating times to the actual value ($T_{R_{14}} = 0.51067$ s) are ($T_{R_{14}}^{\text{clb}} = 0.50523$ s) and ($T_{R_{14}}^{\text{cub}} = 0.51910$ s), respectively. If these values are mapped with their predefined fault locations listed in Table 9.30, then the closest lower and upper points are ($x_1^{\text{clb}} = 5\%$) and ($x_1^{\text{cub}} = 10\%$). The same thing can be applied to R_7 where the actual operating point ($T_{R_7} = 0.80520$ s) is dynamically bounded between ($T_{R_7}^{\text{clb}} = 0.78592$ s) and ($T_{R_7}^{\text{cub}} = 0.81871$ s). Thus, by referring to Table 9.30, the closest lower and upper points to $\tilde{x}_2$ are ($\tilde{x}_2^{\text{clb}} = 5\%$) and ($\tilde{x}_2^{\text{cub}} = 10\%$). Now, by substituting all these values in (9.106), the estimates x_1 and $\tilde{x}_2$ can be calculated as follows:

$$\begin{aligned} x_1 &= 5 + (10-5)\left[\frac{T_{R_{14}}^{F_x} - 0.50523}{0.51910 - 0.50523}\right] \\ &= 360.49027 T_{R_{14}}^{F_x} - 177.13050 \end{aligned} \tag{9.124}$$

$$\begin{aligned} \tilde{x}_2 &= 5 + (10-5)\left[\frac{T_{R_7}^{F_x} - 0.81871}{0.78592 - 0.81871}\right] \\ &= 129.84142 - 152.48551 T_{R_7}^{F_x} \end{aligned} \tag{9.125}$$

The input variables of (9.124) and (9.125) are already determined by R_{14} and R_7, which are respectively equal to ($T_{R_{14}}^{F_x} = 0.51067$ s) and ($T_{R_7}^{F_x} = 0.80520$ s). Thus, the preceding two estimates are equal to ($x_1 = 6.96107\%$) and ($\tilde{x}_2 = 7.06009\%$). Using (9.99), the average estimate is ($\bar{x} = \frac{6.96107+7.06009}{2} = 7.01058\%$). It is obvious that both the individual errors and the overall error are very small.

Although the average estimate calculated by the static linear interpolations is very close to the actual location, the error generated by that approach increases as the actual fault F_x moves away from the two-end relays. To reveal this phenomenon, let's now assume that the actual location of F_x is at 43% away from bus 1 (i.e., 57% away from bus 6). By conducting a 3ϕ fault analysis, the short-circuit currents seen by R_7 and R_{14} are 1202.61 and 945.24 amperes, respectively. Using (9.118) and (9.119), the operating times measured from the two-end relays are ($T_{R_7} = 0.62894$ s) and ($T_{R_{14}} = 0.63931$ s), respectively. Thus, using (9.122) and (9.123), the two estimates are equal to ($x_1 = 23.64805\%$) and ($\tilde{x}_2 = 59.19232\%$), respectively. Finally, the average estimate is ($\bar{x} = \frac{23.64805+59.19232}{2} = 41.42019\%$). As can be seen, each estimate has a big error compared to 43%. But again, the performance can be effectively enhanced by taking the average $\bar{x}$.

The reason behind the preceding phenomenon is that as the fault located near one of the line terminals, the gap between the actual and estimated values is very small because the whole interpolation process is done between the points $x_j^{\min}$ and $x_j^{\max}$. On the opposite side, the dynamic linear interpolation provides highly precise estimations with an ignorable error. For the preceding 3ϕ fault located at 43% away from bus 1, the estimates can be dynamically calculated as follows:

$$\begin{aligned} x_1 &= 40 + (45 - 40)\left[\frac{T_{R_{14}}^{F_x} - 0.62568}{0.64879 - 0.62568}\right] \\ &= 216.35656 T_{R_{14}}^{F_x} - 95.36997 \qquad (9.126) \\ \tilde{x}_2 &= 40 + (45 - 40)\left[\frac{T_{R_7}^{F_x} - 0.64010}{0.62175 - 0.64010}\right] \\ &= 214.41417 - 272.47956 T_{R_7}^{F_x} \qquad (9.127) \end{aligned}$$

Solving these two equations for ($T_{R_{14}} = 0.63931$ s) and ($T_{R_7} = 0.62894$ s) yields ($x_1 = 42.94894\%$) and ($\tilde{x}_2 = 43.04088\%$). Applying (9.99) for the average estimate gives ($\bar{x} = \frac{42.94894+43.04088}{2} = 42.99491\%$). As can be obviously observed, both the

individual estimates and the overall estimate have very small errors.

Logarithmic/Nonlinear Interpolation Models

The same steps can be done here. But, the numeric lower and upper bounds of x_j and T_{R_i} must be substituted in (9.107) instead of (9.103). Also, by considering the same steps done on (9.105) for directly getting $\tilde{x}_j$ from (9.107), the individual estimates x_1 and $\tilde{x}_2$ can be computed as follows:

$$\begin{aligned} x_1 &= 1 + (99-1)\left[\frac{\log T_{R_{14}}^{F_x} - \log 0.49479}{\log 1.12014 - \log 0.49479}\right] \\ &= 119.93995 \log\left(T_{R_{14}}^{F_x}\right) + 85.39237 \qquad (9.128) \\ \tilde{x}_2 &= 1 + (99-1)\left[\frac{\log T_{R_7}^{F_x} - \log 0.84747}{\log 0.47945 - \log 0.84747}\right] \\ &= -172.04578 \log\left(T_{R_7}^{F_x}\right) - 27.47355 \qquad (9.129) \end{aligned}$$

For the preceding actual 3ϕ fault occurred at 7% away from bus 1, the values of these two estimates are ($x_1 = 4.78892\%$) and ($\tilde{x}_2 = 9.80268\%$). Using (9.99), the overall error is minimized where ($\bar{x} = \frac{x_1+\tilde{x}_2}{2} = 7.29580\%$). Comparing with the static linear interpolation approach, there are significant improvements in the individual and overall errors, because the estimates of the former approach were ($x_1 = 3.48859\%$), ($\tilde{x}_2 = 12.25607\%$), and ($\bar{x} = 7.87233\%$).

For the other actual fault location, which is at 43% away from bus 1, the estimates x_1 and $\tilde{x}_2$ using (9.128) and (9.129) are ($x_1 = 31.73534\%$) and ($\tilde{x}_2 = 52.30742\%$). Also, the average value is ($\bar{x} = \frac{x_1+\tilde{x}_2}{2} = 42.02138\%$). Again, by returning back to the former approach, the estimates were ($x_1 = 23.64805\%$), ($\tilde{x}_2 = 59.19232\%$), and ($\bar{x} = 41.42019\%$). This evidence gives a solid conclusion that the static version of this approach is better than that of the classical linear interpolation approach.

Now, let's switch it from the static mode to the dynamic mode by replacing (9.107) with (9.108) for x_1, and (9.109) for $\tilde{x}_2$. For the actual 3ϕ fault occurred at 7%, the same equations given in (9.124) and (9.125) are used here, but with embedding `log`

beside each operating time:

$$\begin{aligned} x_1 &= 5 + (10-5)\left[\frac{\log T_{R_{14}}^{F_x} - \log 0.50523}{\log 0.51910 - \log 0.50523}\right] \\ &= 184.61921 \log\left(T_{R_{14}}^{F_x}\right) + 131.0472 \qquad (9.130) \\ \tilde{x}_2 &= 5 + (10-5)\left[\frac{\log T_{R_7}^{F_x} - \log 0.81871}{\log 0.78592 - \log 0.81871}\right] \\ &= -122.32438 \log\left(T_{R_7}^{F_x}\right) - 19.46798 \qquad (9.131) \end{aligned}$$

Solving the above two equations with the same operating times determined by R_{14} and R_7, the two estimates are $(x_1 = 6.97724\%)$ and $(\tilde{x}_2 = 7.03538\%)$. The average of these two estimates is $(\bar{x} = \frac{x_1+\tilde{x}_2}{2} = 7.00631\ \%)$.

If the actual location is at 43%, then (9.126) and (9.127) are modified with the `log` operator to be as follows:

$$\begin{aligned} x_1 &= 40 + (45-40)\left[\frac{\log T_{R_{14}}^{F_x} - \log 0.62568}{\log 0.64879 - \log 0.62568}\right] \\ &= 137.85486 \log\left(T_{R_{14}}^{F_x}\right) + 104.64238 \qquad (9.132) \\ \tilde{x}_2 &= 40 + (45-40)\left[\frac{\log T_{R_7}^{F_x} - \log 0.64010}{\log 0.62175 - \log 0.64010}\right] \\ &= -171.90205 \log\left(T_{R_7}^{F_x}\right) - 36.69081 \qquad (9.133) \end{aligned}$$

The solutions to these two equations are $(x_1 = 42.97083\%)$ and $(\tilde{x}_2 = 43.02351\%)$, and thus substituting them into (9.99) gives $(x = \frac{x_1+\tilde{x}_2}{2} = 42.99717\%)$. It is obvious that the dynamic mode of the former approach provides high-quality solutions. However, the dynamic mode of this approach can provide better solutions. Therefore, this additional analysis gives another solid conclusion that the logarithmic-based nonlinear interpolation is better than the classical linear interpolation for both modes.

To give a quick recap of all the analysis done above, Figure 9.71 shows the linearity of these four interpolation methods used in estimating actual fault locations. The worst method is the static linear interpolation, and the static nonlinear interpolation scores the second-worst method. The errors of these two methods are represented by the gap between their curves and the actual fault points. On the opposite side, the dynamic linear and nonlinear interpolation versions prove themselves as excellent DOCR-based fault locators. The plots shown in Figure 9.71 are created based on

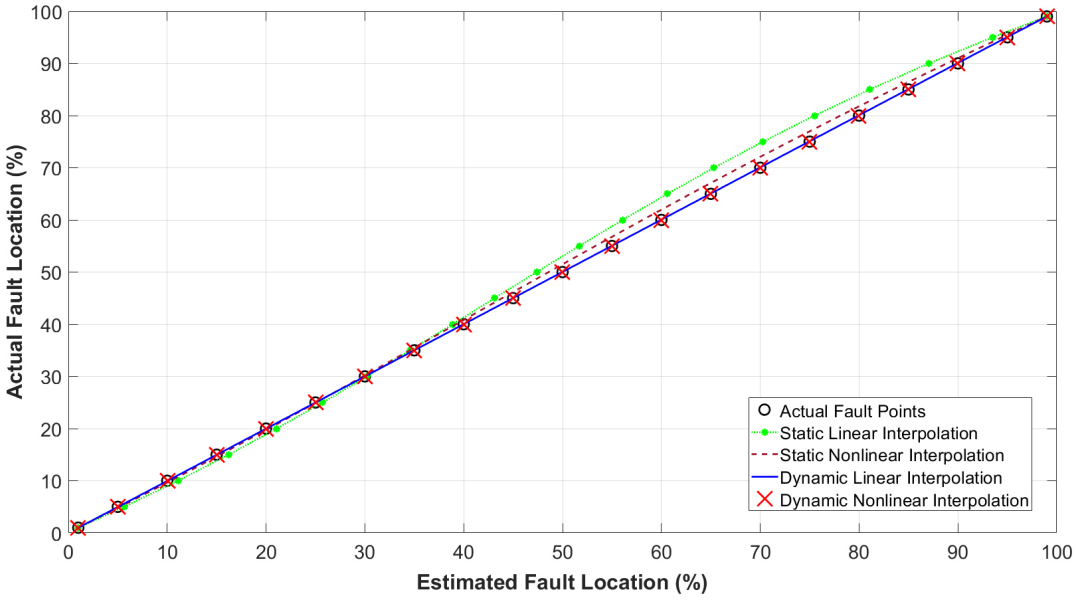


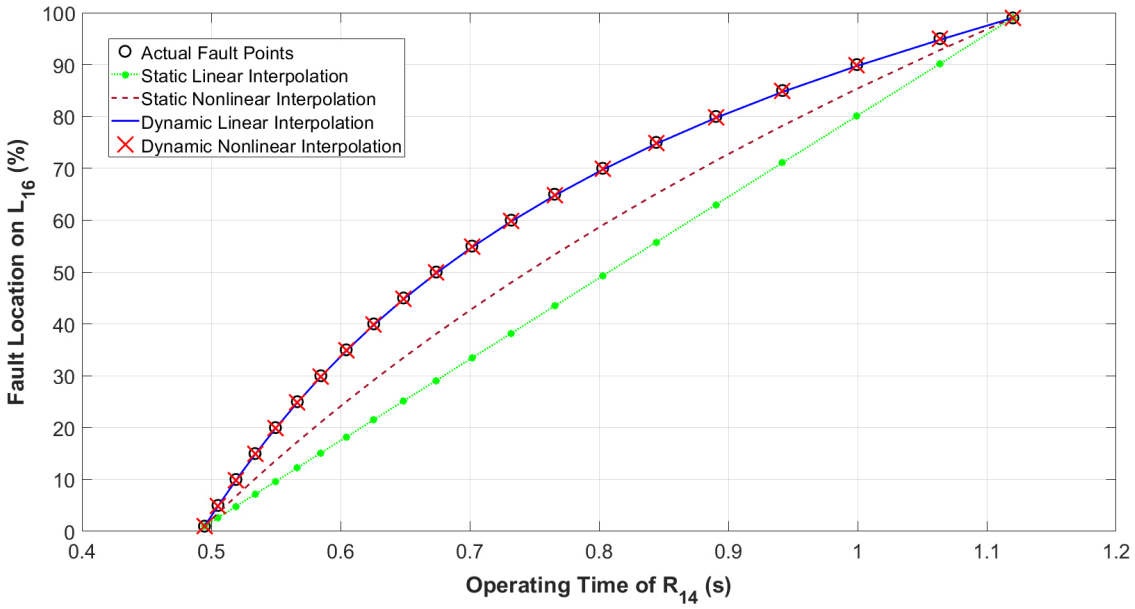
Figure 9.71: Inspecting the Linearity of Both Modes of the Linear and Nonlinear Interpolation Approaches

the average estimate $\bar{x}$ calculated by (9.99). Thus, to give more details about the behavior of each individual estimate, i.e. x_1 and $\tilde{x}_2$, Figure 9.72 shows the fault locations estimated on the faulty line L_{16} by the two-end relays R_{14} and R_7. The first- and second-worst estimators are the static versions of the linear and nonlinear interpolation approaches, respectively, where the estimations of the dynamic versions pass all the actual fault points created on L_{16}.

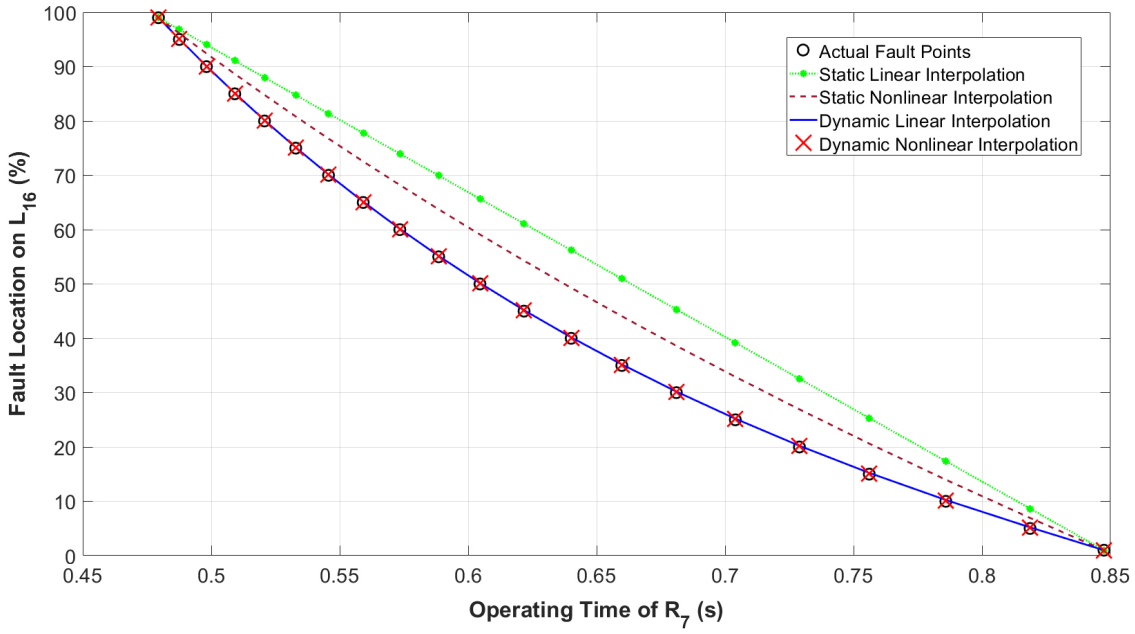
The main drawback associated with the dynamic interpolation versions is that their programming codes need to be recalled every time there is a new actual operating time. The reason behind this is to re-adjust their closest lower and upper bounds used in (9.106), (9.108), and (9.109). This has been seen when jumping from 7% to 43% of the preceding 3ϕ faults. If there is no technical issue with that adaptive process, then the dynamic interpolation methods can be used without any problem. Otherwise, the following regression approaches could be used as alternative rigorous estimators.

Polynomial-Based Regression Models

For the regression part, the statistics and machine learning toolbox of MATLAB



(a) Relay no.14



(b) Relay no.7

Figure 9.72: Estimating Fault Locations Based on the Operating Times Measured from the Two-End Relays of L_{16}

R2017b is used. The analysis is carried out using the **two-sided confidence interval** type with a **confidence level** of 95% for all intervals. To predict the distance between bus 1 and F_x from the average of the operating times of R_7 and R_{14}, the observations given in Table 9.30 are fitted. The following d order polynomial regression models can be used for any 3ϕ fault occurs on L_{16}:

- **1st Order (Linear) Equations:**

$$x_1 = -63.106 + 156.59T_{R_{14}} \tag{9.134}$$
$$\tilde{x}_2 = 619.478 - 268.53T_{R_7} \tag{9.135}$$

- **2nd Order (Quadratic) Equations:**

$$x_1 = -190.62 + 503.72T_{R_{14}} - 221.42T_{R_{14}}^2 \tag{9.136}$$
$$\tilde{x}_2 = 416.69 - 896.19T_{R_7} + 480.84T_{R_7}^2 \tag{9.137}$$

- **3rd Order (Cubic) Equations:**

$$x_1 = -344.44 + 1129.5T_{R_{14}} - 1038.4T_{R_{14}}^2 + 342.71T_{R_{14}}^3 \tag{9.138}$$
$$\tilde{x}_2 = 619.5 - 1855.4T_{R_7} + 1965.7T_{R_7}^2 - 752.52T_{R_7}^3 \tag{9.139}$$

- **4th Order (Quartic) Equations:**

$$x_1 = -508.39 + 2017.5T_{R_{14}} - 2794.5T_{R_{14}}^2 + 1846.3T_{R_{14}}^3 - 470.76T_{R_{14}}^4 \tag{9.140}$$
$$\tilde{x}_2 = 776.73 - 2846.1T_{R_7} + 4276.8T_{R_7}^2 - 3118.3T_{R_7}^3 + 896.85T_{R_7}^4 \tag{9.141}$$

- **5th Order (Quintic) Equations:**

$$x_1 = -668.07 + 3097.6T_{R_{14}} - 5657.6T_{R_{14}}^2 + 5564.9T_{R_{14}}^3 - 2838.4T_{R_{14}}^4 + 591.65T_{R_{14}}^5 \tag{9.142}$$
$$\tilde{x}_2 = 827.36 - 3244.7T_{R_7} + 5519.6T_{R_7}^2 - 5037T_{R_7}^3 + 2363.7T_{R_7}^4 - 444.29T_{R_7}^5 \tag{9.143}$$

- **6th Order (Sextic) Equations:**

$$\begin{aligned} x_1 = &-812.78 + 4271.9T_{R_{14}} - 9563T_{R_{14}}^2 + 12379T_{R_{14}}^3 \\ &- 9418.2T_{R_{14}}^4 + 3927.4T_{R_{14}}^5 - 693.98T_{R_{14}}^6 \end{aligned} \tag{9.144}$$

$$\begin{aligned} \tilde{x}_2 = &\, 779.16 - 2789.5T_{R_7} + 3742.4T_{R_7}^2 - 1365.8T_{R_7}^3 \\ &- 1868.6T_{R_7}^4 + 2137.7T_{R_7}^5 - 651.3T_{R_7}^6 \end{aligned} \tag{9.145}$$

- **7th Order (Septic) Equations:**

$$\begin{aligned} x_1 = &-957.69 + 5644.9T_{R_{14}} - 15062T_{R_{14}}^2 + 24447T_{R_{14}}^3 \\ &- 25094T_{R_{14}}^4 + 15983T_{R_{14}}^5 - 5777.6T_{R_{14}}^6 + 907.17T_{R_{14}}^7 \end{aligned} \tag{9.146}$$

$$\begin{aligned} \tilde{x}_2 = &\, 438.76 + 962.39T_{R_7} - 13863T_{R_7}^2 + 44223T_{R_7}^3 \\ &- 72234T_{R_7}^4 + 66875T_{R_7}^5 - 33526T_{R_7}^6 + 7109T_{R_7}^7 \end{aligned} \tag{9.147}$$

- **8th Order (Octic) Equations:**

$$\begin{aligned} x_1 = &-1294.2 + 9297.1T_{R_{14}} - 32199T_{R_{14}}^2 + 69859T_{R_{14}}^3 \\ &- 99437T_{R_{14}}^4 + 92982T_{R_{14}}^5 - 55061T_{R_{14}}^6 \\ &+ 18734T_{R_{14}}^7 - 2790.7T_{R_{14}}^8 \end{aligned} \tag{9.148}$$

$$\begin{aligned} \tilde{x}_2 = &\, 372.63 + 1796.3T_{R_7} - 18436T_{R_7}^2 + 58478T_{R_7}^3 \\ &- 99843T_{R_7}^4 + 100905T_{R_7}^5 - 59594T_{R_7}^6 \\ &+ 18457T_{R_7}^7 - 2149.5T_{R_7}^8 \end{aligned} \tag{9.149}$$

- **9th Order (Nonic) Equations:**

$$\begin{aligned} x_1 = &-4624.4 + 50088T_{R_{14}} - 252026T_{R_{14}}^2 + 753944T_{R_{14}}^3 \\ &- 1454221T_{R_{14}}^4 + 1863867T_{R_{14}}^5 - 1583053T_{R_{14}}^6 \\ &+ 858063T_{R_{14}}^7 - 269172T_{R_{14}}^8 + 37224T_{R_{14}}^9 \end{aligned} \tag{9.150}$$

$$\begin{aligned} \tilde{x}_2 = &-3828.2 + 61486T_{R_7} - 393526T_{R_7}^2 + 1426649T_{R_7}^3 \\ &- 3292235T_{R_7}^4 + 5042478T_{R_7}^5 - 5134163T_{R_7}^6 \\ &+ 3352257T_{R_7}^7 - 1273624T_{R_7}^8 + 214500T_{R_7}^9 \end{aligned} \tag{9.151}$$

Table 9.31 shows the performance of these polynomial regression models, which are graphically represented in Figure 9.73. It can be clearly observed that increasing

the polynomial order d does not necessarily improve the performance. The ordinary and adjusted coefficients of determination (R^2 and R^2_{adj}) reach the saturation level when d equals 3 and above. Thus, there is no reason to increase d more than 3 if the main objective is to enhance these two coefficients. Also, the error degrees of freedom (DF) decreases as d increases. Furthermore, $d > 3$ means dealing with complicated models that consume more computation times. However, it may be good to set d above 3 for increasing the F-statistic and decreasing its corresponding p-value. But, the main questions are: *How much the order should be? Is it identical for both relays? Is it same for all network branches?* By referring to Table 9.31, the best order that gives the highest F-statistic and the lowest p-value for R_7 is $d = 5$. For R_{14}, the best F-statistic and p-value are respectively scored at $d = 9$ and $d = 6$. The benefit from these F-statistics and their p-values is to statistically answer whether the regression model is better than the intercept-only model[41] or not by conducting the following hypothesis test [214]:

$$H_0\text{: } x_j = \theta_{0,i} \text{ (constant model), versus}$$
$$H_1\text{: } x_j = \theta_{0,i} + \theta_{1,i}T_{R_i} + \theta_{2,i}T^2_{R_i} + \theta_{3,i}T^3_{R_i} + \cdots + \theta_{d,i}T^d_{R_i}$$

where the null hypothesis H_0 is rejected based on the F-statistic or its p-value. If the p-value is used, then H_0 is rejected if the p-value is less than $\alpha = 5\%$.

Based on the values listed in Table 9.31, all the F-statistics and their p-values are highly significant. This means that all the regression models pass the preceding statistical hypothesis test, and thus it is hard to depend on this test in selecting the best order. The last performance criterion covered in this regression analysis is about the root mean squared error (RMSE). The best RMSE scored for R_7 and R_{14} are $d = 7$ and $d = 9$, respectively. However, we should ask: *Is it worth to increase d even for a small improvement on RMSE?* The answer to this vital question can be provided by conducting a further statistical hypothesis test to check the contribution of each new degree added to the regression model. This can be carried out on (9.110) using the t-statistic as follows:

$$H_0\text{: } \theta_{d,i} = 0 \text{ (regression model with } d-1 \text{ order), versus}$$
$$H_1\text{: } \theta_{d,i} \neq 0 \text{ (regression model with } d \text{ order)}$$

[41] It is also called the constant model.

Table 9.31: Performance of the Polynomial Models Given in (9.134)-(9.151)

Polynomial Model		**Regression Performance Criteria**							
Name	**Order**	$\mathbf{R_i}$	**DF**	$\mathcal{N}^a$	**RMSE**	$\mathbf{R^2}$	$\mathbf{R^2_{adj}}$	**F-statistic**	**p-value**
linear	1	R_7	19	32.318	5.68	0.968	0.966	571	1.24E-15
		R_{14}	19	58.8004	7.67	0.941	0.938	305	3.69E-13
quadratic	2	R_7	18	0.766	0.875	0.999	0.999	1.24E+04	5.45E-29
		R_{14}	18	3.893	1.97	0.996	0.996	2.44E+03	1.23E-22
cubic	3	R_7	17	0.0101	0.101	1	1	6.26E+05	4.64E-43
		R_{14}	17	0.1973	0.444	1	1	3.22E+04	4.20E-32
quartic	4	R_7	16	2.73E-05	0.00522	1	1	1.75E+08	6.79E-61
		R_{14}	16	0.0082	0.0907	1	1	5.78E+05	4.70E-41
quintic	5	R_7	15	4.60E-06	0.00214	1	1	8.28E+08	9.56E-63
		R_{14}	15	3.05E-04	0.0175	1	1	1.25E+07	4.39E-49
sextic	6	R_7	14	4.45E-06	0.00211	1	1	7.13E+08	1.44E-58
		R_{14}	14	2.69E-05	0.00519	1	1	1.18E+08	4.29E-53
septic	7	R_7	13	4.30E-06	0.00207	1	1	6.33E+08	2.01E-54
		R_{14}	13	1.67E-05	0.00409	1	1	1.63E+08	1.38E-50
octic	8	R_7	12	4.66E-06	0.00216	1	1	5.11E+08	5.34E-50
		R_{14}	12	1.56E-05	0.00395	1	1	1.53E+08	7.58E-47
nonic	9	R_7	11	5.05E-06	0.00225	1	1	4.19E+08	1.24E-45
		R_{14}	11	8.33E-06	0.00289	1	1	2.54E+08	1.93E-44

[a]It is the sum of the square of the residuals obtained for each d divided by the error degrees of freedom. This term will be explained later.

The null hypothesis is rejected based on the t-statistic or its p-value of the dth term. To do that for each end relay, the following tests are conducted:

- Selecting the best order for x_1:
  - Comparing (9.136) with (9.134):
    The p-value of $\theta_{2,14} = 2.8686 \times 10^{-12} < \alpha$, so $d = 2$ cannot be rejected.
  - Comparing (9.138) with (9.136):
    The p-value of $\theta_{3,14} = 1.1731 \times 10^{-12} < \alpha$, so $d = 3$ cannot be rejected.
  - Comparing (9.140) with (9.138):
    The p-value of $\theta_{4,14} = 1.1337 \times 10^{-12} < \alpha$, so $d = 4$ cannot be rejected.

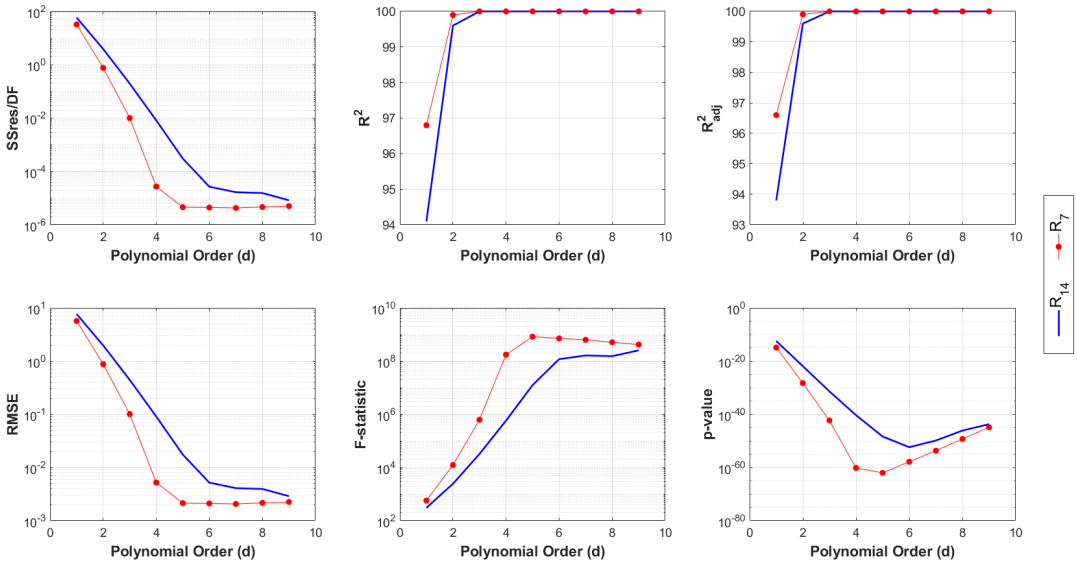


Figure 9.73: Performance of the Polynomial Models Given in (9.134)-(9.151)

- Comparing (9.142) with (9.140):
  The p-value of $\theta_{5,14} = 2.3384 \times 10^{-12} < \alpha$, so $d = 5$ cannot be rejected.
- Comparing (9.144) with (9.142):
  The p-value of $\theta_{6,14} = 5.6246 \times 10^{-9} < \alpha$, so $d = 6$ cannot be rejected.
- Comparing (9.146) with (9.144):
  The p-value of $\theta_{7,14} = 0.0086423 < \alpha$, so $d = 7$ cannot be rejected.
- Comparing (9.148) with (9.146):
  The p-values of all theta coefficients, i.e. $\theta_{3,14}$ to $\theta_{8,14}$, of (9.148) are bigger than $\alpha = 0.05$; especially the last one where $\theta_{8,14} = 0.18905$. Thus, this test indicates a possible reduction in the number of terms used in (9.148). Because (9.146) cannot be rejected, so it means that $d = 8$ is not significant.

- Selecting the best order for $\tilde{x}_2$:
  - Comparing (9.137) with (9.135):
    The p-value of $\theta_{2,7} = 2.7191 \times 10^{-16} < \alpha$, so $d = 2$ cannot be rejected.
  - Comparing (9.139) with (9.137):
    The p-value of $\theta_{3,7} = 1.2811 \times 10^{-17} < \alpha$, so $d = 3$ cannot be rejected.

- Comparing (9.141) with (9.139):
  The p-value of $\theta_{4,7} = 3.2972 \times 10^{-22} < \alpha$, so $d = 4$ cannot be rejected.

- Comparing (9.143) with (9.141):
  The p-value of $\theta_{5,7} = 2.1538 \times 10^{-7} < \alpha$, so $d = 5$ cannot be rejected.

- Comparing (9.145) with (9.143):
  The p-values of $\{\theta_{2,7}, \theta_{3,7}, \theta_{4,7}, \theta_{5,7}, \theta_{6,7}\}$ of (9.145) are bigger than $\alpha = 0.05$. The worst p-value is 0.65624, which is scored with $\theta_{3,7}$. The last term has a p-value of $\theta_{6,7} = 0.2406$. In regression, if the p-value of any term is less than α, then it is an indication that there is at least one predictor needs to be removed from the regression model. In regression analysis, it is preferable to keep the polynomial model in its standard form; i.e. keeping the intercept and all the lower-order terms[42]. This means that the 6th order of (9.145) is insignificant, and thus (9.143) is statistically selected as the best polynomial regression model for R_7.

The above process is known as the forward selection method because the polynomial order d is gradually increased until getting a bad t-statistic. Alternatively, the process can be started from the other side, i.e. setting the order at a high value to ensure that its t-statistic is not significant. Then, start decreasing the order by deleting the highest term until getting significant t-statistic. This process is known as the backward selection method where the solution could be different than that of the first method [214].

There is also another popular method used in the literature to find the optimum polynomial order. This method can be carried out by either finding the global minimum or a satisfying value of the following equation:

$$\mathcal{N} = \frac{\mathrm{SS}_{\mathrm{res}}(d)}{\mathrm{DF}} \tag{9.152}$$

where $\mathrm{SS}_{\mathrm{res}}(d)$ is the residual sum of squares obtained from the dth order polynomial equation.

[42] Such models are known as **hierarchical polynomial regression models**.

The $\mathrm{SS}_{\mathrm{res}}(d)$ and DF terms can be mathematically calculated for the jth polynomial equation as follows:

$$\mathrm{SS}_{\mathrm{res}_j}(d) = \sum_{i=1}^{\mathcal{O}} \left(x_{j,i} - \bar{x}_{j,i}\right)^2 \tag{9.153}$$

$$\mathrm{DF}_j = \mathcal{O} - d_j - 1 \tag{9.154}$$

where $\mathcal{O}$ and d_j are respectively the number of data and the jth polynomial order used in the regression analysis.

This test can be conducted by inspecting the column $\mathcal{N}$ listed in Table 9.31 and shown in the first subplot of Figure 9.73. For x_1, it can be found that there is a slight improvement on $\mathcal{N}$ after jumping from $d = 7$ to $d = 8$. Thus, $d = 7$ can be selected as the best order although $d = 8$ and $d = 9$ give better solutions. For $\tilde{x}_2$, the global value of $\mathcal{N}$ occurs at $d = 7$ as also shown in the top-left subplot of Figure 9.73, and thus it could be selected as the best order. However, inspecting $\mathcal{N}$ when d jumps from 5 to 6 shows that there is a slight improvement on $\mathcal{N}$. Thus, $d = 5$ is preferred. The main objective in selecting the optimum polynomial order is to keep it as low as possible[43].

From this analysis, the final estimate $\bar{x}$ is calculated by taking the average of (9.146) and (9.143) using (9.99). Thus, comparing with dynamic linear/nonlinear interpolation approaches, the polynomial regression approach can be used for any fault location without adjusting any parameter once the model is successfully designed. However, it can be obviously seen from all these statistical tests that the main drawback of the polynomial regression approach are the weary steps used in selecting d. This hidden challenge forces us to think about designing compact fault locators using nonlinear regression models. Some of these models are presented below:

Asymptotic Regression Models

The mathematical expression of the general asymptotic nonlinear regression model is given in (9.111). There are three theta coefficients need to be determined by using optimization algorithms. In this study, the default optimization algorithm used in MATLAB R2017b (i.e., the trust-region algorithm) is chosen in all the nonlinear

[43] This process is known as the Occam's or Ockham's razor.

regression models presented in this section. A confidence level of $\alpha = 95\%$, for all the intervals, is used with a convergence tolerance of $\varepsilon = 10^{-8}$ and maximum iterations of 10^4.

The fitted nonlinear curves of the two-end relays of L_{16} are found as follows:

$$x_1 = 113.51811 - 518.00930 \exp\left(-3.10603 T_{R_{14}}\right) \tag{9.155}$$

$$\tilde{x}_2 = -31.57224 + 780.74759 \exp\left(-3.73428 T_{R_7}\right) \tag{9.156}$$

The full details about the results of these two nonlinear regression models are shown in Figure 9.74. From these results, it can be said that the limitations associated with the preceding interpolation and polynomial regression approaches can be solved with the compact models given in (9.155) and (9.156). However, the root mean squared errors shown in Figure 9.74 are relatively high compared with other models. Thus, it will be good if there are other nonlinear regression models can solve these limitations with better RMSE. This motivated us to design the DTCC-based nonlinear regression models presented below:

DTCC-Based Regression Models

It has been shown that there are two possible forms can be used to construct this nonlinear regression approach. The first one is given in (9.116), and it is a direct inversion of (9.102). The main drawbacks of this model are: 1. it contains many coefficients that need to be tuned and optimized through a nonlinear regression analysis, and 2. it is hard to find a suitable starting point that can achieve a good convergence. To compromise between the simplicity of (9.111) and the accuracy of (9.116), a simplified DTCC-based nonlinear regression model is given in (9.117). Thus, by canceling the gain and exponent coefficients of (9.116), the whole regression process can be significantly improved. This includes the computation time, the optimization search space, the total number of iterations, the complexity, etc. By applying the same initialization parameters used in the asymptotic regression approach, the results of the two versions of the DTCC-based regression approach for the two-end relays of L_{16} are as follows:

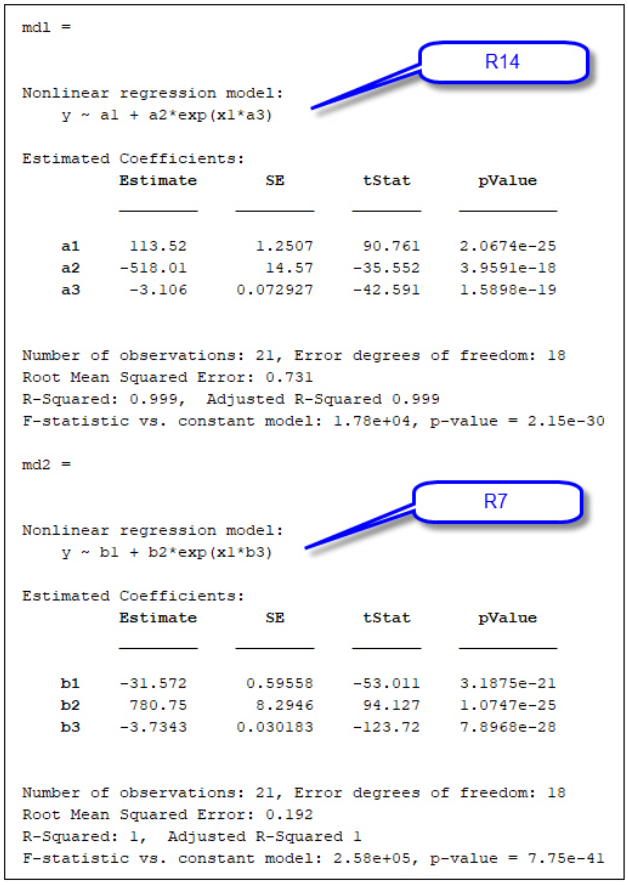


Figure 9.74: MATLAB Results Obtained by Analyzing the Asymptotic Nonlinear Regression Models Given in (9.155) and (9.156)

- **Version No.1:**

$$x_1 = 852.09291\left[\frac{2.36407}{T_{R_{14}} - 0.49270} + 5.36871\right]^{-0.97427} \tag{9.157}$$

$$\tilde{x}_2 = 255.05990\left[\frac{-1.72770}{T_{R_7} - 0.85687} - 2.11072\right]^{-1.04693} \tag{9.158}$$

- **Version No.2:**

$$x_1 = 163.88940 - \frac{68.11548}{T_{R_{14}} - 0.07636} \tag{9.159}$$

$$\tilde{x}_2 = -103.91898 + \frac{79.46815}{T_{R_7} - 0.08832} \tag{9.160}$$

Figures 9.75 and 9.76 show the performance obtained from analyzing the full and reduced DTCC nonlinear regression models for R_7 and R_{14}. As mentioned earlier, (9.159) compromises between (9.155) and (9.157), and (9.160) compromises between

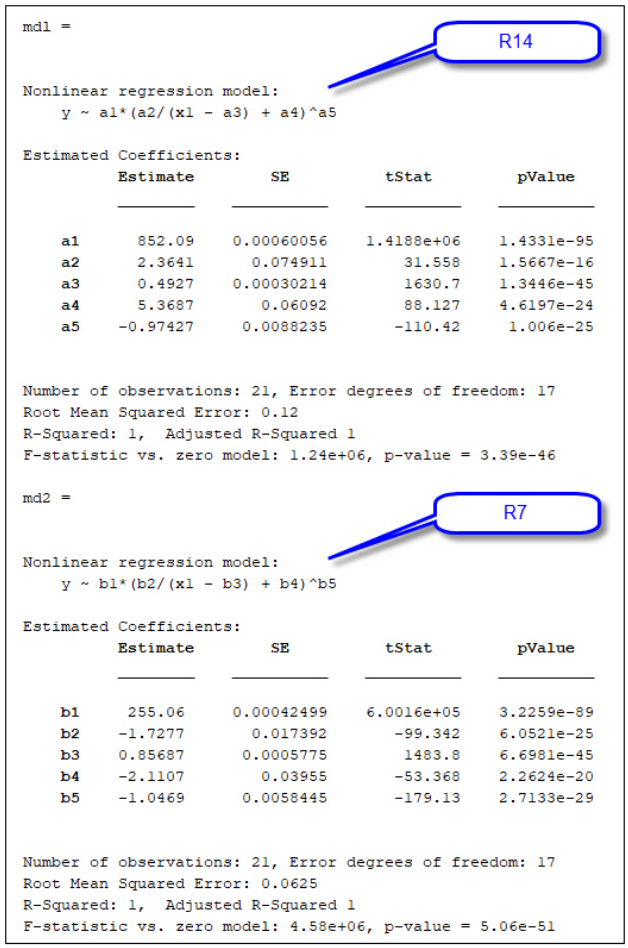


Figure 9.75: MATLAB Results Obtained by Analyzing the Full DTCC Nonlinear Regression Models Given in (9.157) and (9.158)

(9.156) and (9.158). Thus, to have a simpler and efficient model, then (9.117) is the correct choice. Otherwise, (9.116) can be selected if seeking for more accurate and precise model. Again, the dynamic interpolation approaches are good if there are enough well-distributed observations and if there is no any coding issue to update the lower and upper bounds of the program automatically. The polynomial regression models can also be a good choice if the optimal order d is selected for all ϱ DOCRs installed on the network. Table 9.32 shows the fault locations estimated by the best models. It is clear that the average estimate $\bar{x}$ can significantly enhance the overall error of most approaches. For example, x_1 and $\tilde{x}_2$ of the dynamic linear interpolation approach have RMSE of 0.213 and 0.161, respectively, while taking the average estimate $\bar{x}$ using (9.99) can reduce RMSE down to 0.047. This observation

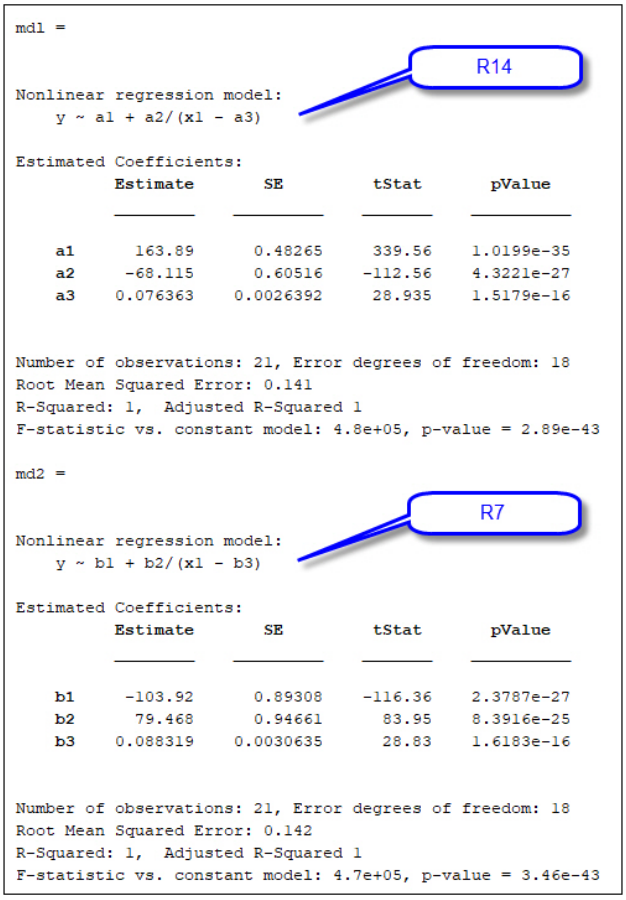


Figure 9.76: MATLAB Results Obtained by Analyzing the Reduced DTCC Nonlinear Regression Models Given in (9.159) and (9.160)

can be graphically described in Figure 9.77 by plotting the absolute errors of each response. Although the average estimate is the winner in most cases, sometimes one of the individual estimates could win too. For example, $\tilde{x}_2$ of the asymptotic nonlinear regression approach shows better estimations in both Table 9.32 and Figure 9.77.

9.8.2.2 Locating Single Line-to-Ground Unbalanced Faults

The same steps applied in the preceding symmetrical fault can also be applied here to locate single line-to-ground unbalanced faults. The ORC stage is not touched here, because the optimal settings of TMS and PS are valid for all the fault types. Thus, for the IDMT standard, Table 8.1 can be used again to set all the 14 DOCRs shown in Figure 9.68. The last step is to measure the single line-to-ground faults at predefined

Table 9.32: Estimated Locations of the 3ϕ Faults Created on L_{16} Using the Best Interpolation and Regression Approaches

Actual	**Interpolation-Based Approaches**						**Regression-Based Approaches**											
Fault	**Dynamic Linear**			**Dynamic Nonlinear**			**Polynomial**			**Asymptotic**			**DTCC - ver.1**			**DTCC - ver.2**		
Location	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$
Model	(9.106)	(9.106)	(9.99)	(9.108)	(9.109)	(9.99)	(9.146)	(9.143)	(9.99)	(9.155)	(9.156)	(9.99)	(9.157)	(9.158)	(9.99)	(9.159)	(9.160)	(9.99)
1	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	2.11	1.40	1.76	0.90	1.10	1.00	1.10	0.76	0.93
5	4.87	5.21	5.04	4.92	5.12	5.02	4.99	5.00	5.00	5.67	5.13	5.40	5.03	4.95	4.99	5.06	4.88	4.97
10	9.84	10.24	10.04	9.91	10.14	10.03	10.00	10.00	10.00	10.22	9.92	10.07	10.09	9.91	10.00	10.04	10.00	10.02
15	14.83	15.23	15.03	14.91	15.13	15.02	15.00	15.00	15.00	14.85	14.80	14.82	15.10	14.93	15.02	15.01	15.08	15.04
20	19.83	20.22	20.02	19.90	20.13	20.02	20.00	20.00	20.00	19.57	19.76	19.66	20.07	19.97	20.02	19.98	20.13	20.05
25	24.82	25.20	25.01	24.90	25.12	25.01	25.00	25.00	25.00	24.38	24.76	24.57	25.03	25.01	25.02	24.94	25.15	25.05
30	29.81	30.20	30.00	29.89	30.12	30.00	30.00	30.00	30.00	29.28	29.81	29.54	29.98	30.04	30.01	29.91	30.16	30.04
35	34.80	35.19	34.99	34.89	35.11	35.00	35.00	35.00	35.00	34.26	34.87	34.57	34.94	35.06	35.00	34.90	35.14	35.02
40	39.79	40.18	39.98	39.88	40.10	39.99	40.00	40.00	40.00	39.33	39.95	39.64	39.91	40.06	39.99	39.89	40.10	40.00
45	44.78	45.17	44.97	44.87	45.10	44.99	45.00	45.00	45.00	44.47	45.02	44.74	44.89	45.06	44.98	44.90	45.06	44.98
50	49.77	50.16	49.97	49.87	50.09	49.98	50.00	50.00	50.00	49.67	50.08	49.88	49.89	50.04	49.97	49.92	50.01	49.96
55	54.76	55.15	54.96	54.86	55.09	54.98	55.00	55.00	55.00	54.92	55.14	55.03	54.91	55.02	54.97	54.95	54.95	54.95
60	59.75	60.15	59.95	59.86	60.08	59.97	60.00	60.00	60.00	60.20	60.17	60.19	59.95	60.00	59.97	60.01	59.91	59.96
65	64.74	65.14	64.94	64.86	65.08	64.97	65.00	65.00	65.00	65.48	65.19	65.34	65.00	64.97	64.99	65.06	64.87	64.97
70	69.73	70.14	69.93	69.86	70.08	69.97	70.00	70.00	70.00	70.72	70.19	70.45	70.06	69.95	70.00	70.12	69.84	69.98
75	74.73	75.12	74.93	74.86	75.07	74.96	75.00	75.00	75.00	75.88	75.16	75.52	75.12	74.93	75.02	75.18	74.83	75.01
80	79.72	80.12	79.92	79.86	80.06	79.96	80.00	80.00	80.00	80.91	80.11	80.51	80.16	79.93	80.05	80.21	79.86	80.03
85	84.73	85.12	84.93	84.87	85.07	84.97	85.00	85.00	85.00	85.74	85.05	85.39	85.18	84.95	85.06	85.19	84.91	85.05
90	89.72	90.11	89.92	89.88	90.05	89.96	90.00	90.00	90.00	90.28	89.95	90.11	90.11	89.97	90.04	90.10	89.99	90.04
95	94.79	95.09	94.94	94.92	95.04	94.98	95.00	95.00	95.00	94.49	94.83	94.66	94.95	95.03	94.99	94.90	95.12	95.01
99	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	97.55	98.73	98.14	98.72	99.10	98.91	98.63	99.26	98.94
RMSE	0.213	0.161	0.047	0.114	0.093	0.023	0.003	0.002	0.002	0.677	0.178	0.413	0.108	0.056	0.032	0.130	0.132	0.039

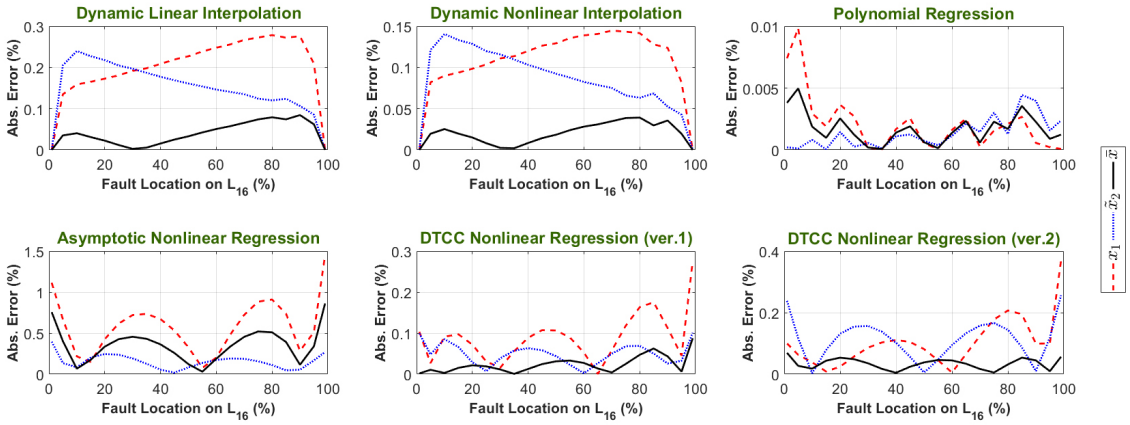


Figure 9.77: Absolute Errors of the Individual and Overall Estimates of the Best Fault Location Approaches Tabulated in Table 9.32

points of the faulty line. If the same line L_{16} is used again in this simulation, then Table 9.30 is replaced with Table 9.33.

Linear and Nonlinear Interpolation Models

The best linear and nonlinear interpolation methods are those designed with adaptive lower and upper bounds. That is, the dynamic versions given in (9.106), (9.108), and (9.109) for x_1 and $\tilde{x}_2$. This is the beauty of using interpolation approaches. Their main structures remain constant without any change. The only required additional information is Table 9.33, which is mainly used to map $T_{R_i}^{\text{clb}}$ and $T_{R_i}^{\text{cub}}$ of $T_{R_i}^{F_x}$ with the column of x_j to find the corresponding new values of x_j^{clb} and x_j^{cub}.

On the opposite side, the linear and nonlinear regression models designed for the 3ϕ fault type are not valid for the single line-to-ground fault type. Thus, there is a need to re-model (9.110), (9.111), (9.116), and (9.117) again using the observations tabulated in Table 9.33 as follows:

Polynomial-Based Regression Models

To find the optimal order of (9.110) for x_1 and $\tilde{x}_2$, the observations are regressed repeatedly for each incremental increase in d until reaching the optimal value of $\mathcal{N}$;

Table 9.33: Operating Times of the Two-End Relays Calculated Based on L-G Faults Created at Different Points of L_{16} of the IEEE 8-Bus Test System (IDMT)

Fault	**Bus 1: R_{14}**		Bus 6: **R_7**	
Location	**Short-Circuit**	**Operating**	**Short-Circuit**	**Operating**
x (%)	**Current (A)**	**Time (s)**	**Current (A)**	**Time (s)**
1	708.78	0.87327	584.18	1.84122
5	698.68	0.88945	590.36	1.79126
10	686.19	0.91065	598.40	1.73087
15	674.12	0.93250	606.83	1.67254
20	662.47	0.95501	615.65	1.61633
25	651.24	0.97816	624.86	1.56230
30	640.42	1.00195	634.45	1.51048
35	630.01	1.02636	644.42	1.46087
40	620.01	1.05135	654.77	1.41341
45	610.41	1.07693	665.50	1.36806
50	601.23	1.10297	676.61	1.32475
55	592.46	1.12945	688.10	1.28341
60	584.11	1.15625	699.97	1.24397
65	576.17	1.18332	712.24	1.20629
70	568.66	1.21049	724.90	1.17034
75	561.58	1.23761	737.97	1.13599
80	554.93	1.26454	751.46	1.10315
85	548.71	1.29111	765.41	1.07169
90	543.06	1.31652	779.66	1.04187
95	537.81	1.34128	794.43	1.01319
99	533.96	1.36020	806.56	0.99114

given in (9.152). From the regression analysis, it has been found that the 5th order and the 6th order polynomial equations provide very good estimations for x_1 and $\tilde{x}_2$, respectively. These two polynomial models are numerically expressed as follows:

$$\begin{aligned} x_1 = & -1705.7 + 6788.5T_{R_{14}} - 11569T_{R_{14}}^2 + 10389T_{R_{14}}^3 \\ & -4756.1T_{R_{14}}^4 + 883.08T_{R_{14}}^5 \end{aligned} \tag{9.161}$$

$$\begin{aligned} \tilde{x}_2 = & 850.5 - 2191.1T_{R_7} + 2953.3T_{R_7}^2 - 2400.7T_{R_7}^3 \\ & +1156.7T_{R_7}^4 - 305.3T_{R_7}^5 + 33.994T_{R_7}^6 \end{aligned} \tag{9.162}$$

Asymptotic Regression Models

As mentioned earlier, solving this model requires using LR analysis tools. The general equation is given in (9.111). The best models found for x_1 and $\tilde{x}_2$ are:

$$x_1 = 470.26034 - 706.42758 \exp\left(-0.47062T_{R_{14}}\right) \tag{9.163}$$

$$\tilde{x}_2 = -66.46632 + 462.07387 \exp\left(-1.03975T_{R_7}\right) \tag{9.164}$$

DTCC-Based Regression Models

As discussed before, these models are nonlinear. The mathematical expressions of the full and reduced versions are numerically obtained as follows:

- **Version No.1:**

$$x_1 = 1786.61139 \left[\frac{14.85657}{T_{R_{14}} - 0.87160} - 2.11462\right]^{-0.86740} \tag{9.165}$$

$$\tilde{x}_2 = 1175.02576 \left[\frac{-20.51286}{T_{R_7} - 1.85068} - 9.77998\right]^{-0.93558} \tag{9.166}$$

- **Version No.2:**

$$x_1 = 879.42522 - \frac{3480.74343}{T_{R_{14}} + 3.09310} \tag{9.167}$$

$$\tilde{x}_2 = -175.68295 + \frac{424.74963}{T_{R_7} + 0.5570} \tag{9.168}$$

Table 9.34 shows the fault locations estimated by the best models obtained for the single line-to-ground fault type. The absolute errors of the individual and overall estimates of each approach are graphically presented in Figure 9.78.

Table 9.34: Line-to-Ground Fault Locations Estimated on L_{16} by Using the Best Interpolation and Regression Approaches

Actual	Interpolation-Based Approaches						Regression-Based Approaches											
Fault	Dynamic Linear			Dynamic Nonlinear			Polynomial			Asymptotic			DTCC - ver.1			DTCC - ver.2		
Location	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$
Model	(9.106)	(9.106)	(9.99)	(9.108)	(9.109)	(9.99)	(9.146)	(9.143)	(9.99)	(9.155)	(9.156)	(9.99)	(9.157)	(9.158)	(9.99)	(9.159)	(9.160)	(9.99)
1	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.01	1.90	1.66	1.78	0.67	0.89	0.78	1.86	1.43	1.64
5	4.90	5.07	4.99	4.94	5.01	4.97	4.98	5.00	4.99	5.45	5.29	5.37	5.25	5.10	5.17	5.43	5.20	5.31
10	9.92	10.09	10.01	9.98	10.00	9.99	10.00	10.00	10.00	10.06	9.94	10.00	10.37	10.10	10.24	10.05	9.97	10.01
15	14.93	15.09	15.01	14.99	15.01	15.00	15.01	15.00	15.01	14.77	14.72	14.75	15.30	15.05	15.18	14.77	14.83	14.80
20	19.93	20.10	20.01	19.99	20.01	20.00	20.01	20.00	20.00	19.57	19.60	19.59	20.15	20.01	20.08	19.58	19.75	19.67
25	24.93	25.11	25.02	24.99	25.02	25.01	25.00	25.00	25.00	24.46	24.58	24.52	24.99	24.97	24.98	24.47	24.74	24.60
30	29.94	30.11	30.02	30.00	30.02	30.01	29.99	30.00	30.00	29.42	29.62	29.52	29.85	29.95	29.90	29.44	29.76	29.60
35	34.94	35.11	35.03	35.00	35.03	35.01	34.99	35.00	35.00	34.45	34.70	34.58	34.75	34.94	34.85	34.47	34.81	34.64
40	39.94	40.11	40.03	40.00	40.03	40.02	39.99	40.00	40.00	39.55	39.82	39.69	39.70	39.94	39.82	39.57	39.88	39.73
45	44.95	45.12	45.03	45.01	45.03	45.02	45.00	45.00	45.00	44.70	44.95	44.83	44.70	44.96	44.83	44.72	44.96	44.84
50	49.96	50.12	50.04	50.02	50.04	50.03	50.00	50.00	50.00	49.89	50.08	49.99	49.74	49.97	49.86	49.90	50.04	49.97
55	54.97	55.12	55.04	55.03	55.04	55.03	55.01	55.00	55.01	55.09	55.20	55.15	54.83	55.00	54.91	55.10	55.11	55.10
60	59.98	60.11	60.04	60.03	60.04	60.04	60.00	60.00	60.00	60.30	60.29	60.30	59.95	60.01	59.98	60.30	60.16	60.23
65	64.99	65.12	65.05	65.05	65.04	65.04	65.00	65.00	65.00	65.49	65.36	65.42	65.09	65.03	65.06	65.49	65.20	65.34
70	70.00	70.11	70.06	70.06	70.04	70.05	69.99	70.00	70.00	70.63	70.38	70.50	70.23	70.04	70.14	70.62	70.22	70.42
75	75.02	75.11	75.07	75.07	75.04	75.06	74.99	75.00	74.99	75.70	75.36	75.53	75.34	75.05	75.20	75.69	75.20	75.45
80	80.03	80.11	80.07	80.09	80.03	80.06	79.99	80.00	80.00	80.67	80.28	80.47	80.41	80.05	80.23	80.66	80.17	80.41
85	85.11	85.13	85.12	85.16	85.06	85.11	85.03	85.02	85.02	85.51	85.16	85.34	85.40	85.06	85.23	85.50	85.11	85.30
90	90.06	90.10	90.08	90.11	90.03	90.07	89.98	89.99	89.99	90.08	89.94	90.01	90.17	90.00	90.09	90.07	89.97	90.02
95	95.10	95.09	95.10	95.14	95.03	95.09	95.00	95.00	95.00	94.49	94.67	94.58	94.82	94.97	94.89	94.48	94.83	94.65
99	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	97.82	98.41	98.11	98.37	98.92	98.65	97.82	98.68	98.25
RMSE	0.060	0.102	0.052	0.063	0.031	0.044	0.011	0.004	0.007	0.538	0.332	0.430	0.288	0.056	0.170	0.529	0.199	0.360

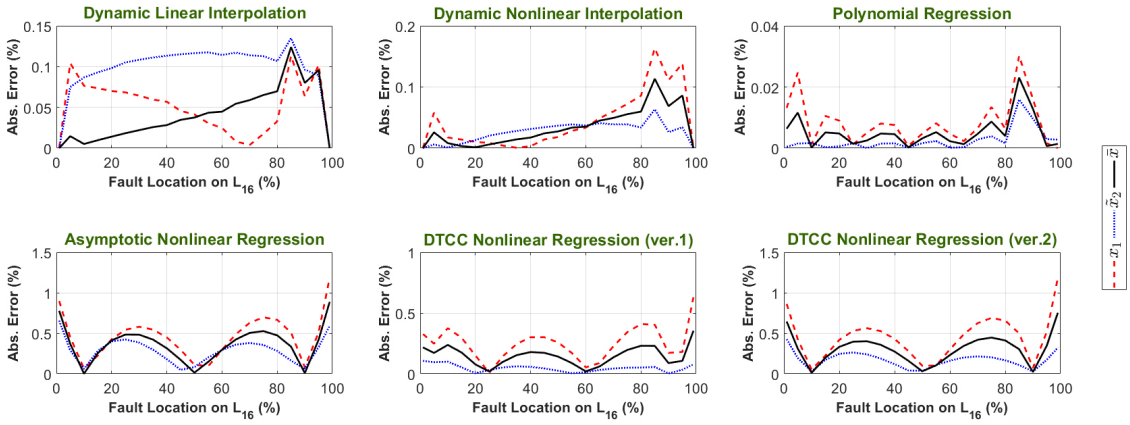


Figure 9.78: Absolute Errors of the Individual and Overall Estimates of the Best Fault Location Approaches Tabulated in Table 9.34

9.8.2.3 Using Other Standard TCCCs for DOCRs

It is important to know that the whole process depends on the time-current characteristic curve (TCCC) used in all ϱ DOCRs. This fact has already been revealed in (9.100)-(9.102). The standard TCCC used in the preceding simulations was IDMT where the $\{\alpha, \beta, \gamma\}$ coefficients are tabulated in Table 4.2 and graphically depicted in Figure 4.7a.

Thus, all that it needs is to find the optimal settings of TMS and PS again based on the new TCCC used in DOCRs. There is no need to do any fault analysis if the unified TCCC for all ϱ DOCRs is changed. However, the operating times measured from each two-end relays should be re-calculated since their inverse characteristic curves are changed. To achieve that, the same BBO-LP optimization algorithm is used again for this part of the experiment.

Now, let's say that the IEC very inverse TCCC is used instead of IDMT. From Table 4.2, the new $\{\alpha, \beta, \gamma\}$ coefficients are respectively equal to $\{1, 13.5, 0\}$. Initiating BBO-LP with these new parameters gives the updated optimal settings of TMS and PS. These settings are listed in Table 9.35 for all the 14 DOCRs connected in the IEEE 8-bus test system shown in Figure 9.68.

Because L_{16} is the branch selected for this numerical experiment, so the operating

Table 9.35: Optimal Relay Settings of the IEEE 8-Bus Test System When the IEC Very Inverse TCCC Is Implemented

Primary	**Optimal Settings**	
Relay	**TMS**	**PS (A)**
R_1	0.1363	0.5
R_2	0.1342	1.0
R_3	0.1844	0.8
R_4	0.1313	1.5
R_5	0.1302	1.0
R_6	0.2466	0.6
R_7	0.1373	1.5
R_8	0.1294	0.8
R_9	0.2454	0.8
R_{10}	0.1292	1.5
R_{11}	0.1686	0.6
R_{12}	0.1493	1.0
R_{13}	0.1000	0.5
R_{14}	0.1000	1.5
$\sum_{i=1}^{14} T_i^{\text{pr}}$	5.063066 s	

times of R_7 and R_{14} are calculated using (4.4) as follows:

$$T_{R_7} = \frac{13.5(0.1373)}{\left[\frac{\left(\frac{5}{800}\right)I_{R_7}}{1.5}\right]^1 - 1} = \frac{1.8536}{0.0042I_{R_7} - 1} \tag{9.169}$$

$$T_{R_{14}} = \frac{13.5(0.1)}{\left[\frac{\left(\frac{5}{800}\right)I_{R_{14}}}{1.5}\right]^1 - 1} = \frac{1.3500}{0.0042I_{R_{14}} - 1} \tag{9.170}$$

In the preceding part of the experiment, it has been seen that the gradient values of I_{R_7} and $I_{R_{14}}$ listed in Table 9.30 can generate the required vectors of the two independent variables T_{R_7} and $T_{R_{14}}$, so the 3ϕ fault locator for L_{16} can be modeled. Similarly, for this part, using I_{R_7} and $I_{R_{14}}$ listed in Table 9.33 can generate the required vectors for the single line-to-ground fault type.

If the 3ϕ fault type is considered again for this TCCC, then Table 9.36 is produced.

This table is the heart of all the incoming interpolation and regression approaches.

Linear and Nonlinear Interpolation Models

For the dynamic linear and nonlinear interpolation approaches, the closest lower and upper bounds to the actual operating time $T_{R_i}^{F_x}$ are mapped with the corresponding closest lower and upper bounds of x_j listed in Table 9.36. Thus, the parameters $\{T_{R_i}^{\text{clb}}, T_{R_i}^{\text{cub}}, x_j^{\text{clb}}, x_j^{\text{cub}}\}$ can be defined. These values are then substituted into (9.106), (9.108), and (9.109) for both relays. The average estimate $\bar{x}$ can be determined by using (9.99).

Polynomial-Based Regression Models

Again, for this approach, the minimization procedure is used to find the best polynomial orders for x_1 and $\tilde{x}_2$. This can be done either by finding the smallest value of $\mathcal{N}$ via optimizing (9.152) or reaching a satisfying value. If this selection procedure is used, then the following two optimal polynomial models are obtained for x_1 and $\bar{x}_2$:

$$\begin{aligned} x_1 = & -286.53 + 2026.6T_{R_{14}} - 6180.6T_{R_{14}}^2 + 12649T_{R_{14}}^3 \\ & -17055T_{R_{14}}^4 + 14510T_{R_{14}}^5 - 7037.7T_{R_{14}}^6 \\ & +1480.8T_{R_{14}}^7 \end{aligned} \tag{9.171}$$

$$\begin{aligned} \tilde{x}_2 = & 375.09 - 1535.9T_{R_7} + 2879.8T_{R_7}^2 - 3230.3T_{R_7}^3 \\ & +2005.1T_{R_7}^4 - 532.67T_{R_7}^5 \end{aligned} \tag{9.172}$$

Asymptotic Regression Models

As said before, the main objective of using nonlinear regression approaches is to avoid searching for optimal d as faced with the polynomial regression approach. The best nonlinear regression models found, by solving (9.111) for x_1 and $\tilde{x}_2$, are:

$$x_1 = 119.78204 - 277.46450 \exp\left(-2.81068T_{R_{14}}\right) \tag{9.173}$$

$$\tilde{x}_2 = -43.56130 + 383.31006 \exp\left(-3.25260T_{R_7}\right) \tag{9.174}$$

Table 9.36: Operating Times of the Two-End Relays Calculated Based on 3ϕ Faults Created at Different Points of L_{16} of the IEEE 8-Bus Test System (Very Inverse)

Fault	**Bus 1: R_{14}**		**Bus 6: R_7**	
Location	**Short-Circuit**	**Operating**	**Short-Circuit**	**Operating**
x (%)	**Current (A)**	**Time (s)**	**Current (A)**	**Time (s)**
1	1291.33	0.30518	907.51	0.65928
5	1255.12	0.31605	933.78	0.63439
10	1210.68	0.33049	967.10	0.60539
15	1167.45	0.34586	1001.01	0.57848
20	1125.37	0.36226	1035.53	0.55344
25	1084.38	0.37981	1070.66	0.53009
30	1044.45	0.39862	1106.43	0.50825
35	1005.51	0.41885	1142.86	0.48779
40	967.56	0.44064	1179.99	0.46856
45	930.55	0.46419	1217.84	0.45046
50	894.47	0.48970	1256.47	0.43337
55	859.33	0.51740	1295.90	0.41722
60	825.11	0.54756	1336.20	0.40190
65	791.84	0.58046	1377.41	0.38737
70	759.56	0.61640	1419.59	0.37354
75	728.30	0.65570	1462.80	0.36036
80	698.14	0.69869	1507.12	0.34777
85	669.15	0.74568	1552.64	0.33573
90	641.52	0.79675	1599.28	0.32423
95	615.30	0.85213	1647.32	0.31317
99	595.48	0.89939	1686.75	0.30465

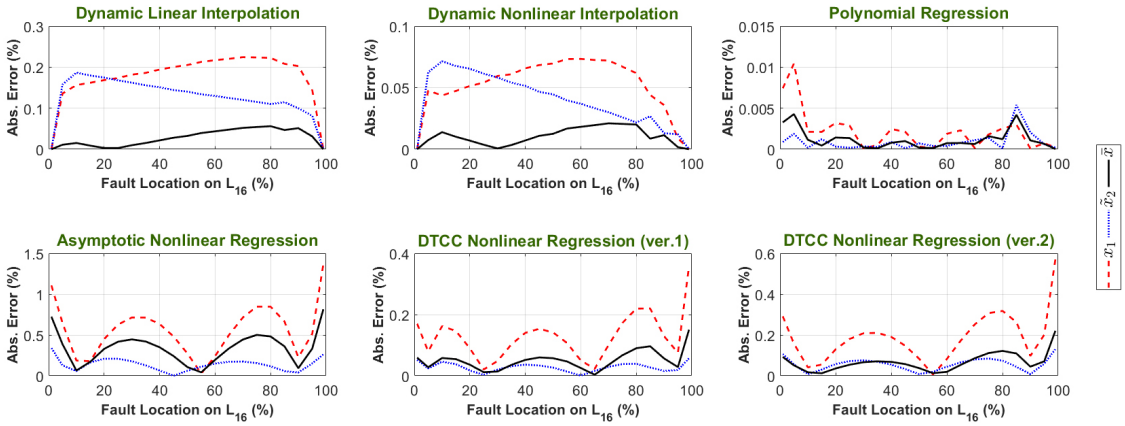


Figure 9.79: Absolute Errors of the Individual and Overall Estimates of the Best Fault Location Approaches Tabulated in Table 9.37

DTCC-Based Regression Models

Again, the models listed under this approach are nonlinear. The general expressions of the full and reduced versions are given in (9.116) and (9.117), respectively. The numerical solutions obtained for the full and reduced versions are given below:

- **Version No.1:**

$$x_1 = 958.24700 \left[\frac{3.19829}{T_{R_{14}} - 0.30336} + 5.75792 \right]^{-0.94376} \tag{9.175}$$

$$\tilde{x}_2 = 292.52366 \left[\frac{-1.80141}{T_{R_7} - 0.66651} - 2.09119 \right]^{-1.02132} \tag{9.176}$$

- **Version No.2:**

$$x_1 = 177.70116 - \frac{85.54377}{T_{R_{14}} + 0.17973} \tag{9.177}$$

$$\tilde{x}_2 = -129.17184 + \frac{107.19445}{T_{R_7} + 0.16488} \tag{9.178}$$

The estimations obtained from the best models, for different 3ϕ faults created on L_{16} of the IEEE 8-bus test system, are tabulated in Table 9.37. The TCCC of the two-end DOCRs (i.e., R_7 and R_{14}) is modeled using the IEC very inverse standard given in Table 4.2. The absolute error plots of these estimations are shown in Figure 9.79 for each approach.

Table 9.37: 3ϕ Fault Locations Estimated on L_{16} by Using the Best Interpolation and Regression Approaches (Very Inverse)

Actual	Interpolation-Based Approaches						Regression-Based Approaches											
Fault	Dynamic Linear			Dynamic Nonlinear			Polynomial			Asymptotic			DTCC - ver.1			DTCC - ver.2		
Location	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$	x_1	$\tilde{x}_2$	$\bar{x}$
Model	(9.106)	(9.106)	(9.99)	(9.108)	(9.109)	(9.99)	(9.146)	(9.143)	(9.99)	(9.155)	(9.156)	(9.99)	(9.157)	(9.158)	(9.99)	(9.159)	(9.160)	(9.99)
1	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	2.11	1.34	1.72	0.83	1.05	0.94	1.29	0.89	1.09
5	4.86	5.16	5.01	4.95	5.06	5.01	4.99	5.00	5.00	5.65	5.13	5.39	5.08	4.98	5.03	5.16	4.94	5.05
10	9.84	10.19	10.02	9.96	10.07	10.01	10.00	10.00	10.00	10.19	9.94	10.06	10.16	9.95	10.06	10.04	9.99	10.02
15	14.84	15.18	15.01	14.95	15.07	15.01	15.00	15.00	15.00	14.82	14.84	14.83	15.15	14.96	15.05	14.95	15.03	14.99
20	19.83	20.17	20.00	19.95	20.07	20.01	20.00	20.00	20.00	19.55	19.79	19.67	20.09	19.98	20.04	19.87	20.06	19.96
25	24.83	25.17	25.00	24.95	25.06	25.00	25.00	25.00	25.00	24.37	24.79	24.58	25.02	25.00	25.01	24.82	25.07	24.95
30	29.82	30.16	29.99	29.94	30.06	30.00	30.00	30.00	30.00	29.29	29.82	29.55	29.95	30.02	29.99	29.79	30.08	29.93
35	34.81	35.16	34.98	34.94	35.05	35.00	35.00	35.00	35.00	34.29	34.87	34.58	34.90	35.03	34.96	34.79	35.07	34.93
40	39.81	40.15	39.98	39.93	40.05	39.99	40.00	40.00	40.00	39.37	39.94	39.65	39.86	40.04	39.95	39.81	40.05	39.93
45	44.80	45.14	44.97	44.93	45.05	44.99	45.00	45.00	45.00	44.52	45.00	44.76	44.85	45.03	44.94	44.85	45.03	44.94
50	49.79	50.14	49.97	49.93	50.04	49.99	50.00	50.00	50.00	49.73	50.06	49.89	49.86	50.03	49.94	49.92	50.01	49.96
55	54.79	55.13	54.96	54.93	55.04	54.98	55.00	55.00	55.00	54.97	55.11	55.04	54.89	55.01	54.95	54.99	54.98	54.99
60	59.78	60.13	59.96	59.93	60.04	59.98	60.00	60.00	60.00	60.24	60.15	60.20	59.95	60.00	59.97	60.08	59.96	60.02
65	64.78	65.12	64.95	64.93	65.03	64.98	65.00	65.00	65.00	65.50	65.17	65.34	65.02	64.98	65.00	65.17	64.93	65.05
70	69.78	70.12	69.95	69.93	70.03	69.98	70.00	70.00	70.00	70.71	70.17	70.44	70.10	69.97	70.04	70.25	69.92	70.09
75	74.78	75.12	74.95	74.93	75.03	74.98	75.00	75.00	75.00	75.85	75.16	75.50	75.17	74.96	75.07	75.31	74.92	75.11
80	79.78	80.11	79.94	79.94	80.02	79.98	80.00	80.00	80.00	80.85	80.12	80.48	80.22	79.96	80.09	80.32	79.93	80.12
85	84.79	85.11	84.95	84.96	85.03	84.99	85.00	85.01	85.00	85.66	85.06	85.36	85.22	84.97	85.10	85.26	84.96	85.11
90	89.80	90.10	89.95	89.96	90.01	89.99	90.00	90.00	90.00	90.23	89.96	90.09	90.13	89.98	90.06	90.10	89.99	90.04
95	94.86	95.08	94.97	94.99	95.01	95.00	95.00	95.00	95.00	94.49	94.85	94.67	94.92	95.02	94.97	94.80	95.06	94.93
99	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	97.63	98.74	98.19	98.64	99.06	98.85	98.43	99.13	98.78
RMSE	0.184	0.135	0.033	0.056	0.045	0.012	0.003	0.001	0.002	0.651	0.159	0.396	0.149	0.031	0.060	0.224	0.064	0.082

9.8.3 Final Structure of the Proposed Fault Locator

In the preceding comprehensive experiment, three different scenarios have been covered to describe the working principle of this fault locator in details. The backbone of the entire technique is the relationship between the fault location and the operating times measured by two DOCRs installed on both ends of the faulty branch. All the interpolation and regression approaches are designed based on this relationship. Thus, the core of the three scenarios covered in the experiment is to generate Tables 9.30, 9.33, and 9.36. However, creating these tables requires proceeding some fundamental stages for each dynamic change on the network condition. The logical reason behind that comes from the fact that the optimal TMS and PS of DOCRs are adaptively adjusted for any topological or operational change.

To summarize all these stages, Figure 9.80 shows the final structure of the proposed fault locator. As can be seen from that diagram, it has an automatic mechanism to detect any dynamic change. They could be topology changes (such as opening branches and isolating busbars) or operational changes (such as changing the set-points of generating units and varying the settings of capacitor banks). For the topological change, the network needs to be re-configured again. Also, all the P/B relay pairs of the new topology need to be identified. If it is an operational change, then the preceding two sub-stages are bypassed. However, the network PF sub-stage needs to be updated for any topological or operational type. Then, to solve the ORC problem of the given network, it is important to carry out a 3ϕ short-circuit analysis with considering the proper coordination criteria of those listed in Table 4.1. Finally, the optimal values of TMS and PS of all ϱ DOCRs connected in the network can be obtained by using any global n-dimensional optimization algorithm. This adaptive ORC stage continuously checks if there is any topological or operational change, so the preceding steps are repeated again and again. This stage is extensively studied in the literature, and it is known as an **adaptive coordination** strategy.

The responsibility of the second stage is to design all the required fault location models for each two-end relays of all ß branches. Thus, the latest network state updated by PF is sent to the second stage. This information is utilized during doing another short-circuit analysis. This analysis is carried out for all the fault types, and it is repeated until covering all the branches. Furthermore, different predefined fault

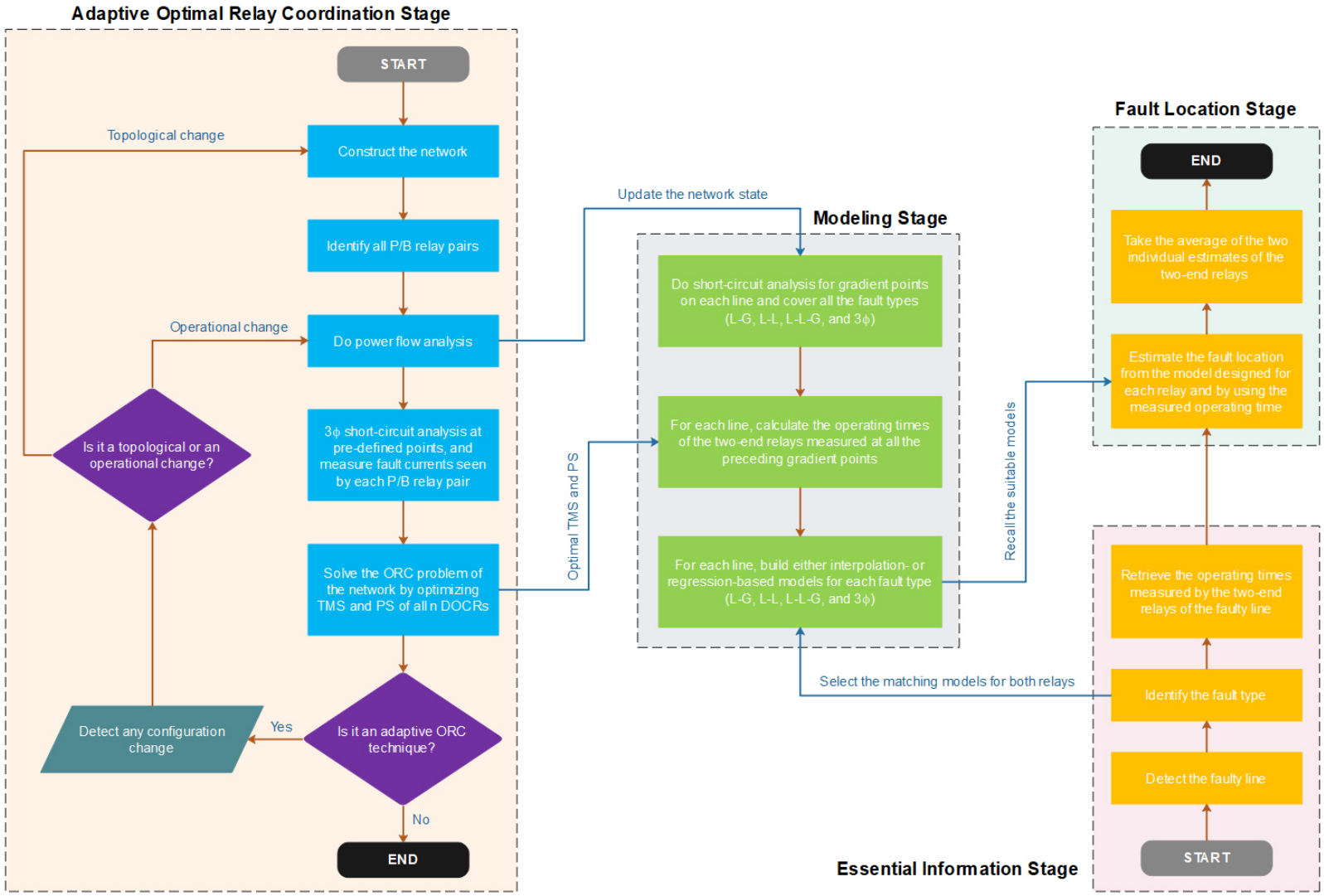


Figure 9.80: Final Structure of the Proposed Adaptive Fault Locator Including All the Required Stages

locations are considered for each branch. They should be gradually distributed from 1% (i.e., at the near-end point) to 99% (i.e., at the far-end point). Then, by knowing the optimal TMS and PS of each relay, the operating times of these relays can be calculated for any fault and at any point of any branch. These predefined operating times and fault locations can then be used to design all the required fault location models. These models could be designed based on linear interpolation, nonlinear interpolation, linear regression, or nonlinear regression.

The third and fourth stages contain optimization-free algorithms because all the optimization tasks are done in the previous two stages. The third stage is responsible to detect the faulty branch and the type of that fault. Also, the operating times recorded by the two-end relays of the faulty branch are retrieved again, which are sent to the fourth stage. These operating times act as the independent variables of the fault location models designed before in the second stage. The last step is to take the average of the two fault estimations calculated from the corresponding two-end relays.

9.8.4 Overall Accuracy and Uncertainty Errors

It is obvious that, in most cases, the estimation errors of the two-end relays can be minimized if the average value is taken using (9.97) or (9.99). However, many sources of uncertainty could be faced in real-world applications. Based on this, the location of actual faults could not exactly match the DTCC curves of the two-end relays. Assume that $\wp$ sources of uncertainty are disturbing the operation of the network. Then, the deviation between the actual fault location x and its estimate $\bar{x}$ can be mathematically expressed as:

$$x = \bar{x} + \sum_{j=1}^{\wp} \varepsilon_j \tag{9.179}$$

That is, these $\wp$ disturbances are translated as the sum of errors $\sum_{j=1}^{\wp} \varepsilon_j$ associated with the final estimation model. With unbiased residuals that satisfy the normality test, the actual fault locations are supposed to be normally distributed above and below the fitted DTCC curves of the two-end relays. However, this claim has not been proven yet.

Regardless of the subject matter of uncertainty errors, the technique can still

give a good estimation. For example, the impedance of branches can be affected by the surrounding temperature [12, 13]. It is known that the short-circuit current decreases as the impedance increases, and the later one increases as the surrounding temperature increases. Based on this, during the winter season, the surrounding temperature decreases and thus the impedance decreases as well. Therefore, the short-circuit current increases and the operating time of the ith relay decreases. Suppose, by chance, the operating time received from the ith relay intersects with one of the predefined observations. Thus, by taking into account the temperature effect, the correct predefined fault location that should be mapped with the measured operating time could become the closest lower or upper bound (i.e., x^{clb} or x^{cub}). However, interpolating these points will give an estimated location near that new bound. The same thing happens with linear and nonlinear regressions. This part of the study could be covered in future work where the most influent uncertainty sources can be highlighted and then trying either to eliminate or minimize their effects. One of the possible tools is to model the entire system with the temperature/frequency-based (TFB) technique reported in [12, 13].

9.8.5 Further Discussion

Although the major parts of this technique are covered in Figure 9.80 there are many other enhancements could be made to improve its performance. For example:

- *How many predefined points need to be covered for each branch? In this study, gradient points of steps 5% are taken between the near-end and the far-end points of each ith relay. But, what is the best step-size? Is it 5%, 10%, or unequal steps and randomly selected points?*
- *In the modeling part, should the two-end relays be modeled using the same approach? What if the first model is designed using the dynamic nonlinear interpolation approach and the second model is designed using a dth order polynomial regression approach?*
- *What if one or more of "in-between" terms of polynomial regression models is removed instead of the last term*[44]*?*

[44]Such models are known as **non-hierarchical polynomial regression models**. That is, dealing

- *Again, in the regression models, what if a piecewise linear or quadratic polynomial equations are used instead of increasing d up to 7?*
- *Could AI, such as ANNs and SVMs, be good add-ons to precisely estimate fault locations?*
- *In Figure 9.80, is it feasible to deactivate the adaptive link of the ORC stage?* This question is raised here because it is known that the fault location stage is a post-fault action, and thus there is no need to have a very fast fault locator. This is one of the main differences between protective relays and fault locators listed in Table 9.29.

Also, there are other pending points that need to be covered to ensure that the operation of Figure 9.80 is fully trusted. Such points are:

- Mal-operation case of primary DOCRs.
- Independent detection of in-zone and out-zone faults.
- Independent detection of fault types.
- Transient network topologies due to unequal operating speeds of CBs during solving the ORC problem.
- Networks with multi-terminal branches.
- Networks with FACTS devices.

All these points can be considered as open doors that need to be closed by conducting more future researches.

with polynomial equations where some lower-order terms are missing. For example, using a septic polynomial equation without the quintic term.

Chapter 10

Novel Machine Learning Computing Systems with Some Applications in Electric Power Systems

Nowadays, couples of computing systems have been introduced in the literature to perform many applications, such as function approximation, pattern classification, categorization/clustering, forecasting/prediction, control, and optimization. For example, linear regression (LR), nonlinear regression (NLR), and all the flavors of artificial neural networks (ANNs) and support vector machines (SVMs). LR is commonly used for simple data where the relation between its coefficients is linear, while NLR is used when that relation is nonlinear. ANNs and SVMs are more efficient and they can be used for complicated applications. However, each one of these approaches has its own strengths and weaknesses. In this chapter, a new computing system called "**universal functions originator** (**UFO**)" is introduced. This system can generate highly complicated mathematical models as well as simplifying them down to very concise models. The operation is done automatically through two independent optimization algorithms. Different arithmetic operators (including addition, subtraction, multiplication, and division) can be entered into the search pool. Also, any analytical function (including basic, logarithmic, exponential, trigonometric, and hyperbolic functions) can be dragged into that pool. UFO has been mathematically designed and practically tested with function approximation problems. It can also be used for the applications listed above, including **anomaly detection**, **function complication**, **function simplification**, **dimension expansion**, **dimension reduction**, and ***high*-dimensional function visualization**. This novel computing system shows an impressive performance with many promising uses and distinct capabilities. This chapter reveals the mechanism of UFO and solves some numerical problems via an advanced graphical user interface (GUI) designed just to validate the process of this computing system. At the end of the chapter, some UFO-based hybrid computing systems are introduced.

10.1 Introduction to Machine Learning Tools

In modern sciences, especially in engineering and computer/data science, AI is a wide branch of research. This branch comprises many fields, such as [324]: 1. **problem-solving**, 2. **knowledge**, **reasoning**, and **planning**, 3. **uncertain knowledge and reasoning**, 4. **learning**, 5. **communicating**, **perceiving**, and **acting**, and 6. **robotics**. Also, there are many tools used in AI, such as: 1. search and optimization, 2. artificial neural networks (ANNs), 3. **logic**, 4. **statistical learning methods** and classifiers, and 5. **uncertain reasoning** through **probabilistic methods**. For each type of these tools, there are many sub- and sub-sub-tools. For example, optimization algorithms can be divided into three main groups, and each group contains many algorithms. These three groups are: 1. classical optimization algorithms, 2. meta-heuristic optimization algorithms, and 3. hybrid optimization algorithms. Each one of these groups contains many types of optimization algorithms. Some examples of meta-heuristic optimization algorithms are: genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), etc. Moreover, each specific optimization algorithm comes with different sub-types and versions. For instance, GA, micro GA (μGA), and stud GA (sGA) [345].

Similar to the optimization field, ANN is a wide topic of AI. Currently, there are many ANN types presented in the literature. These types can be categorized into two categories: 1. **feed-forward networks** and 2. **recurrent/feedback networks**. Also, each one of them has multiple sub-types. For example, single-layer perceptron (SLP), multilayer perceptron (MLP), **probabilistic neural network** (**PNN**), **autoencoder** (**AE**), time-delay neural network (TDNN), and convolutional neural network (CNN), all are sub-types of feed-forward neural networks. Hopfield, **Boltzmann machine**, long short-term memory (LSTM), **self-organizing map** (**SOM**), **learning vector quantization** (**LVQ**), **adaptive resonance theory** (**ART**), and **echo state network** (**ESN**), all are sub-types of recurrent/feedback neural networks. Each one of these networks has its own strengths and weaknesses, and thus a wise selection is needed for each specific application. As a summary, ANNs can be applied in the following applications: 1. function approximation, 2. categorization/-clustering, 3. pattern classification, 4. control, 5. prediction/forecasting, 6. anomaly detection, and 7. optimization. The first three applications are the heart of what is

called machine learning (ML), which was coined by Arthur Samuel [331].

ML is a major field of AI, which is exactly the fourth field listed in the first paragraph; i.e., the learning field. It is not revealing a secret to say that ML is frequently used by data scientists and engineers to do many crucial applications. If the features of a given dataset are automatically extracted by the learning algorithm itself, then this special type of ML is called **deep learning** (**DL**). The most commonly used ML algorithms are: 1. linear regression (LR), 2. nonlinear regression (NLR), 3. ANNs-based algorithms, 4. support vector machines (SVMs), 5. **decision tree**, 6. **naive Bayes classifiers**, 7. **k-nearest neighbors** (**kNN**) **classifier**, 8. **k-means clustering**, 9. **random forest**, 10. **gradient and adaptive boosting algorithms**, 11. **dimensionality reduction algorithms**, 12. **ensemble methods**, 13. **singular value decomposition** (**SVD**), 14. **independent component analysis** (**ICA**), and 15. **principal component analysis** (**PCA**). These ML algorithms can be grouped into four broad groups: 1. **supervised learning**, 2. **unsupervised learning**, 3. **semi-supervised learning**, and 4. **reinforcement learning**.

By referring to the preceding ML algorithms, each one of them has its own strengths and weaknesses, advantages and disadvantages, or pros and cons. The next subsections briefly highlight LR, NLR, ANNs, and SVMs.

10.1.1 Linear Regression

Linear regression is the simplest and most popular ML algorithm. It is a fundamental step in ML field, which can be carried out by using very basic programs to determine its optimal coefficients. LR is commonly used if the given data is simple. The word "linear" does not mean that the regression model is restricted to the first-order polynomial equation. Rather, it means that the mathematical model expressed by LR has linear coefficients. Thus, any polynomial equation with any degree can be used as a basis to create the regression model. For example, if the first-order polynomial equation is used, then the fitted curve can be mathematically expressed by the following linear equation:

$$\hat{y} = \beta_0 + \beta_1 x \tag{10.1}$$

where $\hat{y}$ is the response, x is the predictor, and βs are the model coefficients.

If the regression model is constructed as a dth order polynomial equation, then

still LR should be used to fit that model. The reason behind this is that the relation between its βs is linear. The following equation explains this point:

$$\hat{y} = \sum_{l=0}^{d} \beta_l x^l = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d \tag{10.2}$$

Thus, linear, quadratic, cubic, biquadratic, quartic, etc, all are regressed using LR. If the LR model consists of multiple predictors, then it is known as **multiple linear regression** (**MLR**). Suppose that n predictors are involved. Then, the first-order MLR model can be expressed as follows:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{10.3}$$

If there is an interaction between x_1 and x_2, then (10.3) becomes:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \xi_{1,2}\ x_1 x_2 + \cdots + \beta_n x_n \tag{10.4}$$

where $\xi_{1,2}$ is the coefficient of the interaction term between x_1 and x_2.

In general, the LR method has many applications. For example, it can be used as a function approximation to estimate the actual response y. Also, it can be used in forecasting as a deterministic prediction tool [254]. The mathematical models of LR are very simple, which can be created without using any analytical function[1]. These LR models can be easily embedded inside an external software or hardware with very limited memory, and thus they can be processed very fast in any basic computing machine. Add to that, their mathematical models are readable and can be interpreted to explain many mysterious facts hidden behind the scene. For example, in (10.1), the intercept β_0 means a fixed value that could lead the analysts to point out the initial state or figure of a given data. Also, the slope β_1 can help the analysts to estimate the rate of change. In spite of all these wonderful strengths, LR has many inevitable drawbacks and limitations. One of the main inherent weaknesses of LR is that it permits only polynomial equations to be used in its regression process. For n-dimensional problems, LR must be expressed to include part of, or all, the following

[1]This is true unless the original predictors are transformed before being fed into the regression analysis [214].

terms:

$$
\begin{aligned}
\hat{y}(X) \;=\;& \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \\
&+ \gamma_1 x_1^2 + \gamma_2 x_2^2 + \cdots + \gamma_n x_n^2 \\
&+ \xi_{1,1} x_1 x_2 + \xi_{1,2} x_1 x_3 + \cdots + \xi_{1,n-1} x_1 x_n \\
&+ \xi_{2,1} x_2 x_3 + \xi_{2,2} x_2 x_4 + \cdots + \xi_{2,n-2} x_2 x_n \\
&\qquad\qquad \vdots \\
&+ \xi_{n-1,1} x_{n-1} x_n \\
&+ \zeta_{1,1} x_1^2 x_2 + \zeta_{1,2} x_1^2 x_3 + \cdots + \zeta_{1,n-1} x_1^2 x_n \\
&+ \zeta_{2,1} x_2^2 x_1 + \zeta_{2,2} x_2^2 x_3 + \cdots + \zeta_{2,n-1} x_2^2 x_n \\
&\qquad\qquad \vdots \\
&+ \zeta_{n,1} x_n^2 x_1 + \zeta_{n,2} x_n^2 x_3 + \cdots + \zeta_{n,n-1} x_n^2 x_{n-1} \\
&+ \varphi_{1,1} x_1^2 x_2^2 + \varphi_{1,2} x_1^2 x_3^2 + \cdots + \varphi_{1,n-1} x_1^2 x_n^2 \\
&\qquad\qquad \vdots
\end{aligned}
\tag{10.5}
$$

Thus, to fit an exponential or an inverse response, a higher-order polynomial equation is needed; without guarantee getting a satisfying result. The other possible option is to apply the transformation technique, which is useful in some data. For other more complex nonlinear responses, LR is not suitable.

10.1.2 Non-Linear Regression

To resolve the nonlinearity problem associated with LR, the analysts are forced to use NLR in case the results are wanted to be rendered as readable mathematical equations. Nowadays, there are many open-source/commercial programming languages and software available to the researchers to define their own nonlinear models, and then some built-in optimization algorithms are executed to obtain the optimal coefficients of their models. Such commercial software and programming languages are MATLAB, MINITAB, SAS, SPSS, **NCSS**, and **Stata**. On the opposite side, many open-source packages and libraries can be used in R and Python languages. A part of these software and programming languages provide some ready-made regression models. For example, in MINITAB, users can select many nonlinear models available

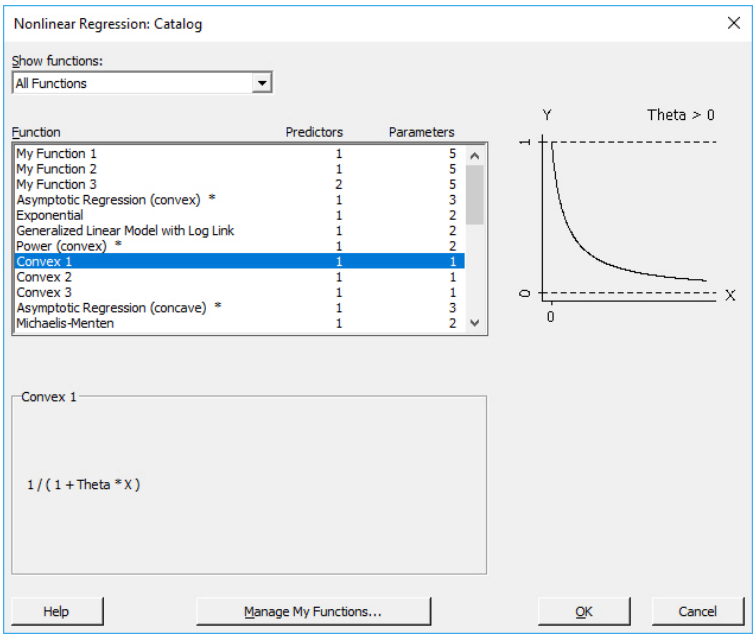


Figure 10.1: MINITAB's Internal Catalog of Ready-Made NLR Functions

inside an internal catalog as shown in Figure 10.1. Some of these ready-made models are listed below:

- **Asymptotic Regression:**

$$\hat{y} = \theta_1 - \theta_2 \ \exp\left(-\theta_3 x\right) \tag{10.6}$$

- **Exponential:**

$$\hat{y} = \theta_1 \ \exp\left(\theta_2 x\right) \tag{10.7}$$

- **Generalized Linear Model with Log Link:**

$$\hat{y} = \exp\left(\theta_1 + \theta_2 x\right) \tag{10.8}$$

- **Power:**

$$\hat{y} = \theta_1 x^{\theta_2} \tag{10.9}$$

- **Michaelis-Menten:**

$$\hat{y} = \frac{\theta_1 x}{\theta_2 + x} \tag{10.10}$$

- **Gompertz Growth:**

$$\hat{y} = \theta_1 \ \exp\Big(-\exp\left(\theta_2 - \theta_3 x\right)\Big) \tag{10.11}$$

- **Logistic Growth:**

$$\hat{y} = \theta_1 + \frac{\theta_2 - \theta_1}{1 + \exp\left(\frac{x-\theta_3}{\theta_4}\right)} \tag{10.12}$$

- **Loglogistic Growth:**

$$\hat{y} = \theta_1 + \frac{\theta_2 - \theta_1}{1 + \exp\left(\theta_4 \ln\left(\frac{x}{\theta_3}\right)\right)} \tag{10.13}$$

- **Weibull Growth:**

$$\hat{y} = \theta_1 + (\theta_2 - \theta_1) \exp\left(-\theta_3 x^{\theta_4}\right) \tag{10.14}$$

- **1-Parameter Sigmoid:**

$$\hat{y} = 1 - \exp\left(-x^{\theta}\right) \tag{10.15}$$

- **2-Parameter Sigmoid 1:**

$$\hat{y} = 1 - \exp\left(-\theta_1 x^{\theta_2}\right) \tag{10.16}$$

- **2-Parameter Sigmoid 2:**

$$\hat{y} = 1 - \exp\left(-\exp\left(\theta_1 - \theta_2 x\right)\right) \tag{10.17}$$

- **Bragg Equation:**

$$\hat{y} = \theta_1 + (\theta_2 - \theta_1) \exp\left(-\theta_3 \left(x - \theta_4\right)^2\right) \tag{10.18}$$

- **Holiday:**

$$\hat{y} = \frac{1}{\theta_1 + \theta_2 \left(x - \theta_3\right)^2} \tag{10.19}$$

- **Rational Polynomial Function 1:**

$$\hat{y} = \frac{\theta_1 + \theta_2 x}{1 + \theta_3 x + \theta_4 x^2} \tag{10.20}$$

- **Rational Polynomial Function 2:**

$$\hat{y} = \frac{\theta_1 + \theta_2 x + \theta_3 x^2}{1 + \theta_4 x + \theta_5 x^2} \tag{10.21}$$

- **Fourier 1:**

$$\hat{y} = \theta_1 \ \cos\left(x + \theta_4\right) + \theta_2 \ \cos\left(2x + \theta_4\right) + \theta_3 \tag{10.22}$$

- **Fourier 2:**

$$\begin{aligned} \hat{y} &= \theta_1 \cos(x+\theta_4) + \theta_2 \cos(2x+\theta_4) \\ &\quad + \theta_3 \cos(3x+\theta_4) \end{aligned} \tag{10.23}$$

- **Nernst Equation:**

$$\hat{y} = \theta_1 - \theta_2 \Big[\ln(x_1+\theta_3) - \ln(x_2) \Big] \tag{10.24}$$

- **Enzyme Reaction:**

$$\hat{y} = \frac{x_1 x_2}{\theta_1 + \theta_2 x_1 + \theta_3 x_1 x_2 + \theta_4 x_1 x_3} \tag{10.25}$$

where θs are the NLR coefficients that need to be obtained through fitting the data by some internal optimization algorithms.

Although NLR can solve the nonlinearity issue of LR it needs many adjustments from the analysts. It can be said that the biggest barrier that forces many researchers to flee away from this approach is its complexity in choosing the appropriate regression model. Even if the correct model is selected, NLR might fail to fit the data if the initial parameters are not suitable. Also, the lower and upper limits of each coefficient play an important role in finding the optimal results.

10.1.3 Artificial Neural Networks

A brief introduction to ANNs is given in Chapter 5. Compared with LR and NLR, ANNs are more powerful and efficient in many highly complicated problems. They can be used to solve all the technical problems, difficulties and challenges faced with LR and NLR. The knowledge of these networks, which represents the response or output variable of LR and NLR, can be built without referring to any mathematical expression. Thus, ANNs act as black-boxes. Actually, this is a double-edged sword. It can be seen as a source of strength because ANNs can estimate the system response directly without trying to express any weary mathematical equation. However, this approach makes the whole process secret and nobody can know what is going on inside these mysterious networks. Based on this, it limits many possible applications; as will be discussed later in UFO. Also, some of the drawbacks associated with ANNs

are concentrated in the selection of features, topology, number of hidden layers, and number of neurons assigned to each layer; as previously shown in Figure 5.4. Add to that, it is hard to know the best learning algorithm and its settings or hyperparameters. Moreover, the best set of activation functions used in the hidden layers and the output layer is an open question. Furthermore, a long CPU time is required to train ANNs with big data. These data should be normalized before feeding them to ANNs. Last but not least, ANNs do not guarantee to reach optimal results nor providing 100% reliability [68, 182].

10.1.4 Support Vector Machines

Again, a brief introduction to SVMs is given in Chapter 5. The main drawbacks of SVMs are concentrated in the selection of the kernel functions and their hyperparameters. Also, the processing speed required for the training and testing stages is relatively high. Moreover, SVMs lack transparency of results and they suffer from discrete data [78].

10.2 Universal Functions Originator

All the facts listed in the preceding subsections for LR, NLR, ANNs, and SVMs, give us enough motivation to think about a technique that could solve some of the inherent weaknesses of the existing computing systems. ANNs are a good choice, but the problem of black-box models limits their ability and capability. This problem can be faced in modern ML/DL where multiple performance criteria are needed. LR and NLR are very good in the **explainability** and **interpretability criteria**, but they lack the accuracy; especially with complicated datasets, and vice versa for ANNs and SVMs. Agencies and authorities need explanations for many AI-based applications and products, especially when human life is at risk. For example, self-driving cars and medical diagnosis where the explanations of their decisions could be legally required. All these facts stimulate us to design a new computing system that is flexible like LR, but strong enough to deal with nonlinear models like NLR, and has the ability to accurately explain the variability of datasets like ANNs and SVMs. A technique that can explain everything by readable mathematical equations, so the preceding performance criteria can all be satisfied. This goal could be depicted in Figure 10.2.

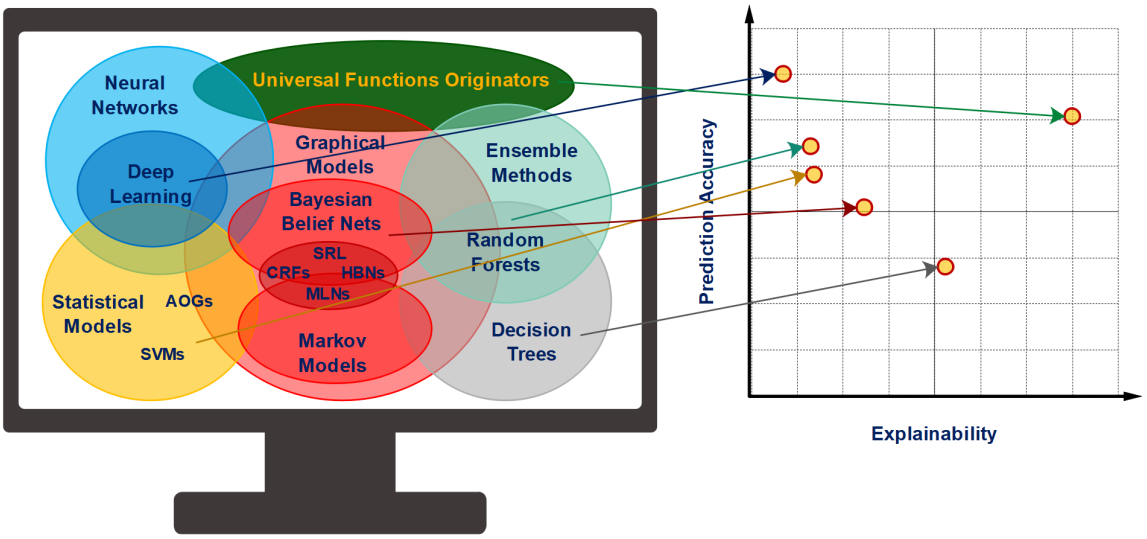


Figure 10.2: UFO Can Achieve Both the Explainability and the Prediction Accuracy

Building such system can help to generate complicated mathematical models that can be utilized in many aspects, like considering them as objective functions in optimization algorithms. Here, in this study, a new computing system called "universal functions originator (UFO)" is introduced.

This novel computing system is designed by merging concepts taken from control systems, fuzzy systems, classical optimization, mixed-integer meta-heuristic optimization, and linear/nonlinear regression analysis; as shown in Figure 10.3. UFO is a multi-purpose computing system that can be used as an alternative to LR, NLR, ANNs, SVMs, and others. In UFO, everything is rendered as mathematical equations. The equation structures are not constant where each term (of arithmetic operators, analytical functions, and coefficients) is subject to dynamic change until finding the optimal mathematical model. Some of the distinct features that can be provided by UFO are [22, 23]:

- It can convert simple functions to unimaginable/unbelievable highly complicated mathematical equations, which might be used for some future applications; similar to **encoding** and **decoding** concepts.

- If applicable, UFO could simplify highly complicated equations to some compact equivalent equations.

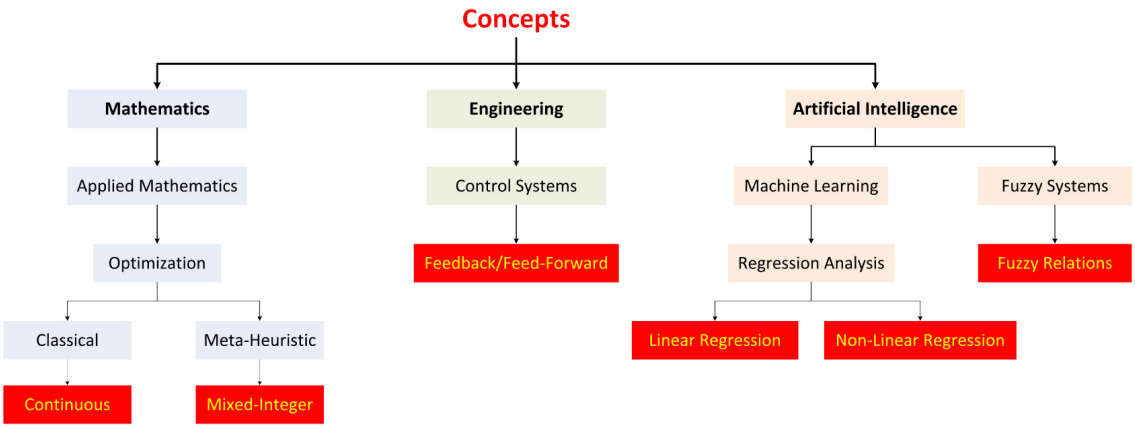


Figure 10.3: UFO is Built by Merging Concepts Taken from Different Disciplines

- It can act as a universal dual linear/nonlinear regression unit.

- It can be used to visualize *high*-dimensional functions.

In general, this computing system could enter many AI fields, such as function approximation, pattern classification, categorization/clustering, anomaly detection, forecasting/prediction, control, optimization, dimension reduction, dimension expansion, function simplification, function complication, and *high*-dimensional function visualization.

This novel computing system shows an impressive performance with many promising uses and distinct capabilities. This chapter reveals the mechanism behind UFO and then some numerical problems are solved via an advanced graphical user interface (GUI) designed just to validate the process of UFO. Also, this chapter shows the main features, capabilities, and strengths of UFO and the main differences with other known computing systems presented in the literature. Furthermore, this chapter highlights many important notes and experimental observations during implementing UFO, which could help other researchers to do further modifications.

10.2.1 Basic UFO Structure for Single-Response Datasets

Suppose that a dataset consists of n predictors $\{x_1, x_2, \cdots, x_n\}$ and one response y. If that response is approximated by a function f, then the approximated response $\hat{y}$

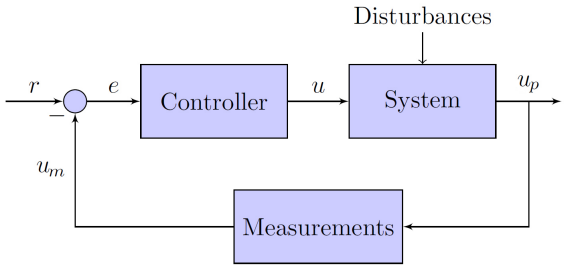


Figure 10.4: Basic Feedback Control Loop

can be mathematically expressed as follows:

$$\hat{y} = f\left(x_1, x_2, \cdots, x_k, \cdots, x_n\right) \quad ; \quad k = 1, 2, \cdots, n \tag{10.26}$$

If a vector notation is used, then (10.26) becomes:

$$\hat{y} = f\left(X\right) \quad ; \quad X = [x_1, x_2, \cdots, x_n] \tag{10.27}$$

Referring to LR and NLR, (10.26) or (10.27) can be solved by LR if there is no nonlinearity between the coefficients and the analytical function f acts as an identity function; i.e., $f = 1$. If there is a nonlinear relation between the coefficients and/or $f \neq 1$, then NLR should be used. The problem with NLR is that everything here is defined manually. The user must define the analytical function f with the initial coefficient values (i.e., the starting point or guess) and the side constraints of each coefficient.

Now, let's borrow the block-diagram concept from the subject of control systems. The basic **feedback control loop** is shown in Figure 10.4. If a unity gain is placed in the measurement block, then $u_m = u_p$. The objective here is to match between the process variable u_p and the set-point r by minimizing the error e through the controller. Suppose that the process variable u_p of the control system is the response $\hat{y}$ estimated by the regression analysis, and the set-point r is the actual response y of the regression analysis. Thus, a similarity can be made between regression analysis and control systems.

It is known that the response y is a function of n predictors. Thus, if the regressed response $\hat{y}$ is decomposed into v functions $\{f_1, f_2, \cdots, f_j, \cdots, f_v\}$, instead of just one function as seen in (10.26)-(10.27), and each jth analytical function has an exponent c_j and the result (i.e., $[f_j(X)]^{c_j}$) is multiplied by an external weight w_j to have a

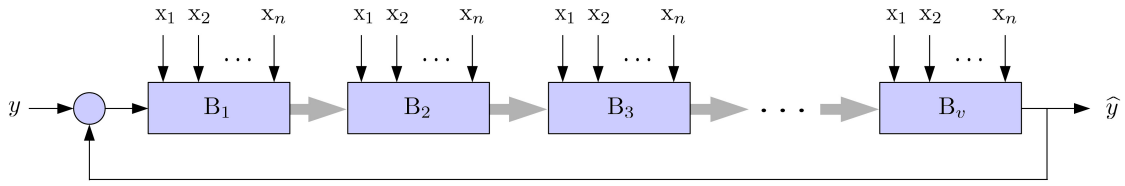


Figure 10.5: Basic Block-Diagram of UFO with One Output Stream

more general function $g_j(X)$, then the whole problem can be illustrated through the block-diagram shown in Figure 10.5. That is, v blocks are needed where each block is occupied by one of the preceding v functions. The thick arrows shown in Figure 10.5 mean that the system has a unidirectional flow of information, which is from the left side to the right side. However, the interaction between every two blocks is unknown. It could be a multiplication; similar to the feedback control loop shown in Figure 10.4. Alternatively, that interaction could also be made by using any standard or user-defined arithmetic operator; including addition, subtraction, and division.

Thus, it will not be adhered to what has been said literally about the mathematical background of the feedback control system. Instead, suppose that the recycle stream of the preceding block-diagram is opened; or in other words, there is no feedback signal. Also, suppose that the error of that open-loop control system is minimized through an external tool with no any delay between the actual response y (which represents the set-point r) and the estimated response $\hat{y}$ (which represents the process variable u_p).

For example, if the v blocks are connected between each other using just the addition operator "+" with considering all the assumptions listed above, then the preceding UFO computing system with one output (O/P) stream shown in Figure 10.5 can be modified to be as that shown in Figure 10.6.

The illustrated UFO block diagram shown in Figure 10.6 represents just one possible model. Thus, by referring to the basic concepts of fuzzy systems and uncertainty, various types of arithmetic operators could be placed between the blocks; instead of using just "×" as in Figure 10.5 or "+" as in Figure 10.6. Also, the other more advanced arithmetic operators (such as modulus, quotient, and exponent) can be used. Furthermore, any other arithmetic operators, including user-defined and fuzzy-based

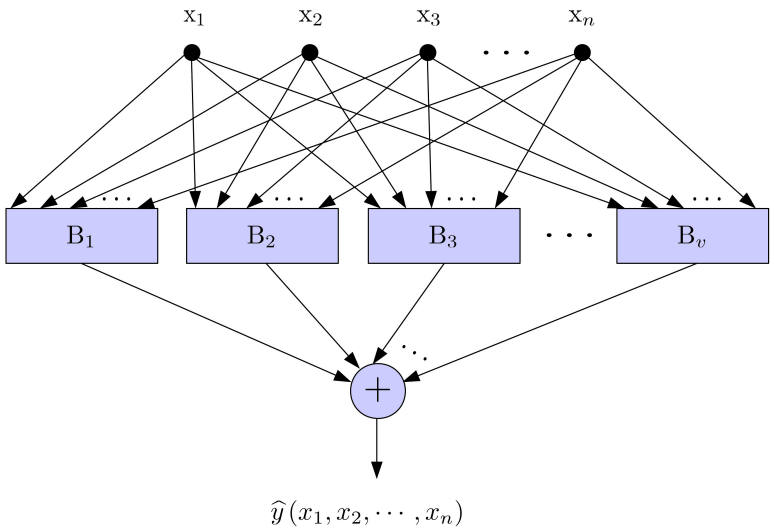


Figure 10.6: Illustrated UFO with Only One O/P Stream and Addition Operators Between Blocks

types, can also be used for more advanced UFO structures. For the sake of simplicity and clarity, let's just focus on the basic four arithmetic operators; i.e., $+$, $-$, $\times$, and $\div$. Once the concept is fully understood, any other operator can be embedded using the same steps.

By considering the preceding assumptions regarding the feedback stream and the system delay, a more general UFO computing system with one O/P stream can be constructed. But, before showing that general-purpose block diagram, let's suppose that each jth block B_j is occupied by a function $g_j(X)$; where $j = 1, 2, \cdots, v$. If all the four basic arithmetic operators are involved during building the overall function $\hat{y}(X)$, then the general-purpose block-diagram of UFO with one O/P stream can be shown in Figure 10.7. The mathematical symbol $\circledcirc$ means a **universal arithmetic operator**. Thus, $\circledcirc$ could be $+$, $-$, $\times$, or $\div$.

After decomposing $\hat{y}(X)$ into v functions $\{g_1, g_2, \cdots, g_j, \cdots, g_v\}$, let's now vary the analytical functions inside them. Thus, each jth analytical function f_j could be any known, unknown, or even user-defined function. For example, f_j could be any analytical function of the following categories:

- **Basic Functions:** $1 \times ()$, $\frac{1}{()}$, $\sqrt{()}$, $|\ |$, $\lfloor\ \rfloor$, $\lceil\ \rceil$, $\lfloor\ \rceil$, $\|\ \|$, $()!$, $()!!$, etc
- **Exponential/Logarithmic Functions:** $\exp()$, $\ln()$, $\log_{10}()$, $\log_2()$, etc

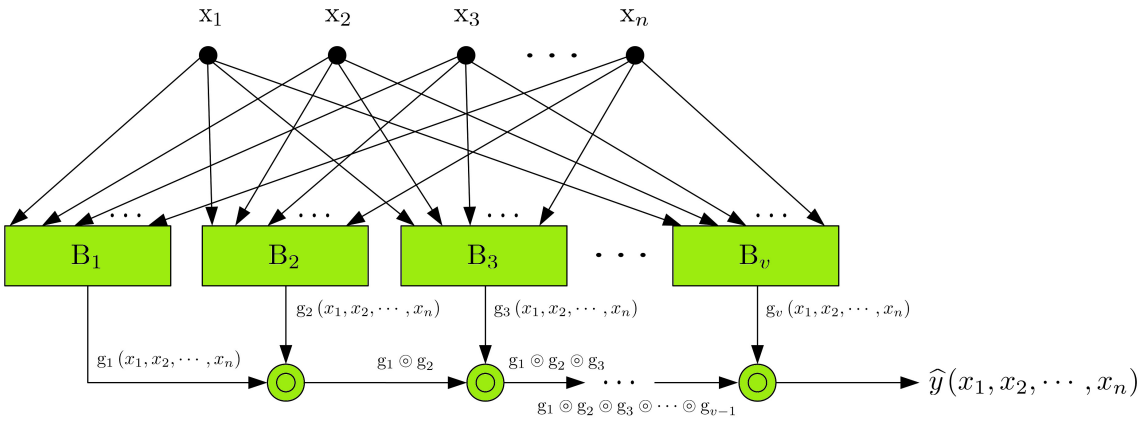


Figure 10.7: Illustrated UFO with Only One O/P Stream and Universal Arithmetic Operators

- **Trigonometric Functions:** sin(), cos(), tan(), csc(), sec(), or cot()
- **Hyperbolic Functions:** sinh(), cosh(), tanh(), csch(), sech(), or coth()
- **Inverse Trigonometric Functions:** $\sin^{-1}()$, $\cos^{-1}()$, $\tan^{-1}()$, $\csc^{-1}()$, $\sec^{-1}()$, or $\cot^{-1}()$
- **Inverse Hyperbolic Functions:** $\sinh^{-1}()$, $\cosh^{-1}()$, $\tanh^{-1}()$, $\operatorname{csch}^{-1}()$, $\operatorname{sech}^{-1}()$, or $\coth^{-1}()$
- **Unfamiliar Functions:** exsec(), excsc(), versin(), vercos(), coversin(), covercos(), sinc(), si(), Si(), Ci(), Cin(), Shi(), Chi(), etc
- **User-Defined Functions**, comprising:
  - **Logistic:**
  $$f_j(x) = \frac{1}{1 + e^{-x}} \tag{10.28}$$
  - **ElliotSig/Softsign:**
  $$f_j(x) = \frac{x}{1 + |x|} \tag{10.29}$$
  - **Inverse Square Root Unit (ISRU):**
  $$f_j(x) = \frac{x}{\sqrt{1 + \alpha x^2}} \tag{10.30}$$
  where α is a coefficient.

– **SoftPlus:**

$$f_j(x) = \ln\left(1 + e^x\right) \tag{10.31}$$

– **Bent Identity:**

$$f_j(x) = \frac{\sqrt{x^2+1}-1}{2} + x \tag{10.32}$$

– **Sigmoid Linear Unit** (**SiLU**):

$$f_j(x) = \frac{x}{1+e^{-x}} \tag{10.33}$$

– **Soft Clipping:**

$$f_j(x) = \frac{1}{\alpha}\log\left(\frac{1+e^{\alpha x}}{1+e^{\alpha(x-1)}}\right) \tag{10.34}$$

– **Gaussian:**

$$f_j(x) = e^{-x^2} \tag{10.35}$$

By referring to what has been said earlier, the relation between g_j and f_j can be mathematically explained as follows:

$$g_j\left(X\right) = w_j \cdot \left[f_j\left(a_{0,j} \odot_{1,j} a_{1,j} \cdot x_1^{b_{1,j}} \odot_{2,j} a_{2,j} \cdot x_2^{b_{2,j}} \odot_{3,j} \cdots \odot_{n,j} a_{n,j} \cdot x_n^{b_{n,j}}\right)\right]^{c_j} \tag{10.36}$$

which can also be expressed using the following notation:

$$\begin{cases} \bar{x}_{i,j} = a_{i,j} x_i^{b_{i,j}} \\ u_{0,j} = a_{0,j} \\ u_{i,j} = \odot_{i,j}(u_{i-1,j}, \bar{x}_{i,j}) \quad 1 \le i \le n \\ g_j(X) = w_j(f_j(u_{n,j}))^{c_j} \end{cases} \tag{10.37}$$

where

- $\odot_{k,j}$: the kth arithmetic operator assigned to the kth predictor of the jth block B_j; it could be $+$, $-$, $\times$, or $\div$
- f_j : the analytical function assigned to the jth block B_j; it could be $1\times()$, $1/()$, $\sin()$, $\cos()$, $\cot^{-1}()$, $\cosh()$, $\mathrm{csch}^{-1}()$, $\exp()$, $\ln()$, $\log_2()$, $\log_{10}()$, etc
- w_j : the weight assigned to the jth block B_j; where $w_j \in [w_j^{\min}, w_j^{\max}]$
- $a_{0,j}$: the intercept of the jth block B_j; where $a_{0,j} \in [a_{0,j}^{\min}, a_{0,j}^{\max}]$

- $a_{k,j}$: the kth weight assigned to the kth predictor located in the jth block B_j; where $a_{k,j} \in [a_{k,j}^{\min}, a_{k,j}^{\max}]$

- $b_{k,j}$: the kth exponent assigned to the kth predictor located in the jth block B_j; where $b_{k,j} \in [b_{k,j}^{\min}, b_{k,j}^{\max}]$

- c_j : the exponent assigned to the jth analytical function f_j located in the jth block B_j; where $c_j \in [c_j^{\min}, c_j^{\max}]$

As can be clearly seen from (10.36), the mathematical symbol $\odot$ has the same meaning of $\circledcirc$; i.e., acting as a universal arithmetic operator. The symbol $\odot$ is used internally between the $(n+1)$ a-coefficients of each jth analytical function, while the symbol $\circledcirc$ is used externally between the v block functions $\{g_1, g_2, \cdots, g_j, \cdots, g_v\}$. Thus, in UFO, $\odot$ and $\circledcirc$ are respectively defined as the **internal** and **external universal arithmetic operators**.

It has to be said that (10.36) can be replaced with other more complicated forms, like embedding an internal analytical function in the place of the intercept[2] $a_{0,j}$, **internal weight** $a_{k,j}$, **internal exponent** $b_{k,j}$, **external exponent** c_j, or/and **external weight** w_j. But right now, let's just use (10.36) to explain how UFO works. Once the core concept of UFO is understood, then any mathematical form can be used for g_j.

In (10.27), it has been seen that the regressed response $\hat{y}$ is decomposed into v functions $\{f_1, f_2, \cdots, f_v\}$. Also, each f_j is further expanded by an exponent c_j and a weight w_j to form a more general function g_j, which is mathematically expressed in (10.36). Thus, by referring to Figure 10.7, the response $\hat{y}(X)$ estimated by UFO can be mathematically expressed as follows:

$$\hat{y}(X) = g_1(X) \circledcirc_1 g_2(X) \circledcirc_2 \cdots \circledcirc_{v-1} g_v(X) \tag{10.38}$$

As said before, the symbols $\circledcirc$ and $\odot$ are identical. The optimal sets of $\circledcirc$ and $\odot$ can be obtained by finding the best possible mathematical model. This means that all the model variables $\{a_0, a, b, c, w, f, \odot, \circledcirc\}$ need to be optimized.

After borrowing some concepts from control systems, fuzzy systems, and LR/NLR analysis, let's also refer to the optimization field. To make UFO work, two different

[2]It can be called either the **block bias** or the **local constant term** of the jth block.

branches of the optimization field are needed. The first branch deals with global stochastic/probabilistic-based meta-heuristic optimization algorithms, and the second branch deals with local gradient-based classical optimization algorithms. To be more specific, UFO depends on a sub-branch of meta-heuristic optimization algorithms that deals with mixed-integer optimization problems, while the local classical optimization algorithm used in UFO could be a continuous or a mixed-integer type based on the settings defined by the user. It has to be said that these two optimization techniques are separated from each other, and each one occupies one main stage of UFO[3].

The objective here is to construct multiple models by varying $\{a_0, a, b, c, w, f, \odot, \circledcirc\}$, and then enhancing the performance of some selected models by fixing $\{f, \odot, \circledcirc\}$ and fine-tuning $\{a_0, a, b, c, w\}$. That is, to solve this highly non-convex nonlinear mixed-integer optimization problem, both meta-heuristic and classical optimization algorithms are required. The first one acts as a **global mixed-integer optimization algorithm** to compose all $g_j(X)$ functions so that the overall mathematical function $\hat{y}(X)$ with its initial parameters can be generated. Then, the **local gradient-based optimization algorithm** is used to tune the parameters $\{a_0, a, b, c, w\}$. Therefore, the total number of optimization variables depends on the number of blocks v and the number of predictors n involved in the problem. For the global mixed-integer optimization stage, its dimension $\aleph$ can be computed by using the following formula:

$$\aleph = 3vn + 5v - 1 \tag{10.39}$$

This formula can be obtained by referring to (10.36) and (10.38). There are "v" variables of type $\{w, f, a_0, c\}$, "$v \times n$" variables of type $\{\odot, a_k, b_k\}$, and "$v - 1$" variables of type $\circledcirc$. Thus, by adding them together, (10.39) can be constructed as follows:

$$\begin{aligned} \aleph &= (v \times 4) + (v \times n \times 3) + (v - 1) \\ &= 4v + v - 1 + 3vn \\ &= 3vn + 5v - 1 \end{aligned}$$

For the local gradient-based optimization stage, all the variables of type $\{f, \odot, \circledcirc\}$ are vanished. Thus, its dimension $\mathfrak{D}$ can be computed by using the following formula:

$$\mathfrak{D} = 2vn + 3v \tag{10.40}$$

[3] i.e., they are not hybridized.

This formula can be obtained by subtracting all the variables of type $\{f, \odot, \circledcirc\}$ from (10.39) as follows:

$$\begin{aligned}\mathfrak{D} &= (3vn + 5v - 1) - (v) - (vn) - (v - 1) \\ &= 3vn + 5v - 1 - v - vn - v + 1 \\ &= vn(3 - 1) + v(5 - 1 - 1) - 1 + 1 \\ &= 2vn + 3v\end{aligned}$$

To clarify (10.36)-(10.40), let's suppose $v = 2$, $n = 4$, $f = [\log_{10}, \tanh^{-1}]$, $\odot_1 = [+, -, +, /]$, $\odot_2 = [/, +, -, +]$, and $\circledcirc = \times$. Using (10.38), the actual response y can be approximated as follows:

$$\hat{y}(X) = g_1(X) \times g_2(X) \tag{10.41}$$

where the block functions $g_1(X)$ and $g_2(X)$ can be mathematically expressed, using (10.36), as follows:

$$g_1(X) = w_1 \left[\log_{10} \left(a_{0,1} + a_{1,1} \cdot x_1^{b_{1,1}} - a_{2,1} \cdot x_2^{b_{2,1}} + \frac{a_{3,1} \cdot x_3^{b_{3,1}}}{a_{4,1} \cdot x_4^{b_{4,1}}} \right) \right]^{c_1} \tag{10.42}$$

$$g_2(X) = w_2 \left[\tanh^{-1} \left(\frac{a_{0,2}}{a_{1,2} \cdot x_1^{b_{1,2}}} + a_{2,2} \cdot x_2^{b_{2,2}} - a_{3,2} \cdot x_3^{b_{3,2}} + a_{4,2} \cdot x_4^{b_{4,2}} \right) \right]^{c_2} \tag{10.43}$$

Observe that $a_{0,2}$ does not behave as an intercept anymore. Instead, it acts as a weight to the term $a_{1,2}^{-1} \cdot x_1^{-b_{1,2}}$. Thus, the first internal universal arithmetic operator $\odot_{1,j}$ will decide the type of $a_{0,j}$. It is treated as an intercept if $\odot_{1,j} = \{+, -\}$. Otherwise, $a_{0,j}$ is treated as an internal weight similar to $a_{k,j}$.

The problem dimension $\aleph$ of the building stage can be calculated via (10.39) as follows:

$$\aleph = 3vn + 5v - 1 = 3(2)(4) + 5(2) - 1 = 33 \tag{10.44}$$

which can be solved by using any global mixed-integer optimization algorithm, or even through a `for`-loop with a random solution generator to reduce the CPU time. That is, both probabilistic and stochastic types can be implemented here.

Also, the problem dimension $\mathfrak{D}$ of the tuning stage can be determined by applying (10.40) as follows:

$$\mathfrak{D} = 2vn + 3v = 2(2)(4) + 3(2) = 22 \tag{10.45}$$

and this problem can be solved by using any local gradient-based optimization algorithm.

10.2.1.1 Overall Mechanism of UFO

By collecting all the steps explained above, the following stages are required to build any basic UFO:

1. **Pooling Stage:**
   In this stage, various types of analytical functions and arithmetic operators are selected to enter the pool. Thus, the pool size, quality, and complexity depend on the number and type of analytical functions and arithmetic operators selected for $\{f, \odot, \circledcirc\}$ before starting UFO.

2. **Building Stage:**
   This stage can be created by using any global mixed-integer optimization algorithm. UFO can generate functions by (10.36) and then substituting them in (10.38) to get an overall function with its initial coefficients. These two equations should be used with UFO containing only one O/P stream. Thus, two other equations should be used with UFO containing multiple O/P streams; as will be seen later in the next subsection. Thus, there are different approaches to generate the required mathematical model. The options available to the user can be summarized as follows:

   - Selecting all the analytical functions available in the pooling stage, so UFO can find the proper one every time it is executed.
   - Selecting some specific analytical functions to limit the final mathematical model between some specific analytical functions. For example, if only $\{\exp, \sin, \tan, \ln\}$ are selected, then the models generated by UFO will contain all or some of these analytical functions.
   - Selecting only one analytical function. This means that the user wants to get a mathematical model based on a specific analytical function type.

   That is, if multiple analytical functions are selected during the initialization phase, then the improper analytical functions will be rejected automatically

from the building stage during searching for the best mathematical model.

In this stage, even if the variables $\{a_0, a, b, c, w\}$ are kept continuous, a mixed-integer optimization algorithm is required because it is allowed to select only some specific types of analytical functions. Also, the internal and external arithmetic operators are sets of some limited elements; which are any combination of $\{+, -, \times, \div\}$. A global stochastic or probabilistic meta-heuristic optimization algorithm is preferred for this stage to ensure covering the entire search space and checking many mathematical models as much as possible.

To avoid complicating UFO design and to accelerate the processing speed, the meta-heuristic mixed-integer optimization algorithm used in this stage can be replaced by just a few lines of code to generate a random population per each iteration; or even just one candidate solution per each iteration. This might help to increase the exploration level as well because the most pressure is applied to the tuning stage to exploit the search space.

3. **Tuning Stage:**

The functions generated in the building stage are not well fitted, because the optimization algorithm used in that stage is supposed to be either a stochastic or a probabilistic meta-heuristic type not a gradient-based type. Thus, the mission of the tuning stage is to fit the mathematical models generated in the last stage by using any tool, such as the **trust-region reflective** (**TRR**) or Levenberg-Marquardt (LM) algorithm. Here, UFO can be programmed to tune the entire population created in the building stage, or it can be just the elite candidate solutions or any user-defined number of solutions required to be tuned.

As said before, this stage is responsible to fine-tune just $\{a_0, a, b, c, w\}$. The remaining three discrete variables, i.e. $\{f, \odot, \circledcirc\}$, are kept as they were in the building stage. This is why $\aleph > \mathfrak{D}$, always, for any positive integer number of v. This can be proved by equating (10.39) with (10.40) to have the following relation:

$$n = \frac{1 - 2v}{v} \tag{10.46}$$

It is obvious that the number of predictors n becomes negative if $v \geqslant 1$, which is impossible.

4. **Evaluation Stage:**

   This is an optional stage that can be activated to further evaluate the models generated in the preceding two stages, so a strong judgment can be built upon. Similar to ANNs and other ML computing systems, this stage divides the dataset into three parts. The biggest part is used to build the overall function $\hat{y}(X)$, and the remaining two parts are used to test and validate the results. Although it is an optional stage, activating it is very important to acquire stable results and avoid the overfitting phenomenon.

10.2.1.2 Objective Function

As with other ML computing systems, the performance of UFO can be measured by looking into the difference between the actual response y and the approximated response $\hat{y}$. There are many ways to measure that difference. Each one of them can be considered as an objective function when it is used in UFO. Such metrics are coefficient of correlation (R), coefficient of determination (R^2), mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), error sum of squares (SSE), etc. Of course, each one of these methods has its pros and cons.

Before jumping to the design constraints, it has to be mentioned that UFO can be programmed to solve multi-objective problems by considering different performance criteria at the same time. For example, minimizing MSE and maximizing R^2. Furthermore, the CPU time can be utilized to measure the complexity of the mathematical function generated by UFO. Thus, the user could prefer to have $\hat{y}$ where both MSE and CPU time performance criteria are satisfied. These are just some possible upgrades to the prospective future versions of UFO.

10.2.1.3 Design Constraints

It is known that the function $\hat{y}(X)$ approximated by UFO using a finite number of observations. During building and tuning that function, UFO should check all the elements of $\hat{y}(X)$. Thus, the function generated by UFO is considered feasible, viable and applicable if the following criteria are satisfied:

- all the elements of $\hat{y}$ should not be complex (i.e., $\Re\{\hat{y}\} \neq \hat{y}$ and $\Im\{\hat{y}\} \neq 0$),

- all the elements of $\hat{y}$ should not equal infinity (i.e., $\pm\infty$), and
- all the elements of $\hat{y}$ should not be undefined (i.e., $0/0, 0\times\infty, \infty/\infty, \infty-\infty$).

The first constraint could be relaxed in case the function approximated by UFO is wanted in a complex form.

Nowadays, there are many constraint-handling techniques presented in the literature, which are previously listed in Figure 2.6. Based on many numerical experiments conducted on UFO, it has been found that the basic random search method consumes a significant CPU time compared with some death and non-death exterior penalty functions (EPFs). However, the former method could beat EPFs in terms of solution quality and convergence rate. A further discussion about this point will be covered later.

The other highly important point is about the possibility to expedite detecting feasible solutions and enhancing the convergence rate. This can be done by embedding a pre-processing sub-algorithm to have some controls on the variables generated in the building stage and, in a slighter way, in the tuning stage. The goal here is to give some guidance to these two stages to avoid some values that might violate one or more of the above three constraint criteria. To clarify this vague point, suppose $v=3$ and $\circledcirc\in\{+,-,\times,\div\}$. Applying (10.38) leads to the following:

$$\hat{y}\,(X)=g_1(X)\circledcirc_1 g_2(X)\circledcirc_2 g_3(X) \tag{10.47}$$

If $g_1(X)\neq 0$ and $g_2(X)=0$, then that pre-processing sub-algorithm should ensure $\circledcirc_1\neq\div$. For that case, $\circledcirc_1=\times$ is accepted only if $\circledcirc_2\neq\div$ when $g_3(X)=0$. The same thing happens when $g_1(X)=0$, $g_2(X)\neq 0$, and $\circledcirc_1=\times$. Here, $\circledcirc_2$ should not be $\div$ when $g_3(X)=0$. To overcome this issue, a rule can be applied so if $g_j(X)=0$ then $\circledcirc_{j-1},\circledcirc_j\in\{+,-\}$.

Internally, between predictors, the same strategy could be applied to enhance the UFO performance. By referring to (10.36), the kth predictor x_k in the jth block can be deactivated by setting its weight $a_{k,j}$ to zero, and the internal universal arithmetic operators $\odot_{k,j},\odot_{k+1,j}\in\{+,-\}$. The same thing can be used for the intercept $a_{0,j}$, which is accepted to be zero if $\odot_{1,j}\in\{+,-\}$. That is, the multiplication and division operators require some precautions if someone wants to involve them. UFO can operate without considering any of these enhancements by just rejecting or penalizing

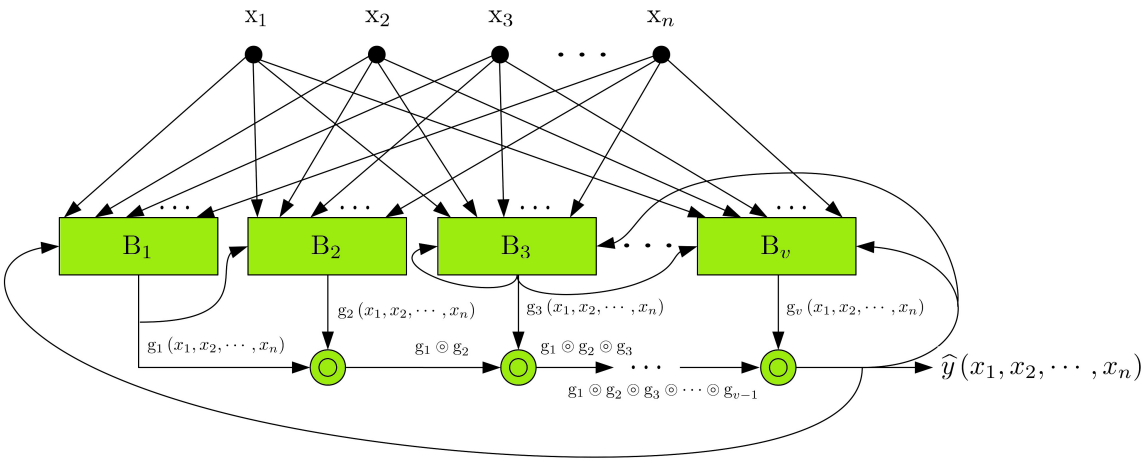


Figure 10.8: Illustrated Recurrent UFO (or RUFO) with Only One O/P Stream and Universal Arithmetic Operators

infeasible functions. But, this strategy requires additional time to explore the search space. The other possible option is by deactivating some arithmetic operators in the search pool.

Furthermore, to have advanced UFO structures, then dynamic variable bounds could be used based on each function type selected for the jth block. For example, if $f_j = \exp$, then all the elements inside that exponential function should lie in a reasonable domain; such as [-10,10], which gives a range of [4.5400E-05,2.2026E+04].

These are some useful points that could be partially or fully involved to have robust UFO structures.

10.2.1.4 Recurrent UFO

It has been seen that the feedback loop of Figure 10.5 was open during constructing UFO with one O/P stream. It is important to say that more complicated UFO computing systems can be built by considering some recurrent streams between blocks. Such systems can be called **recurrent UFO** (or just **RUFO**). For example, Figure 10.8 shows one possible RUFO structure where the output ports of some blocks are recycled to other blocks.

There are some prospective future uses of RUFO, especially in forecasting and other applications where a part of the past data needs to be memorized and then utilized again. However, such computing systems are much more difficult than normal

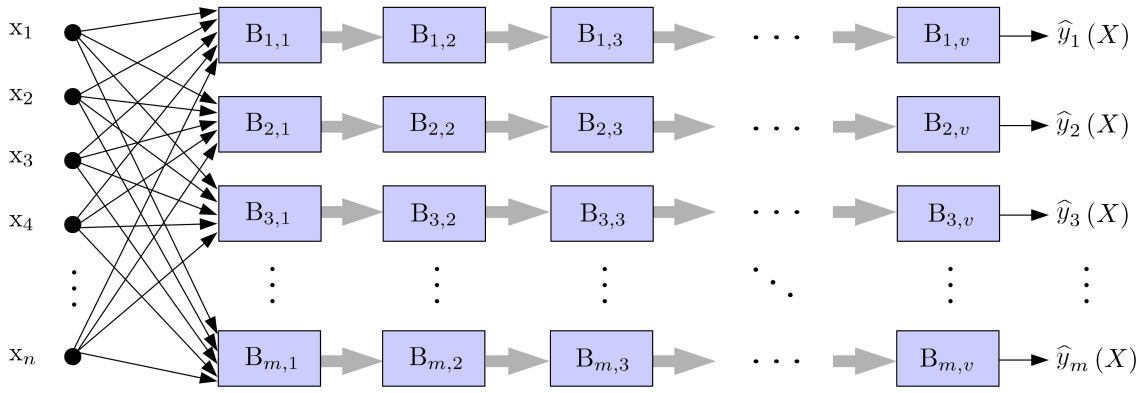


Figure 10.9: Basic Block-Diagram of UFO with m O/P Streams

designs, and their optimization dimensions are higher than those calculated by (10.39) and (10.40).

10.2.2 Basic UFO Structure for Multiple-Response Datasets

After understanding the mechanism of UFO with one O/P stream, let's now suppose that there are m responses $\{y_1, y_2, \cdots, y_m\}$ that need to be predicted. To do that, different approaches can be applied here. One of the simplest approaches is to use m structures of Figure 10.5 and then letting them work in parallel. By considering the same preceding assumptions for the feedback stream and the system delay, a basic UFO with m O/P streams can be built. Such structure is graphically explained in Figure 10.9.

As can be clearly seen from Figure 10.9, there is no interconnection between any jth block of the ith row to other blocks located in other rows. Also, the information has a unidirectional flow, which is from the left side to the right side of each block. This means that there is no recycle or feedback connection between the blocks located in the same row as previously seen in Figure 10.8. This property is hard to be achieved in neural networks where the neurons are highly interconnected with each other. In UFO, the rows can be isolated or partially/fully interconnected between each other.

For the sake of simplicity, the basic block-diagram shown in Figure 10.9 is adopted in explaining the mechanism of UFO with multiple outputs. Once the concept is understood, the other more advanced structures can be built. Now, because there are m rows, let's denote each block as $B_{i,j}$ where the subscript $i = 1, 2, \cdots, m$ and the

subscript $j = 1, 2, \cdots, v$. Thus, g_j and f_j will respectively become $g_{i,j}$ and $f_{i,j}$, and the overall expression $\hat{y}\,(X)$ will become $\hat{y}_i\,(X)$ where the subscript i represents the desired row. Based on this, (10.36) must be replaced by the following expression:

$$g_{i,j}\,(X) = w_{i,j} \cdot \left[f_{i,j} \left(a_{0,i,j} \odot_{1,i,j} a_{1,i,j} \cdot x_1^{b_{1,i,j}} \odot_{2,i,j} \cdots \odot_{n,i,j} a_{n,i,j} \cdot x_n^{b_{n,i,j}} \right) \right]^{c_{i,j}} \tag{10.48}$$

and by adopting the format used in (10.37), the above equation can also be expressed as follows:

$$\begin{cases} \bar{x}_{k,i,j} = a_{k,i,j} x_k^{b_{k,i,j}} \\ u_{0,i,j} = a_{0,i,j} \\ u_{k,i,j} = \odot_{k,i,j}(u_{k-1,i,j}, \bar{x}_{k,i,j}) \quad ; \quad 1 \leq k \leq n \\ g_{i,j}(X) = w_{i,j}(f_{i,j}(u_{n,i,j}))^{c_{i,j}} \end{cases} \tag{10.49}$$

Also, to express the predicted variable of each ith stream, (10.38) must be replaced by the following expression:

$$\hat{y}_i\,(X) = g_{i,1}(X) \circledcirc_{i,1} g_{i,2}(X) \circledcirc_{i,2} \cdots \circledcirc_{i,v-1} g_{i,v}(X) \tag{10.50}$$

Similar to what have been done in (10.36)-(10.38), by considering ith rows with internal and external universal arithmetic operators $\{\odot, \circledcirc\}$, the practical version of the new UFO computing system with m O/P streams can be graphically explained through the block-diagram shown in Figure 10.10.

The optimization problem dimensions of this block-diagram can be calculated by just multiplying (10.39) and (10.40) by m rows to have the following two formulas:

$$\aleph = 3mvn + 5mv - m \tag{10.51}$$

$$\mathfrak{D} = 2mvn + 3mv \tag{10.52}$$

where $\aleph$ stands for the dimension of the global mixed-integer optimization algorithm used in the building stage, and $\mathfrak{D}$ stands for the dimension of the local gradient-based optimization algorithm used in the tuning stage.

10.2.3 Graphical User Interface

To validate the operation of UFO, an advanced GUI is introduced. This GUI is built based on the structure given in Figure 10.7 to run UFO with a single output stream

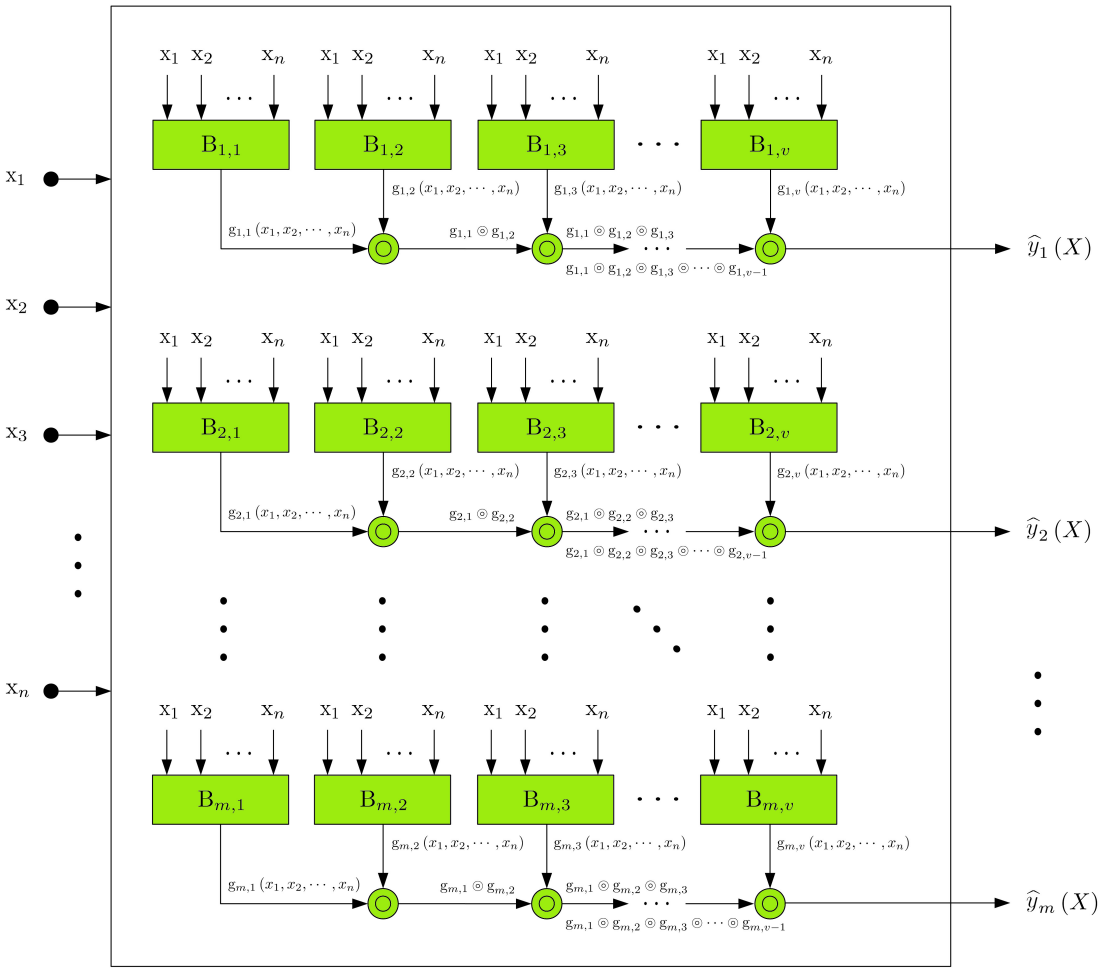


Figure 10.10: Illustrated UFO with Multiple O/P Streams and Universal Arithmetic Operators

for any value of n and v. This means that the mathematical expressions required to run this GUI are (10.36)-(10.40).

The main window of the GUI is shown in Figure 10.11. As can be clearly seen from that snapshot, there are three other tabs available in the program. The second and third tabs are designed to adjust all the parameters used in the building and tuning stages. The last tab is responsible to display all the results numerically in a real-time manner. These three tabs are respectively shown in Figures 10.12-10.14.

The appearance of the GUI shows that the maximum number of v is restricted up to 9. This is just an internal tying in the GUI, which can be modified to any other number. The reason is, based on different experiments, the convergence of many

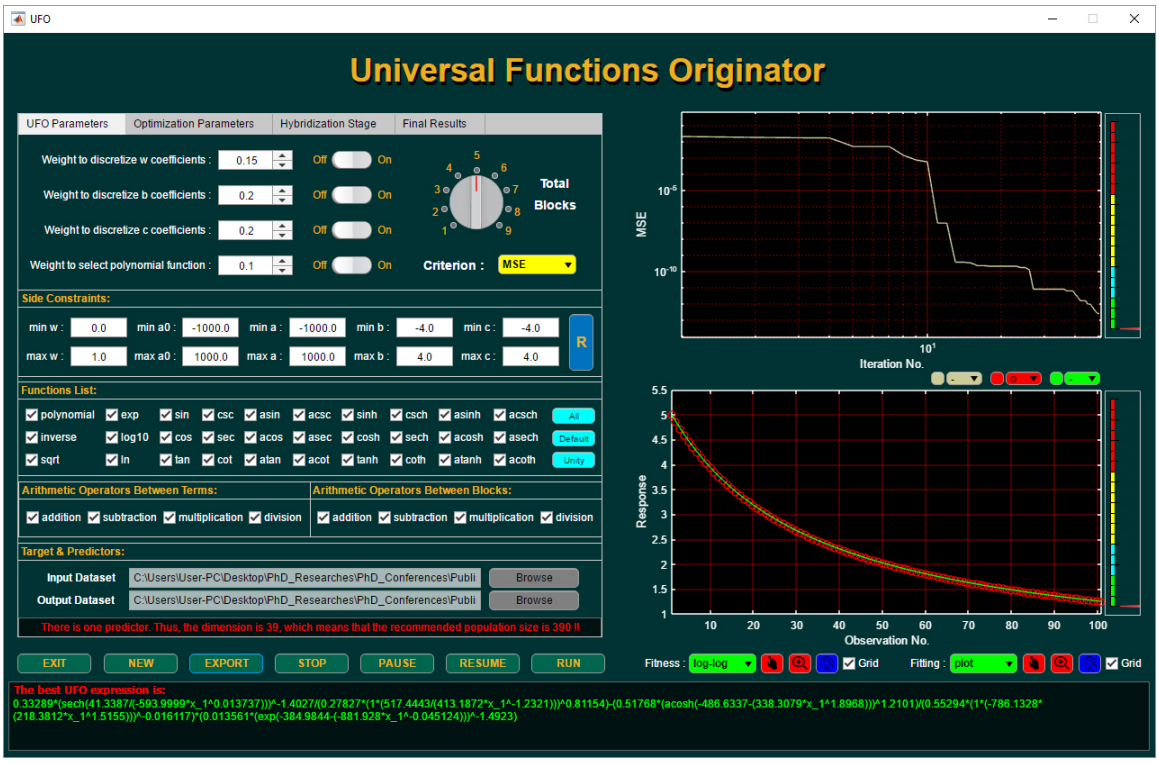


Figure 10.11: The First Fully Functioning GUI Used to Validate the Operation of UFO and Evaluate Its Performance

problems can be achieved using just $v = 1$, 2, or 3.

In this GUI, the building stage is designed with two popular meta-heuristic optimization algorithms that are modified to solve mixed-integer optimization problems. The first one is the differential evolution (DE) algorithm, and the second one is the biogeography-based optimization (BBO) algorithm. More detailed information about these two meta-heuristic optimization algorithms is covered in Chapter 2.

The selection between them can be done by a selector switch. A possible hybridization can be done between any one of these two optimization algorithms with the opposition-based learning (OBL) algorithm, so the other sides of the search space can also be explored within the same iteration. A full explanation of the OBL algorithm can be found in [345, 362]. For the tuning stage, two possible local gradient-based optimization algorithms can also be chosen through a selector switch when the variable bounds are dynamic. The first one is the TRR algorithm, which is explained in [264, 353]. The other one is the LM algorithm, which is explained in [263]. If the

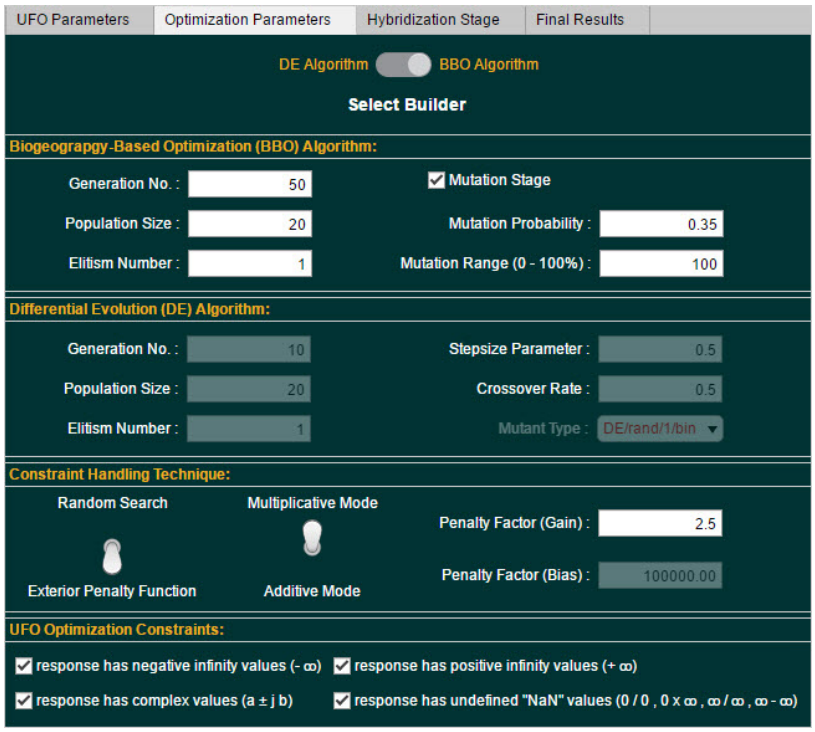


Figure 10.12: The Second GUI Tab. It is Used to Adjust the Building Stage and Its Associated Design Constraints

variable bounds are kept fixed, then there is no other option except to use TRR. To satisfy the design constraints, explained earlier, a selector switch is used to choose either a random search or an EPF method to deal with these constraints. The pseudocode given in Algorithm 1 describes the mechanism of the basic random search method used in the GUI. If the EPF method is chosen, then there are two other options; either using the additive or the multiplicative mode. These two modes are respectively explained (2.20) and (2.21). Thus, the value of the penalized function $\phi(X)$ becomes higher than its actual value in the error function $E(X)$ if, at least, one design constraint is violated. If that individual is feasible, i.e. all the design constraints are satisfied, then $\phi(X) = E(X)$ where $P(X) = 0$.

In addition, there are other features and options available in that GUI. One of these features is that the GUI can automatically detect and display the number of predictors n given in the data. Thus, based on the value of n and the number of blocks v set by the user, the GUI calculates the problem dimension $\aleph$ of the building stage, and then it displays that value just below the data directory path shown in

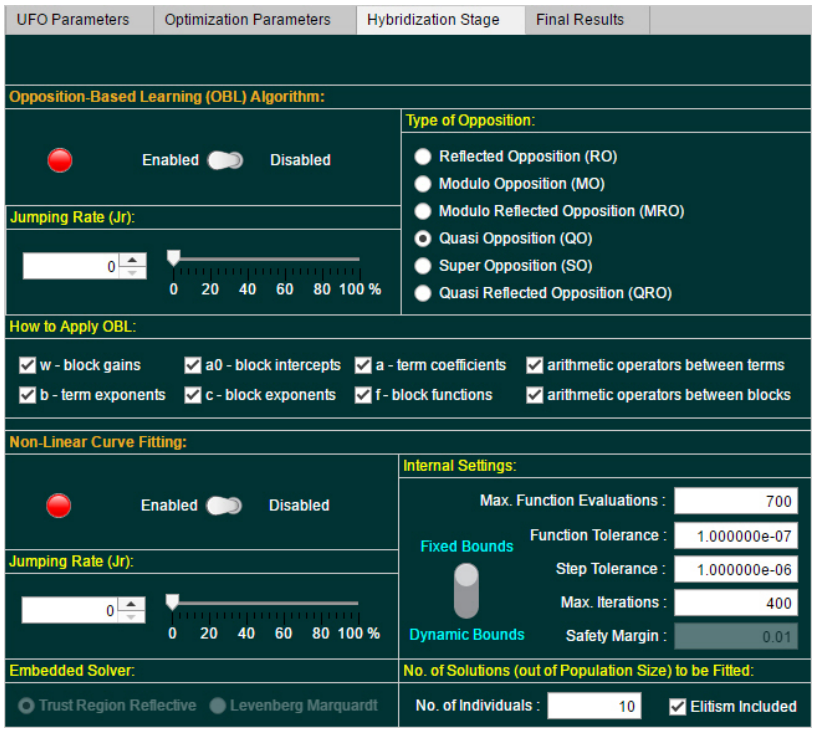


Figure 10.13: The Third GUI Tab. It is Used to Enhance the Building Stage and Also to Adjust the Tuning Stage

the first tab. After that, it recommends the suitable population size $\mathfrak{p}$ to the building stage, which is determined based on a general rule that $\mathfrak{p}$ must equal ten times the problem dimension $\aleph$, as follows:

$$\mathfrak{p} = 10 \times \aleph \tag{10.53}$$

However, (10.53) is not binding advice, because the optimal $\mathfrak{p}$ depends on many factors including the nature of the dataset itself. Also, determining the optimal $\mathfrak{p}$ might require to do some sensitivity analysis and statistical tests [31, 60, 208, 288]. In the literature, there are many smart options, such as the studies reported in [121, 122, 269]. This means that a new rich area of research can be seen here for this part of UFO.

The other useful feature is the online plotting capability for both the fitness and the fitted curves. This feature comes with many options to adjust the appearance of each plot. Also, UFO automatically displays every new optimal mathematical expression generated by the building or tuning stage in the text area located at the

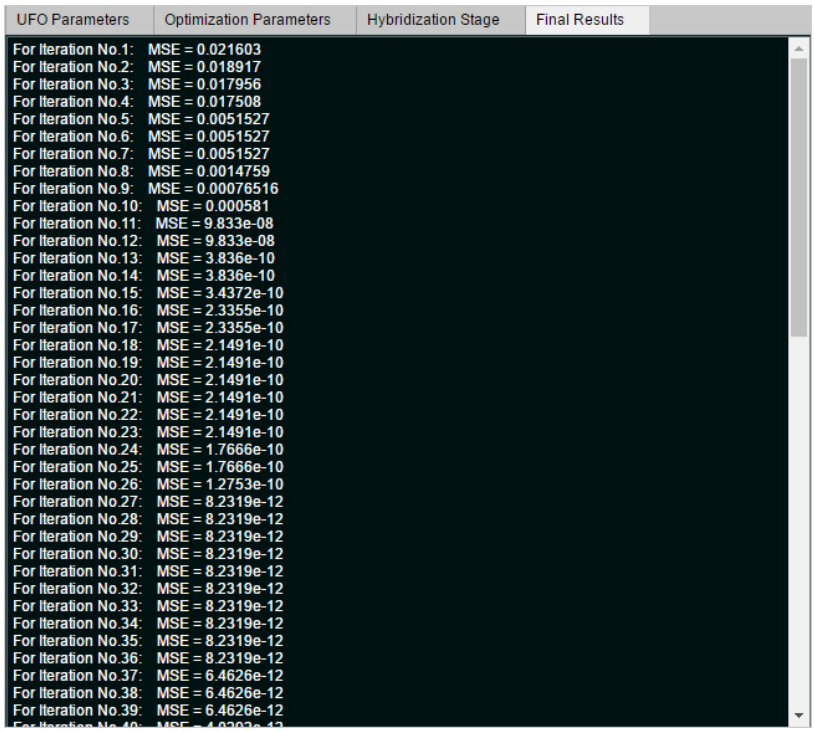


Figure 10.14: The Fourth GUI Tab. It is Used to Display All the Numerical Results Generated by UFO

bottom of the program. This box allows the user to copy and paste that expression directly into MS Word, and then it can be converted to a beautiful math format by just pressing on ALT and = buttons; after selecting all the characters by pressing on CTRL and A. Furthermore, the entire process can be paused, resumed, stopped, or even reset, and all the results can be exported as an MS Excel file classified in several sheets.

10.2.4 Testing UFO with Some Simple Regression Problems

To evaluate the UFO performance, the GUI shown in Figures 10.11-10.14 is used to approximate functions. For that purpose, two numerical experiments are conducted as follows:

Table 10.1: Approximating (10.54) Using the ANN Frameworks of Google's TensorFlow and Mathworks' MATLAB [54]

Framework	**MSE**	**Training Algorithm**	**Iterations**
TensorFlow	$\sim 2 \times 10^{-3}$	Gradient Descent	4,001
MATLAB	$\sim 10^{-10}$	Levenberg-Marquardt	4,001
MATLAB	$\sim \mathbf{10^{-12}}$	Bayesian Regularization	4,001
TensorFlow	$\sim 1.2 \times 10^{-4}$	Gradient Descent	50,000
TensorFlow	$\sim 3 \times 10^{-5}$	AdamOptimizer	4,000
TensorFlow	$\sim 1.5 \times 10^{-5}$	AdamOptimizer[a]	4,000
TensorFlow	$\sim 3 \times 10^{-8}$	AdamOptimizer	40,000

[a]Using an exponential decay learning rate.

10.2.4.1 Dataset with One Predictor

In this experiment, a simple *one*-dimensional non-noisy function is used as a regression problem. The actual mathematical expression of this function is:

$$y = \frac{1}{x} \quad ; \quad x \in [0.2, 0.8] \tag{10.54}$$

A step-size of 0.006 is used to have a linearly spaced vector consisted of 100 observations. The problem is given in [54] as part of a discussion about two popular ANN frameworks; Google's TensorFlow and Mathworks' MATLAB. By using one hidden layer of 10 sigmoid neurons, the results of these two ANN frameworks are tabulated in Table 10.1.

Also, through the statistics and machine learning toolbox of MATLAB R2017b, the preceding function given in (10.54) is approximated via SVM using the actual and standardized data. Table 10.2 shows the results obtained by SVM using 9 different configurations of its kernel function and internal solver; for both data types.

Furthermore, classical LR analysis is used in this experiment. The results obtained by MATLAB are listed in Table 10.3. It is obvious that the MSE performance increases proportionally with the polynomial order. The analysis is stopped with the nonic polynomial model because MATLAB does not support orders greater than 9.

Table 10.2: Approximating (10.54) Using Different SVMs

Configuration[a]		Conventional SVM			Standardized SVM		
Kernel	**Solver**	**Iter.**	**Vectors**	**MSE**	**Iter.**	**Vectors**	**MSE**
Linear	SMO	89	72	0.1904	164	79	0.1388
	ISDA	134	81	0.7996	284	68	0.1736
	L1QP	10	72	0.1904	10	78	0.1440
Polynomial	SMO	104	64	0.1167	59	6	0.0043
	ISDA	854	66	0.1752	869	10	**0.0038**
	L1QP	10	64	0.1172	11	4	0.0043
Gaussian	SMO	59	47	**0.0919**	44	12	0.0081
	ISDA	404	61	0.1433	239	21	0.0162
	L1QP	10	47	0.0920	10	12	0.0081

[a]SMO: sequential minimal optimization, ISDA: iterative single data algorithm, and L1QP: $L1$ soft-margin minimization by quadratic programming.

Table 10.3: Approximating (10.54) Using Different LRs

Order	**MSE**	**Order**	**MSE**	**Order**	**MSE**
1	0.1173	4	1.8825E-4	7	2.8671E-7
2	0.0139	5	2.1735E-5	8	3.2778E-8
3	0.0016	6	2.5003E-6	9	**3.7361E-9**

Now, let's use UFO with the same initialization parameters shown in Figures 10.11-10.14; except for the total number of blocks, which is set to $v = 1$. It has been found that for every run the UFO computing system can provide a very precise solution within just a few iterations. For example, Table 10.4 gives five possible function approximations with their MSE values, which are obtained by UFO for (10.54). Although the maximum iteration number is set to 50, some solutions can be obtained after just a few iterations. It is obvious that UFO scores the best results compared with ANNs, SVMs, and LRs. These results show the UFO capability to find approximated functions by many analytical functions automatically without any external help. Trying to do that with NLR is a very tiring process where each function needs to be manually defined by the user himself/herself plus a wise selection of the starting point and the side constraints of the model coefficients.

It has to be said that the same regression analysis tabulated in Table 10.3 is carried out in Python with the scikit-learn library using very high order polynomial models

Table 10.4: Approximating (10.54) Using UFO with One Block

Run #	Mathematical Expression	MSE
1	$\hat{y}(x) = 0.17546 \times \left[\cot^{-1}\left(-0.00013433 + 999.9972x^{-3.9692}\right)\right]^{-0.25194}$	3.6976E-18
2	$\hat{y}(x) = 0.074774 \times \left[\text{csch}^{-1}\left(-0.00011063 + 999.9098x^{-2.6637}\right)\right]^{-0.37542}$	1.7847E-17
3	$\hat{y}(x) = 0.17724 \times \left[\csc^{-1}\left(6.6542 \times 10^{-5} + 999.9952x^{-3.9923}\right)\right]^{-0.25048}$	**1.8511E-18**
4	$\hat{y}(x) = 0.131 \times \left[\coth^{-1}\left(0.00016646 + 987.6138x^{-3.3924}\right)\right]^{-0.29477}$	7.0826E-18
5	$\hat{y}(x) = 0.1776 \times \left(-6.2995 \times 10^{-7} + 996.28x^{-3.9949}\right)^{0.25032}$	2.1629E-18

($d \gg 9$). The MSE performance of LR reaches its optimal value at MSE = 1.4973E-17 when the polynomial order reaches $d = 30$. Then, that MSE increases again when d exceeds 30. Thus, the best solution obtained by LR is still not better than that obtained by UFO with only one block; i.e., $v = 1$.

Furthermore, it is important to mention two things here. Firstly, UFO is capable to select the best optimum analytical function and then fitting it to match the actual response. Secondly, if the pooling stage contains only one analytical function, then UFO will try its best to optimize that function by globally varying its coefficients and its internal and external arithmetic operators. Table 10.4 validates the first claim. To validate the second claim regarding optimizing one individual analytical function, Table 10.5 lists one possible function approximation detected for each analytical function shown in Figure 10.11. All the mathematical equations tabulated in Table 10.5 are highly precise. Moreover, if UFO is run several times using the same analytical function, then it could find many mathematical equations containing only that analytical function. This phenomenon reinforces the claim that the UFO optimization problems are highly constrained, non-convex, and nonlinear. To prove that, let's adjust the pooling stage to contain only $f = \exp$. For three independent executions, UFO generates the following exponential equations:

$$\hat{y}(x) = 0.3687 \times \left[\exp\left(-484.543 + 483.123x^{0.002958}\right)\right]^{-0.7019} \quad (10.55)$$

$$\hat{y}(x) = 0.9106 \times \left[\exp\left(706.023 - 705.976x^{0.0007197}\right)\right]^{1.969} \quad (10.56)$$

$$\hat{y}(x) = 0.08311 \times \left[\exp\left(258.956 - 260.572x^{-0.002487}\right)\right]^{-1.539} \quad (10.57)$$

As can be clearly seen from (10.55)-(10.57), UFO can still generate many highly precise equations even if only one analytical function is defined in the pooling stage.

Table 10.5: Optimizing Each Analytical Function by UFO to Approximate (10.54)

$\hat{y}(x) = 0.12432 \times \left[-5.5877 \times 10^{-10} + 716.3481x^{-3.1533}\right]^{0.31713}$	$\hat{y}(x) = 1 \times \left[\frac{1}{\frac{153.286}{153.3842x^{1}}}\right]^{-1}$
$\hat{y}(x) = 0.070685 \times \left[\sqrt{2.0934 \times 10^{-7} + 811.4007x^{-2.5283}}\right]^{0.79105}$	$\hat{y}(x) = 0.0064955 \times \left[\exp\left(823.3554 - 818.3473x^{0.0012166}\right)\right]^{1.0056}$
$\hat{y}(x) = 0.0045444 \times \left[\log_{10}\left(1.0877 + 0.7281x^{0.3679}\right)\right]^{-3.9957}$	$\hat{y}(x) = 0.42617 \times \left[\ln\left(\frac{2990.6913}{8463.8354x^{-1.1573}}\right)\right]^{-0.84314}$
$\hat{y}(x) = 0.47649 \times \left[\sin\left(\frac{32.8572}{269.2354x^{-2.838}}\right)\right]^{-0.35239}$	$\hat{y}(x) = 0.0091642 \times \left[\cos\left(523.0655 - 0.30362x^{0.26656}\right)\right]^{-3.9875}$
$\hat{y}(x) = 0.11809 \times \left[\tan\left(\frac{43.0978}{908.674x^{-1.4267}}\right)\right]^{-0.70084}$	$\hat{y}(x) = 0.43151 \times \left[\csc\left(\frac{120.438}{991.1701x^{-2.5085}}\right)\right]^{0.39868}$
$\hat{y}(x) = 0.0073526 \times \left[\sec\left(-230.9069 - 0.085712x^{0.49988}\right)\right]^{1.998}$	$\hat{y}(x) = 0.38785 \times \left[\cot\left(\frac{60.8019}{903.7866x^{-2.8493}}\right)\right]^{0.35094}$
$\hat{y}(x) = 0.43464 \times \left[\sin^{-1}\left(\frac{88.6433}{574.0321x^{-2.2405}}\right)\right]^{-0.44621}$	$\hat{y}(x) = \left(1.6314 \times 10^{-19}\right)\left[\cos^{-1}\left(\frac{549.4935}{727.7463x^{0.0047247}}\right)\right]^{-129}$
$\hat{y}(x) = 0.2713 \times \left[\tan^{-1}\left(\frac{86.221}{949.0733x^{-1.8397}}\right)\right]^{-0.54373}$	$\hat{y}(x) = 0.28349 \times \left[\csc^{-1}\left(0.0004751 + 142.6188x^{-3.9348}\right)\right]^{-0.25414}$
$\hat{y}(x) = 2835.957 \times \left[\sec^{-1}\left(0.0025999\left(741.9875x^{0.0053775}\right)\right)\right]^{-311.4167}$	$\hat{y}(x) = 0.18829 \times \left[\cot^{-1}\left(-0.00016689 + 794.7487x^{-3.9994}\right)\right]^{-0.25004}$
$\hat{y}(x) = 0.7884 \times \left[\sinh\left(\frac{106.6415}{816.441x^{-8.5613}}\right)\right]^{-0.1168}$	$\hat{y}(x) = 0.48814 \times \left[\cosh\left(-188.6685 + 186.7657x^{0.0092334}\right)\right]^{0.58673}$
$\hat{y}(x) = 0.55786 \times \left[\tanh\left(\frac{-130.247}{-935.4649x^{-3.3793}}\right)\right]^{-0.29595}$	$\hat{y}(x) = 0.53903 \times \left[\operatorname{csch}\left(\frac{145.0652}{881.8219x^{-2.9193}}\right)\right]^{0.3425}$
$\hat{y}(x) = \left(2.4917 \times 10^{-10}\right)\left[\operatorname{sech}\left(\frac{819.6646}{111.5549x^{0.03936}}\right)\right]^{-3.3254}$	$\hat{y}(x) = 0.49314 \times \left[\coth\left(\frac{44.0637}{636.8722x^{-3.7783}}\right)\right]^{0.26468}$
$\hat{y}(x) = 0.35742 \times \left[\sinh^{-1}\left(\frac{30.9341}{548.615x^{-2.795}}\right)\right]^{-0.35779}$	$\hat{y}(x) = 0.92461 \times \left[\cosh^{-1}\left(1.0889 + 0.47952x^{-0.6276}\right)\right]^{3.9998}$
$\hat{y}(x) = 0.24526 \times \left[\tanh^{-1}\left(\frac{86.2281}{996.3634x^{-1.7402}}\right)\right]^{-0.57449}$	$\hat{y}(x) = 0.16828 \times \left[\operatorname{csch}^{-1}\left(-8.9493 \times 10^{-5} + 805.5713x^{-3.7548}\right)\right]^{-0.26632}$
$\hat{y}(x) = \left(8.3719 \times 10^{-8}\right)\left[\operatorname{sech}^{-1}\left(\frac{-312.9291}{-667.2605x^{-0.025499}}\right)\right]^{49.4344}$	$\hat{y}(x) = 0.053339 \times \left[\coth^{-1}\left(0.00030081 + 863.6727x^{-2.3067}\right)\right]^{-0.43352}$

Table 10.6: Approximating (10.54) Using UFO with Two Blocks

$\hat{y}(x) = -0.21489 \times \left[565.6764\left(666.2517x^{-2.0847}\right)\right]^{-2.0207} + 0.0002617 \times \left[\coth^{-1}\left(0.0006829 + 952.0301x^{-0.83151}\right)\right]^{-1.2026}$
$\hat{y}(x) = 0.072723 \times \left[-1.9914 \times 10^{-8} + 833.531x^{-2.566}\right]^{0.38972} + \left(7.5529 \times 10^{-10}\right)\left[-58.6811 + 133.2821x^{-1.1791}\right]^{-1.2611}$
$\hat{y}(x) = \frac{0.040303 \times \left[\coth\left(646.9108\left(733.2051x^{-0.00087436}\right)\right)\right]^{-0.73264}}{0.42833 \times \left[-1.3903 \times 10^{-7} + 953.9184x^{-2.9028}\right]^{-0.3445}}$
$\hat{y}(x) = 0.18242 \times \left[\csc^{-1}\left(12.3671\left(10.019x^{-2.8326}\right)\right)\right]^{-0.35303} + \left(4.5827 \times 10^{-6}\right)\left[\sec^{-1}\left(54.6013\left(483.9804x^{-1.4086}\right)\right)\right]^{-0.29277}$
$\hat{y}(x) = -0.25507 \times \left[\csc^{-1}\left(8.1188\left(5.6328x^{-1.8217}\right)\right)\right]^{-0.54687} \times \left(-0.36218\right)\left[\sec\left(\frac{-249.1155}{659.9156x^{0.0062349}}\right)\right]^{3.9898}$
$\hat{y}(x) = \left(-0.65481\right)\left[\sqrt{-1.1469 \times 10^{-6} + 516.2001x^{-1.1695}}\right]^{1.1004} \times \left(-1.7411\right)\left[\sqrt{943.9714(893.3285x^{1.3637})}\right]^{-0.52294}$
$\hat{y}(x) = 0.65251 \times \left[\sin\left(\frac{534.7276}{949.6188x^{-1.33}}\right)\right]^{-0.75076} - 0.044861 \times \left[\sec^{-1}\left(1.2536 + 0.60111x^{-2.1153}\right)\right]^{-3.1868}$
$\hat{y}(x) = \frac{0.10018 \times \left[\sqrt{\frac{578.2222}{487.737x^{-1.209}}}\right]^{-1.0721}}{0.99761 \times \left[\operatorname{csch}^{-1}\left(-0.00032086 + 891.9541x^{-1.0004}\right)\right]^{0.35176}}$
$\hat{y}(x) = 0.83155 \times \left[\operatorname{csch}\left(\frac{178.9529}{769.6303x^{-7.9078}}\right)\right]^{0.12646} - 0.11276 \times \left[\cosh\left(-289.1244 - 86.9383x^{0.0070849}\right)\right]^{-0.75333}$
$\hat{y}(x) = 73.1403 \times \left[\cosh\left(-322.4704 + 869.7255x^{0.021588}\right)\right]^{-0.055133} \times 18.2763 \times \left[\sec^{-1}\left(-667.4785 - 440.7897x^{-2.387}\right)\right]^{50.7236}$
$\hat{y}(x) = 0.27262 \times \left[\operatorname{csch}^{-1}\left(402.9349\left(254.1353x^{-1.0766}\right)\right)\right]^{3.4506} + 0.044937 \times \left[-1.2118 \times 10^{-7} + 251.2577x^{-1.7813}\right]^{0.56139}$
$\hat{y}(x) = 2.8305 \times \left[\tanh^{-1}\left(\frac{8.9822}{306.0156x^{-0.67451}}\right)\right]^{-0.73398} \times 1.7222 \times \left[\tan^{-1}\left(0.0030404\left(8.0499x^{-0.44923}\right)\right)\right]^{1.1249}$
$\hat{y}(x) = 2.9743 \times \left[\cos^{-1}\left(\frac{263.6403}{485.202x^{0.18828}}\right)\right]^{-2.3021} - 2.0532 \times \left[\cosh\left(\frac{460.6656}{-491.3616x^{1.6341}}\right)\right]^{-0.069458}$
$\hat{y}(x) = 0.61724 \times \left[\sinh^{-1}\left(-337.6388 + 379.5425x^{-1.4315}\right)\right]^{-0.029977} \times 0.63717 \times \left[\sin^{-1}\left(\frac{107.9837}{408.586x^{-1.351}}\right)\right]^{-0.74409}$
$\hat{y}(x) = 0.3857 \times \left[\exp\left(-531.8529 + 538.3921x^{-0.010349}\right)\right]^{0.18773} \times 0.47407 \times \left[\log_{10}\left(113.0477\left(850.1277x^{1.8558}\right)\right)\right]^{0.29383}$
$\hat{y}(x) = 0.70554 \times \left[\csc\left(\frac{137.2438}{194.3859x^{-1.0018}}\right)\right]^{0.9987} - 0.26255 \times \left[\operatorname{csch}\left(\frac{55.5266}{214.3004x^{-1.3065}}\right)\right]^{-0.83067}$
$\hat{y}(x) = \frac{0.48504 \times \left[\operatorname{sech}\left(\frac{-305.8158}{67.0936x^{0.20919}}\right)\right]^{-0.57877}}{0.92758 \times \left[\sec^{-1}\left(-0.69167 - 0.53183x^{-1.1229}\right)\right]^{1.716}}$
$\hat{y}(x) = 0.25547 \times \left[\coth\left(665.8644 + 272.4719x^{-1.1728}\right)\right]^{0.8581} \times 0.12974 \times \left[\cot^{-1}\left(910.6428\left(360.5705x^{-3.7283}\right)\right)\right]^{-0.26822}$
$\hat{y}(x) = 0.0040468 \times \left[\sqrt{64.1789\left(94.8587x^{-1.5815}\right)}\right]^{1.2646} + \left(7.4846 \times 10^{-12}\right)\left[\ln\left(34.5214\left(51.9183x^{1.7568}\right)\right)\right]^{-1.5404}$
$\hat{y}(x) = 0.11153 \times \left[\coth^{-1}\left(0.00018533 + 983.4288x^{-3.1416}\right)\right]^{-0.3183} + 0.83108 \times \left[\exp\left(989.6478\left(276.4983x^{-0.92174}\right)\right)\right]^{-1.0032}$

This emphasizes an important fact that UFO is a very powerful computing system that can generate many highly acceptable solutions without necessarily sticking to the actual model.

To generate the exact function given in (10.54), UFO should tune (10.38) with $v = 1$. Thus, there is no $\circledcirc$ because there is only one block. For (10.36), the following settings can be used: $w = 1$, $a_0 = 0$, $a_1 = 1$, $b_1 = 1$, $c = -1$, $f = 1 \times ()$, and $\odot = +$. Also, there are other settings for (10.36), like swapping the value of b_1 with c or using $f = 1/()$ instead of $f = 1 \times ()$.

All these equations are so hard to be found manually by NLR. To make it harder, let's first approximate (10.54) using UFO with two blocks; i.e., $v = 2$. Table 10.6 shows some of the function approximations generated by UFO when $v = 2$. Table 10.7 shows some other functions approximated by UFO when three blocks are used. Now, to make it an unimaginable task, let's increase v to equal five. Table 10.8 shows nine different highly precise functions that behave like the original one given in (10.54). Doing that with NLR, using any existing package or software, is extremely difficult and semi-impossible. In UFO, all these mathematical models can be automatically generated by some easy steps from the end-users.

10.2.4.2 Dataset with Multiple Predictors

In this experiment, UFO is tested by trying to approximate a *multi*-dimensional function. For this mission, one of the popular "multiple predictors" regression problems listed in [214] is selected; specifically, Problem No. 6.18, which is about commercial properties. The problem contains four predictors and one response. The classical LR, using MINITAB software, gives MSE = 1.2925. The complete results are shown in Figure 10.15. As can be clearly seen, the fitted LR model is expressed as follows:

$$\hat{y}(X) = 12.201 - 0.1420x_1 + 0.2820x_2 + \ 0.62x_3 + 0.000008x_4 \tag{10.58}$$

From Figure 10.15, the model could be enhanced by removing x_3. To verify that, let's apply the following statistical hypothesis test:

- **Hypotheses:**

  $H_0 : \hat{y}(X) = \beta_0 + \beta_1 x1 + \cdots + \beta_n x_n(\text{except } \beta_k x_k)$, vs

  $H_1 : \hat{y}(X) = \beta_0 + \beta_1 x1 + \cdots + \beta_n x_n(\text{full model})$

Table 10.7: Approximating (10.54) Using UFO with Three Blocks

$\hat{y}(x) = 0.28716 \times \left[\sqrt{200.1984\left(0.72368x^{-3.988}\right)}\right]^{0.50151} + \frac{0.8219\times\left[412.0825\left(978.6682x^{3.2733}\right)\right]^{-3.6814}}{0.09709\times\left[\cot^{-1}\left(595.7699+506.8876x^{1.7266}\right)\right]^{-3.4883}}$
$\hat{y}(x) = 0.55396 \times \left[\cos\left(-871.2321 + 201.6821x^{-0.1333}\right)\right]^{2.365} \times 0.3604 \times \left[\exp\left(-886.8883 + 9.9104x^{2.3111}\right)\right]^{0.43147}$ $+ 0.22052 \times \left[1.2845 \times 10^{-7} + 422.8289x^{-4}\right]^{0.25}$
$\hat{y}(x) = \frac{0.20564\times\left[\log_{10}\left(344.9036\left(505.6904x^{-3.9691}\right)\right)\right]^{2.6513}}{0.95966\times\left[\cosh^{-1}\left(\frac{855.5958}{66.2019x^{-0.45165}}\right)\right]^{2.6086}} + 0.20268 \times \left[\coth\left(836.5106 + 180.6433x^{-2.8411}\right)\right]^{3.6321}$
$\hat{y}(x) = 0.63059 \times \left[\operatorname{csch}\left(\frac{147.2604}{744.3803x^{-0.45981}}\right)\right]^{3.0819} \times \left(1.0839 \times 10^{-13}\right)\left[\operatorname{csch}^{-1}\left(402.3976 + 21.3574x^{-1.4543}\right)\right]^{-2.3697}$ $+ 0.5353 \times \left[\sinh\left(\frac{59.978}{633.7971x^{-3.7724}}\right)\right]^{-0.26506}$
$\hat{y}(x) = 0.71495 \times \left[\cosh^{-1}\left(315.12\left(100.97x^{2.3476}\right)\right)\right]^{-0.6048} + \frac{0.70055\times\left[\exp\left(\frac{700.2766}{518.3942x^{-0.044932}}\right)\right]^{-1.3313}}{0.20792\times\left[\operatorname{csch}\left(\frac{579.5012}{804.4555x^{-0.56632}}\right)\right]^{-1.6513}}$
$\hat{y}(x) = 0.74657 \times \left[\tanh^{-1}\left(\frac{275.957}{939.1818x^{-0.20412}}\right)\right]^{1.2059} + 0.21469 \times \left[\coth^{-1}\left(\frac{620.866}{122.1709x^{1.0294}}\right)\right]^{-0.96316}$ $- 0.25811 \times \left[\tan^{-1}\left(8.6725 + 3.1393x^{-3.7071}\right)\right]^{-0.7634}$
$\hat{y}(x) = \frac{0.35298\times\left[\frac{556.9623}{425.7536x^{-3.2577}}\right]^{-0.0019521}}{0.79351\times\left[178.8507\left(715.4334x^{2.0678}\right)\right]^{-0.026306}} \Big/ 0.77274\times\left[\frac{591.566}{849.5351x^{-1.5582}}\right]^{0.67261}$
$\hat{y}(x) = 0.0025388 \times \left[\operatorname{sech}\left(142.4783 + 721.5986x^{-1.6552}\right)\right]^{0.13521} \times 0.077937 \times \left[\sin\left(98.448 - 337.1757x^{-1.1683}\right)\right]^{0.66216}$ $+ 0.003323 \times \left[\operatorname{csc}^{-1}\left(\frac{582.3977}{79.2419x^{0.34793}}\right)\right]^{-2.865}$
$\hat{y}(x) = 0.00097822 \times \left[\operatorname{sech}\left(\frac{-319.9498}{968.679x^{2.3483}}\right)\right]^{3.3848} + 0.15806 \times \left[\operatorname{csch}\left(\frac{100.9914}{863.9426x^{-1.1627}}\right)\right]^{0.85976}$ $- 0.99999 \times \left[\sinh\left(340.9859\left(-809.7326x^{-3.9623}\right)\right)\right]^{-0.00028025}$
$\hat{y}(x) = 0.78817 \times \left[\tan^{-1}\left(\frac{433.9396}{715.4548x^{0.3292}}\right)\right]^{2.9726} + 0.0005376 \times \left[\operatorname{sech}\left(515.8009 + 754.1391x^{-0.48638}\right)\right]^{3.8962}$ $+ 0.52078 \times \left[\tan^{-1}\left(\frac{123.327}{581.1304x^{-3.1356}}\right)\right]^{-0.33003}$
$\hat{y}(x) = 0.46215 \times \left[\sec^{-1}\left(-469.8135 - 228.7741x^{1.8082}\right)\right]^{-2.0114} + 0.2997 \times \left[\operatorname{csch}\left(\frac{244.7276}{499.4228x^{-0.59619}}\right)\right]^{1.6805}$ $- 0.04565 \times \left[\cosh^{-1}\left(411.3169 - 285.3804x^{0.21929}\right)\right]^{0.54276}$
$\hat{y}(x) = 0.74478 \times \left[\csc\left(\frac{377.0028}{524.3372x^{-1.1343}}\right)\right]^{0.88302} - \frac{0.48522\times\left[980.053\left(441.6082x^{-0.13446}\right)\right]^{0.22366}}{0.52276\times\left[\cosh\left(\frac{313.8568}{-111.7855x^{0.1957}}\right)\right]^{2.5556}}$

That is, the above hypotheses can be simplified as:

$H_0 : \beta_k = 0$, vs

$H_1 : \beta_k \neq 0$

- **Test Statistic:**

$$\bar{T}_k = \frac{b_k - \beta_{k,0}}{\text{SE}\{b_k\}} \tag{10.59}$$

where the abbreviation SE stands for the standard error. Based on the results shown in Figure 10.15, the t-test for the third predictor x_3 gives the following:

$$\bar{T}_3 = \frac{0.62 - 0}{1.09} = 0.57$$

which is also shown in the results.

- **Decision Rule:**

H_0 is rejected if $|\bar{T}_k| > t_{\alpha/2,\delta}$; where α is the significance level and δ is the degree

Table 10.8: Approximating (10.54) Using UFO with Five Blocks

$\hat{y}(x) = 0.78381 \times \left[\tanh\left(714.1836 + 511.929x^{3.6579}\right)\right]^{1.0593} \times \frac{0.51201 \times \left[\frac{53.3997}{509.4323x^{-0.72815}}\right]^{3.813}}{\frac{0.6934 \times \left[\tan^{-1}\left(-28.2815 + 25.5385x^{-2.8513}\right)\right]^{-0.012412}}{0.13118 \times \left[\csc^{-1}\left(1.1382 + 336.8611x^{-3.0887}\right)\right]^{1.2222}}}$ $- 0.2138 \times \left[\log_{10}\left(592.6504\left(844.1441x^{-0.99428}\right)\right)\right]^{-2.1188}$
$\hat{y}(x) = \frac{\frac{0.2 \times \left[\cosh\left(-380.264\left(533.5572x^{-0.075141}\right)\right)\right]^{-1.6771}}{0.22643 \times \left[\sinh^{-1}\left(-528.1268 - 157.7303x^{2.6865}\right)\right]^{-2.8098}}}{0.25268 \times \left[\tanh\left(297.5591\left(175.8015x^{-0.982}\right)\right)\right]^{-0.47167}} + 0.80804 \times \left[\frac{794.4891}{969.7055x^{-0.93504}}\right]^{-1.0695}$ $+ \left(1.0613 \times 10^{-5}\right)\left[\tanh\left(248.9198\left(842.4548x^{-3.7234}\right)\right)\right]^{1.6776}$
$\hat{y}(x) = 0.22493 \times \left[\operatorname{csch}^{-1}\left(-0.00017281 + 384.955x^{-3.9901}\right)\right]^{-0.25062} - 0.83515 \times \left[\cosh\left(-741.742 - 116.3386x^{-3.4413}\right)\right]^{-2.0615}$ $\times \frac{0.6492 \times \left[\operatorname{csch}\left(842.5606 - 837.8733x^{-0.86678}\right)\right]^{3.9276}}{0.97561 \times \left[\sec^{-1}\left(419.6351\left(130.5246x^{1.3478}\right)\right)\right]^{0.94862}} \times 0.69188 \times \left[\cos^{-1}\left(-74.0703 + 84.5093x^{-3.0207}\right)\right]^{-1.7875}$
$\hat{y}(x) = 0.16954 \times \left[-2.2231 \times 10^{-10} + 757.869x^{-3.7362}\right]^{0.26765} - \frac{0.98445 \times \left[\tan\left(-249.032\left(875.3248x^{3.5743}\right)\right)\right]^{3.1469}}{0.014476 \times \left[\exp\left(588.8714\left(591.1178x^{3.3275}\right)\right)\right]^{3.0857}}$ $+ \frac{0.086001 \times \left[\sin\left(\frac{924.0614}{835.9721x^{2.2019}}\right)\right]^{-3.008}}{0.81012 \times \left[\cosh\left(-129.999\left(80.6873x^{2.483}\right)\right)\right]^{3.7104}}$
$\hat{y}(x) = \frac{\frac{0.054798 \times \left[\operatorname{sech}\left(415.3721 - 146.8216x^{1.9605}\right)\right]^{3.2265}}{0.94015 \times \left[\tanh\left(\frac{-841.916}{28.5981x^{-1.6567}}\right)\right]^{2.6512}}}{0.042993 \times \left[\coth\left(530.6026\left(574.5799x^{-3.0574}\right)\right)\right]^{-3.0561} \times 0.091996 \times \left[\tanh\left(367.3058\left(-921.2646x^{-2.4415}\right)\right)\right]^{-3.2177}}$ $+ 0.18093 \times \left[-5.9973 \times 10^{-10} + 933.0797x^{-3.9999}\right]^{0.25}$
$\hat{y}(x) = 0.28891 \times \left[\sqrt{\frac{138.4233}{570.128x^{-1.1401}}}\right]^{-1.7543} + 0.0015852 \times \left[\coth^{-1}\left(317.084\left(701.7044x^{3.501}\right)\right)\right]^{-3.9321}$ $\times 0.03175 \times \left[\sinh\left(757.0352\left(948.709x^{-3.6447}\right)\right)\right]^{-3.4321} + \frac{\sec^{-1}\left(243.2794\left(808.8609x^{-3.9695}\right)\right)^{-3.6905}}{0.089284 \times \left[954.8456 + 926.579x^{-3.9886}\right]^{3.5338}}$
$\hat{y}(x) = 0.75481 \times \left[\tan^{-1}\left(903.9293\left(656.7784x^{-0.083875}\right)\right)\right]^{-3.9887} \times 0.96754 \times \left[\cot^{-1}\left(-0.00059333 + 375.0625x^{-2.8018}\right)\right]^{-0.35692}$ $+ \frac{\frac{0.0078248 \times \left[\csc^{-1}\left(353.8925 + 672.4628x^{-1.7422}\right)\right]^{3.7315}}{0.0060527 \times \left[\sin^{-1}\left(\frac{919.8442}{854.2373x^{-3.2506}}\right)\right]^{-3.4902}}}{0.99671 \times \left[\sinh\left(41.6377 + 689.471x^{2.5082}\right)\right]^{3.9598}}$
$\hat{y}(x) = 0.98143 \times \left[\cos^{-1}\left(\frac{-493.8149}{981.8258x^{0.21219}}\right)\right]^{3.8448}$ $+ \frac{0.69132 \times \left[\csc^{-1}\left(490.338\left(174.2226x^{-1.592}\right)\right)\right]^{-0.090724}}{0.46756 \times \left[\cosh\left(357.3929 - 294.2559x^{-1.7548}\right)\right]^{0.16519}} \times \left(0.2091\right)\left[\sin^{-1}\left(-357.1289 + 440.421x^{-0.36736}\right)\right]^{-1.9198}$ $- 0.31985 \times \left[\exp\left(\frac{-831.9605}{461.5203x^{0.045963}}\right)\right]^{-2.1682}$
$\hat{y}(x) = 0.64693 \times \left[\cot^{-1}\left(\frac{121.2455}{48.1286x^{2.1216}}\right)\right]^{-0.47135} - 0.1699 \times \left[\sinh^{-1}\left(\frac{460.1562}{955.3551x^{-1.2777}}\right)\right]^{2.578}$ $+ \frac{0.20131 \times \left[\ln\left(-594.1622 - 773.0637x^{3.8372}\right)\right]^{-3.1135}}{0.1224 \times \left[\cot^{-1}\left(815.3567\left(753.7305x^{-3.0855}\right)\right)\right]^{-1.1455}} \times \left(0.95959\right)\left[\operatorname{csch}\left(-350.2832 - 777.3485x^{2.2873}\right)\right]^{3.351}$

of freedom. The significance level is calculated as follows:

$$\alpha = 1 - \text{CL} \tag{10.60}$$

where CL is the confidence level. For example, $\alpha = 0.05$ or 5% if CL = 95%, and $\alpha = 0.01$ or 1% if CL = 99%.

The degree of freedom is calculated as follows:

$$\delta = \mathcal{O} - (n + 1) \tag{10.61}$$

where $\mathcal{O}$ stands for the number of observations in the given dataset.

The other alternative decision rule can be conducted by comparing the p-value with α directly as follows:

H_0 is rejected if the p-value $< \alpha$

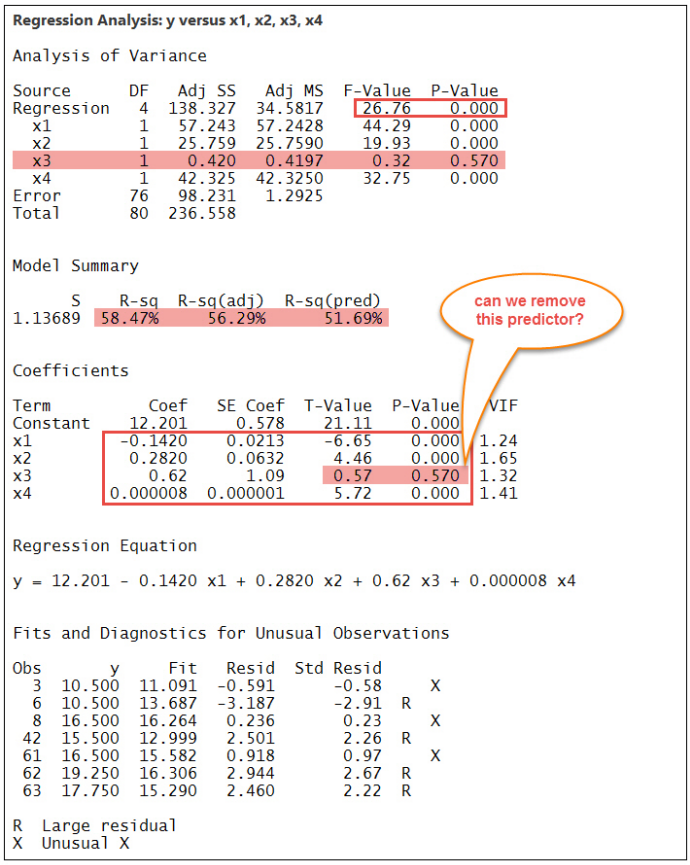


Figure 10.15: The Complete LR Results Obtained for the Second Problem by Using All the Predictors

- **Conclusion:**

  By referring to Figure 10.15, the p-value of $\{x_1, x_2, x_4\}$ are equal to zero. Thus, statistically, they cannot be removed from the regression model. On the opposite side, the p-value of x_3 is 0.57, which is smaller than α for both CL = 95% and CL = 99%. This means that the third predictor is insignificant, and thus the null hypothesis is concluded.

It has been found that the preceding solution can be enhanced to MSE = 1.2810 by removing the third predictor x_3. The complete results are shown in Figure 10.16 where the fitted LR model is expressed as follows:

$$\hat{y}(X) = 12.371 - 0.1442x_1 + 0.2672x_2 + 0.000008x_4 \tag{10.62}$$

For ANN, a topology built based on the LM training algorithm and one hidden

Regression Analysis: y versus x1, x2, x4

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	3	137.91	45.969	35.88	0.000
x1	1	60.84	60.841	47.49	0.000
x2	1	27.86	27.857	21.74	0.000
x4	1	50.29	50.287	39.25	0.000
Error	77	98.65	1.281		
Total	80	236.56			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
1.13189	58.30%	56.67%	53.10%

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	12.371	0.493	25.10	0.000	
x1	-0.1442	0.0209	-6.89	0.000	1.20
x2	0.2672	0.0573	4.66	0.000	1.37
x4	0.000008	0.000001	6.27	0.000	1.27

Regression Equation

y = 12.371 - 0.1442 x1 + 0.2672 x2 + 0.000008 x4

Fits and Diagnostics for Unusual Observations

Obs	y	Fit	Resid	Std Resid	
3	10.500	11.193	-0.693	-0.67	X
6	10.500	13.562	-3.062	-2.76	R
38	13.000	15.207	-2.207	-2.02	R
42	15.500	13.032	2.468	2.24	R
62	19.250	16.292	2.958	2.69	R
63	17.750	15.314	2.436	2.21	R

R Large residual
X Unusual X

Figure 10.16: The Complete LR Results Obtained for the Second Problem by Removing x_3 from the Full Model

layer of 30 neurons is tested. Table 10.9 lists the solutions obtained by 15 different transfer functions (TFs) implemented in the MATLAB framework. For SVM, Table 10.10 shows the solutions obtained by 9 different configurations.

Now, let's start using UFO to regress this numerical problem. By adjusting $w \in [0, 2]$, $c \in [1, 4]$, $\odot \in [+, -]$, $\circledcirc \in [+, -]$, $v = 2$, and $\mathfrak{p} = 10$, an MSE performance of 0.70748 is obtained. The UFO model is mathematically expressed as follows:

$$\begin{aligned}\hat{y}(X) = 1.2614\Big[\sec^{-1}\Big(&- 1.0312 - 0.47708x_1^{0.53066} - \ 233.9268x_2^{-3.9988} + 1.1355x_3^{2.0431} \\ &- 9.7155x_4^{-0.27601}\Big)\Big]^{2.9602} + 1.0853\Big[\sec^{-1}\Big(94.4056 - 94.4155x_1^{-0.0016307} \\ &- 5.6759x_2^{-0.46192} - 377.4971x_3^4 + 1.0886 \times 10^{-5}x_4^{0.79642}\Big)\Big]^{2.0196} \end{aligned} \tag{10.63}$$

Table 10.9: Regressing the Second Problem Using ANNs

TF	MSE	TF	MSE	TF	MSE
compet	2.0115	netinv	3130.8	satlin	1.8715
elliotsig	2.8692	poslin	2.5187	satlins	1.6320
hardlim	2.1945	purelin	**1.2771**	softmax	1.5485
hardlims	2.5235	radbas	9.0554	tansig	5.7999
logsig	9.0898	radbasn	2.2214	tribas	5.1674

Table 10.10: Regressing the Second Problem Using SVMs

Configuration		Conventional SVM			Standardized SVM		
Kernel	**Solver**	**Iterations**	**Vectors**	**MSE**	**Iterations**	**Vectors**	**MSE**
Linear	SMO	1000000	7	71.9666	272	68	1.2650
	ISDA	1000000	4	84.0850	83	81	52.466
	L1QP	7	65	209.705	6	68	1.2655
Polynomial	SMO	0	0	2.9205	11998	75	0.7722
	ISDA	0	0	2.9205	36737	76	1.6112
	L1QP	*the problem is non-convex*			7	75	0.7722
Gaussian	SMO	103	68	0.4967	116	66	**0.6544**
	ISDA	103	80	2.9419	129	71	3.8164
	L1QP	7	68	**0.4966**	8	66	**0.6544**

It is clear that UFO is a competitive computing system comparing with other well-established systems. For this particular problem, UFO beats all the solutions obtained by LRs and ANNs, and most of the solutions obtained by SVM. Only with the Gaussian kernel function using SMO or L1QP solver, SVM can win in this competition. However, UFO is tested here with its primitive structure. There are so many points that someone can maneuver around; which might lead him/her to generate better solutions. Add to that, the solution obtained by UFO is represented in a very concise mathematical equation expressed by only two terms. It is readable, and thus it can be directly implemented in many other applications.

10.2.5 Further Discussion

During designing the UFO structures, many points have been left for this subsection. Also, during testing the performance of the structure given in Figure 10.7, many

phenomena have been observed. To make the study well-arranged, the first sub-subsection is allocated to the observations recorded during conducting the numerical experiments. The next subsubsections cover the capabilities, strengths, applications, and existing/extended features of UFO.

10.2.5.1 Some Observations

It is obvious that UFO did an amazing job to solve the preceding problems. However, some highly important phenomena have been observed, which need to be addressed carefully. The main observations with some technical suggestions are listed below:

- In the building stage, it has been found that it is hard to change the jth analytical function f_j inside each block. This phenomenon could happen due to:
  - the system complexity, or
  - the building stage is likely to trap into local minima every time the tuning stage gets a new solution.
- To solve the above issue:
  - The function tolerance in the tuning stage, which is shown in Figure 10.13, can be set to a small value. Thus, just a little improvement could be applied to the population in each iteration. Also, the value of that tolerance could be exponentially increased as the number of iterations increases.
  - However, reducing the tolerance value could decrease the exploitation level. Thus, the convergence rate could also be affected as a consequence of that adjustment. Based on this, the setting should compromise between the exploration and exploitation levels.
  - Forcing the building stage to select other analytical functions available in the pool when a specific number of iterations is completed. Here, the good past solutions will not be ruined if the elitism stage is activated. Instead, these solutions can be recycled again in the next iterations or they can be stored in a temporary location and then called back if needed.
- The exploration level can be increased by giving more weight to the building

stage, and vice versa for the exploitation level. Both the exploration and exploitation levels are affected by the tuning stage. Thus, giving more weight to the building stage means less weight is given to the tuning stage, and vice versa.

- The exploration level is also affected by the optimization type used in the building stage and the constraint-handling technique embedded in that algorithm. It has been found that the random search technique consumes more CPU time, but it is more capable to explore the search space than EPFs if the population size $\mathfrak{p}$ is very small and the problem is highly nonlinear and non-convex.

- Based on the last point regarding $\mathfrak{p}$, if a meta-heuristic optimization algorithm is used in the building stage, then the exploration level could be increased by increasing the mutation probability and ensuring that the entire population (or most of it) is covered.

10.2.5.2 Differences Between UFO and ANNs

By looking at the mechanism of UFO, it can be realized that there are many major differences between the two computing systems, which make UFO totally different than ANNs. Such differences are:

- ANNs contain only weights and biases. On the opposite side, the basic structure of UFO contains internal/external weights, internal/external exponents, and intercepts. These coefficients can be just values or they could be modeled as internal functions embedded inside the main v functions; which could be considered for highly advanced UFO structures.

- The weights in ANNs are normalized, while the coefficients $\{w, a_0, a, b, c\}$ of UFO are not. Moreover, for each block, the external weight $w_{i,j}$, the internal exponents $b_{k,i,j}$, and the external exponents $c_{i,j}$ can be set as continuous or discrete variables. This property can be activated in meta-heuristic optimization algorithms by adjusting the discretization probabilities of these coefficients via the spinners located at the top-left side of Figure 10.11. If $w_{i,j}$ is switched to the discrete mode, then the jth block of the ith row is completely deactivated when $w_{i,j} = 0$. Thus, $g_{i,j}(X)$ is rejected from the overall function $\hat{y}_i(X)$ approximated

for the ith response. Moreover, the kth term of $g_{i,j}(X)$, i.e. $a_{k,i,j}x_k^{b_{k,i,j}}$, equals $a_{k,i,j}$ when $b_{k,i,j} = 0$. Furthermore, the entire block function $g_{i,j}(X)$ equals one when $c_{i,j} = 0$.

As a future addition, the preceding discretization process can be extended to cover a-coefficients. Thus, a new feature selection approach can be provided here, which is a built-in merit of UFO. For example, if $a_{k,i,j}$ is switched to the discrete mode, then the kth predictor is rejected from $g_{i,j}(X)$ when $a_{k,i,j} = 0$. Similar thing happens with the intercept $a_{0,i,j}$ when it acts as a weight; i.e., when $\odot_{1,i,j}$ is treated as $\times$ (or $\div$ if $a_{1,i,j}x_1^{b_{1,i,j}} \neq 0$). Thus, for that case, the first predictor x_1 is rejected when $a_{0,i,j} = 0$ regardless of the value of $a_{1,i,j}$.

- In most ANNs, the interconnection between neurons is highly recommended, while, in UFO, the interconnection between blocks located in different rows is just an optional feature. That is, each row of UFO can work independently without any connection to any other row.

- After defining the number of data variables (i.e., predictors and responses) and hidden layers in ANNs, the network size can also be increased by increasing the number of neurons assigned to each hidden layer. In UFO, the dimension remains constant once the data size and blocks are defined.

- In ANNs, the knowledge is distributed among nodes, and thus facing a black-box problem. In UFO, the knowledge is confined within a readable mathematical equation.

- ANNs are trained via one optimization algorithm (i.e., back-propagation or meta-heuristic algorithm). Even if a hybrid optimization algorithm is used, it will be placed in one single stage. Of course, there are some special structures where the two independent optimization strategy is applied to optimize the hyperparameters of ANNs, such as the studies reported in [19,198]. However, this additional external optimizer is an optional sub-algorithm, which can be deactivated without affecting the core mechanism of ANNs. On the opposite side, the mathematical equations expressed in UFO are built through two mandatory independent optimization stages. The external one must be implemented to

build mathematical equations via a global mixed-integer probabilistic/stochastic optimization algorithm, and the internal one must be implemented to fit the coefficients of these equations via a local gradient-based classical optimization algorithm.

For recurrent UFO (or RUFO), the concept of recycling streams is borrowed from recurrent neural networks (RNNs). It has to be said that RUFO is just a possible future version of UFO, and thus it is not covered in this study. For the classical UFO, the above differences are still valid when it is compared with RNNs.

10.2.5.3 Differences Between UFO and Classical Symbolic Regression Techniques

Compared with classical **symbolic regression** (**SR**) techniques, similar to the one reported in [369], SRs could perform well in some applications. However, they suffer from many difficulties, such as:

- SRs depend on **genetic programming** (**GP**), while UFO is structured to work with any optimization algorithm; including the primitive random search algorithm (RSA).
- SRs build their mathematical models using a tree-based approach, while UFO is not.
- In SRs, both arithmetic operators and analytical functions are mixed together, while UFO treats them differently where each variable type has its own category and search domain and thus they cannot be mixed together.
- In UFO, the randomness of function expressions is controlled by g_j inside each block. These v functions, i.e. $\{g_1, g_2, \cdots, g_v\}$, act as molds. Thus, the mathematical expressions are randomly changed by varying the arithmetic operators and analytical functions, but without violating the main structures of these v functions. Please, refer to (10.36)-(10.38).
- Based on its controllable random expressions, UFO can be used to universally transform the original predictors. Thus, UFO is flexible to be hybridized with other ML tools.

- In SRs, GP optimizes the equation parameters, analytical functions, and arithmetic operators together, while UFO has multiple stages approach. The initial expression is created in the building stage using any optimization algorithm. Then, in the tuning stage, the analytical functions and arithmetic operators are held constant without any change.

- In [369], it can be clearly seen that the dependent and independent variables of the mathematical expressions generated by the proposed SR do not have exponents.

- As the number of predictors increases, the implementation of SRs becomes very hard. For example, in [369], only *one*- and *two*-dimensional problems are introduced.

- Because UFO has a fixed base-structure or mathematical mold represented by $\{g_1, g_2, \cdots, g_v\}$, so it is very easy to export the mathematical models generated by UFO by just replacing $\{w_j, f_j, a_{0,j}, a_{k,j}, b_{k,j}, c_j, \odot_{k,j}, \circledcirc_j\}$ located in a template file. In SRs, there is a need to trace each subtree and check whether it contains an analytical function, an arithmetic operator, or an independent variable.

- As a metaphor, by referring to classical LR, SRs act like the forward selection method and UFO acts like the backward elimination method[4].

- As the number of nodes increases (which is common with *high*-dimensional problems), the tree will expand and becomes more complicated. Again, in UFO, the problem complexity depends on the base-structure or mathematical mold of g_j and the number of blocks used.

- In SRs, the expression size increases proportionally with the tree's depth, while UFO has a fixed size for all the iterations.

- By referring to the second point of the last subsubsection, it can be concluded that UFO has a built-in feature selection property.

[4]Thus, it is interesting to hybridize them together to see if they can act like the stepwise regression method.

- UFO can be used to reduce the problem dimension when $v < n$, and vice versa when $v > n$. This point will be discussed later.

In general, UFO and SRs have different structures, mechanisms, and search domains. SRs are great tools, no doubt about that, and it is interesting to see a future hybridization between UFO and SRs. For example, **universal symbolic regressions** (**USRs**).

10.2.5.4 Strengths and Weaknesses of UFO

During analyzing the operation behavior of UFO, it has been found that this computing system has many wonderful strengths. Some of them are listed below:

- The results generated by UFO are readable and understandable.
- These mathematical equations could lead to some new facts and reveal some phenomena hidden behind the data.
- Because the results are represented as mathematical equations, so these results can be exchanged with other users either as an electronic format or as a hard-copy.
- Based on the last point, the UFO results can be electronically exchanged by a wide range of formats; including programming languages, text editors, pictures, emails, text messages, etc. This option cannot be done at all with ANNs, because they can be exchanged only with users who have the same programming language that was used in designing these ANNs; or through what is called an Open Neural Network Exchange (ONNX).
- Also, based on the third point, the UFO results can be implemented by using programming languages, MS Excel and its alternatives, scientific calculators, or even by just using hands for some simple functions.
- Again, based on the third point, the UFO results can be saved with a very limited file size; from hundreds of bytes to just a few kilo-bytes.
- The mathematical models extracted from UFO can be used in many applications. Some of these prospective applications are listed in Figure 10.18.

- UFO can be used with both small and big data.
- The AI feature selection problem is solved automatically because it is an integral part of UFO. This property is described in the second point of the differences between UFO and ANNs.

The GUI presented here is just a prototype, which is designed to validate UFO and evaluate its performance. The current programming codes still need to be optimized, and the processing speed can be expedited by applying parallel and multithreaded programming. Also, it is known that MATLAB is proprietary software. Thus, many open-source/free programming languages (such as Python, Java, C/C++, Julia, GNU Octave, and R) can be used to build the next generation UFOs. Some of them are much faster than MATLAB and especially suited for high computational numerical analysis. Based on this, to have a fair comparison with other computing systems, the speed performance of UFO should be evaluated against others once the preceding points are addressed. Of course, similar to other ML tools, UFO must have some inherent weaknesses, which need to be discovered after applying it on a wide range of applications. The three important points that need to be studied in more detail are: 1. the accuracy across different applications, 2. the efficiency versus the data size, and 3. the processing speed.

The next subsection provides some guidance to develop the future versions of UFO. Applying part or all of the guidance could have a significant improvement on the current UFO performance in terms of solution quality and processing speed.

10.2.5.5 Improving the Processing Speed of UFO

Different options can be adopted to accelerate the processing speed of UFO. Three viable options are presented below:

Reducing the Dimension of the Building Stage

One of the main factors that affect the processing speed of UFO is the dimension $\aleph$ of the building stage, especially if a population-based optimization algorithm is implemented in that stage. An effective reduction in $\aleph$ will lead to a significant improvement in the overall CPU time consumed by UFO. This can be done by dealing

with $\{f, \odot, \circledcirc\}$ variables as binary strings; i.e., sequences of bits. To clarify this point, let's see how these discrete variables can be converted to some limited binary strings:

- **Dimension Reduction in f:**
  By referring to the main window of the GUI shown in Figure 10.11, it can be seen that there are 30 analytical functions defined in the pooling stage. Thus, the jth discrete variable f_j lies between 0 and 29; where 0 is the index of the identity function $f_j = 1 \times ()$ and 29 is the index of the inverse hyperbolic cotangent function $f_j = \coth^{-1}()$. Thus, each jth analytical function f_j can be represented by a 5-bit binary variable where the last two states (i.e., 11110 and 11111) are not allowed. For the structure given in Figure 10.7, there are "v" functions. Based on this, one binary string of length "$5 \times v$" can be used instead of using "v" variables. Thus, the dimension reduction in f is $(v - 1)$.

  For example, by taking $v = 2$, all the block functions can be represented by a binary string φ of length 10. To clarify it more, suppose that the binary string φ equals the following:

  $$\varphi = [1010100111]$$

  Then, each five bits will represent one analytical function f_j. Thus, by checking the analytical functions listed in Figure 10.11, column by column, the sequence of bits given in φ represents the following analytical functions:

  $$f_1 = \text{analytical function no. } 21 = \text{csch}()$$

  $$f_2 = \text{analytical function no. } 7 = \cos()$$

  It is important to say that the length of φ and the number of bits used to represent f_j depend on the number of blocks v and the number of analytical functions defined in the pooling stage.

- **Dimension Reduction in $\odot$:**
  If the internal universal arithmetic operators have a basic set of $\odot \in \{+, -, \times, \div\}$, then only two bits can be used to select one operator among these four basic types. That is, every two bits will represent one $\odot_{k,j}$ as follows:

  $$00 : \odot_{k,j} = +$$

$$01 : \odot_{k,j} = -$$

$$10 : \odot_{k,j} = \times$$

$$11 : \odot_{k,j} = \div$$

Because there are "$v \times n$" internal operators for (10.36), so only one binary string of length "$2 \times v \times n$" can be used instead of using "$v \times n$" variables. Thus, the dimension reduction in $\odot$ is $(vn - 1)$.

For example, take $n = 3$ and $v = 2$. Then, there are 6 variables of type $\odot$. To save 5 variables out of 6, a binary string of length 12 is used. For further clarification, suppose that a binary string ξ equals the following:

$$\xi = [110110001001]$$

Thus, ξ can be translated to have the following arithmetic operators:

$$\odot = [\div, -, \times, +, \times, -]$$

- **Dimension Reduction in $\circledcirc$:**
  Similar thing can be done on the external universal arithmetic operators. It is known that, for (10.38), there are "$v - 1$" variables of type $\circledcirc$. Thus, a binary string of length "$2 \times v - 2$" can be used for a basic set of $\circledcirc \in \{+, -, \times, \div\}$. Based on this, $(v - 2)$ variables can be saved.

  For example, assume $v = 10$. This means that there are 9 variables of type $\circledcirc$. Thus, an 18-bit binary string should be used here to save 8 variables. For further clarification, suppose that a binary string ζ equals the following:

$$\zeta = [100001001101001110]$$

  Using the same steps done on ξ, the following arithmetic operators can be extracted from ζ:

$$\circledcirc = [\times, +, -, +, \div, -, +, \div, \times]$$

Summing up all these savings, the total dimension $\aleph$ of the building stage, given in (10.39), is effectively reduced to:

$$\begin{aligned} \aleph &= 3vn + 5v - 1 - (v - 1) - (vn - 1) - (v - 2) \\ &= 2vn + 3v + 3 \end{aligned} \tag{10.64}$$

Therefore, by comparing (10.64) with (10.40), the new dimension $\aleph$ becomes:

$$\aleph = \mathfrak{D} + 3 \tag{10.65}$$

That is, $\aleph$ is bigger than $\mathfrak{D}$, given in (10.40), by just 3. This extra value represents the three binary strings used for $\{f, \odot, \circledcirc\}$.

Reducing the Dimension of the Tuning Stage

It has been found that the tuning stage could be enhanced by tuning one or multiple coefficients of $\{w, a_0, a, b, c\}$ but not all of them. Remember that the variables $\{f, \odot, \circledcirc\}$ are not considered in the tuning stage. Thus, for v blocks, there are "v" variables of type $\{w, a_0, c\}$ and "$v \times n$" variables of type $\{a, b\}$. This means that (10.40) can be effectively reduced down as follows:

- If any one of $\{w, a_0, c\}$ is selected:

$$\mathfrak{D} = v \tag{10.66}$$

- If any vector of $\{a, b\}$ is selected:

$$\mathfrak{D} = vn \tag{10.67}$$

Remember that (10.66) should be used if only one k coefficient of a or b (i.e., a_k or b_k) is tuned. Table 10.11 shows some examples of the normal and reduced dimensions of the building and tuning stages when the single O/P stream structure is used.

Once the reduced dimension formulas of the building and tuning stages are derived, the extended versions for the multiple O/P streams structure given in Figure 10.10 can be easily obtained by just multiplying them by m. For the building stage, (10.64) becomes:

$$\begin{aligned} \aleph &= (2vn + 3v + 3)m \\ &= 2mvn + 3mv + 3m \end{aligned} \tag{10.68}$$

By comparing (10.68) with (10.52), the new dimension $\aleph$ becomes:

$$\aleph = \mathfrak{D} + 3m \tag{10.69}$$

Table 10.11: Comparing (10.39) with (10.64) for the Building Stage, and (10.40) with (10.66) or (10.67) for the Tuning Stage

Case		**Normal Dimension**		**Reduced Dimension**	
n	v	$\aleph$	$\mathfrak{D}$	$\aleph$	$\mathfrak{D}$
1	2	15	10	13	2
3	1	13	9	12	1-3
2	2	21	14	17	2-4
6	4	91	60	63	4-24
4	4	67	44	47	4-16
8	6	173	114	117	6-48
15	9	449	297	300	9-135
18	12	707	468	471	12-216

Again, $\aleph$ is equal to $\mathfrak{D}$ plus $3m$ because there are 3 binary strings for each row used in UFO with multiple O/P streams. Thus, for m rows, the extra variables that need to be added to the problem dimension are $3m$. For the tuning stage, (10.66) and (10.67) become:

- If any one of $\{w, a_0, c\}$ is selected for all the m responses:

$$\mathfrak{D} = mv \tag{10.70}$$

- If any vector of $\{a, b\}$ is selected for all the m streams:

$$\mathfrak{D} = mvn \tag{10.71}$$

Table 10.12 shows some examples of the normal and reduced dimensions of the building and tuning stages when the multiple O/P streams structure is used.

Moving from Population-Based Algorithms to Single-Point Algorithms

There is no argument that the processing speed of any population-based optimization algorithm increases proportionally with the population size $\mathfrak{p}$. Based on this fact, the UFO computing system could be effectively accelerated by ensuring that the global

Table 10.12: Comparing (10.51) with (10.68) for the Building Stage, and (10.52) with (10.70) or (10.71) for the Tuning Stage

Case			Normal Dimension		Reduced Dimension	
n	v	m	$\aleph$	$\mathfrak{D}$	$\aleph$	$\mathfrak{D}$
1	2	2	30	20	26	4
3	1	2	26	18	24	2-6
2	2	2	42	28	34	4-8
6	4	3	273	180	189	12-72
4	4	3	201	132	141	12-48
8	6	4	692	456	468	24-192
15	9	7	3143	2079	2100	63-945
18	12	10	7070	4680	4710	120-2160

meta-heuristic optimization algorithm used in the building stage is not a population-based type. Some of these algorithms are the simulated annealing (SA) and tabu-search (TS) algorithms [345].

The other more saving option is to replace the preceding single-point probabilistic optimization algorithms with just a few lines of code to generate one random solution per each iteration. That is, jumping from the probabilistic mode to the stochastic mode could reduce some percentage of the CPU time. But, such a random search algorithm (RSA) is not smart like SA and TS. However, RSA could be more effective to explore the search space. This interesting point could be addressed in the future with some numerical analysis.

10.2.5.6 Should UFO Compete or Integrate with Other ML Computing Systems?

There is no doubt that entering UFO in competition with other ML tools (such as ANNs and SVMs) will not guarantee to win for both parts, because each tool has its own strengths and weaknesses; and, of course, particular applications. ANNs and SVMs are great tools, and they proved themselves as smart computing systems. Therefore, it would be preferred to integrate UFO with these popular computing systems, so UFO could fill some deficits that may exist in ML problems. So far,

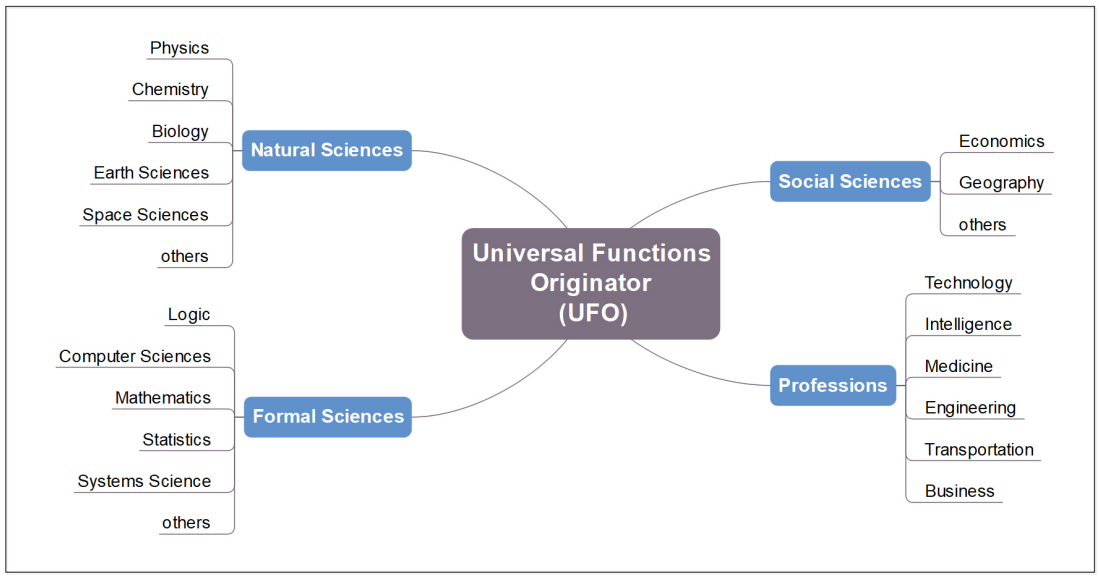


Figure 10.17: Main Disciplines and Their Branches Where UFO Can Be Applied

there are - at least - eight possible hybridizations that can be made between UFO and other existing ML tools. For example, UFO is hybridized with LR and NLR to act as a **universal transformation unit** (**UTU**). Also, UFO is hybridized with SVM as UTU. Based on some early-stage analysis, UFO could make basic SVM structures (i.e., using simple kernel and solver) very competitive to highly advanced SVM structures. Furthermore, UFO could enhance the performance of shallow ANNs (i.e., using one hidden layer with a simple activation function and a basic learning algorithm) to compete with deep ANNs. These hybrid computing systems will be covered later in more detail.

10.2.5.7 Prospective Applications of UFO

From the fact that UFO can present its results as readable mathematical equations, so this novel computing system can be applied to a wide range of applications in all computation-based disciplines shown in Figure 10.17. Some of these applications, which are summarized in Figure 10.18, are listed below:

- UFO can convert real data to mathematical models, which can be used as objective functions in optimization algorithms. This is a very useful application, especially with classical and classical-based hybrid optimization algorithms

where the derivative part of their objective functions is required.

- UFO can be involved in nonlinear control systems to provide highly precise approximations of actual responses.
- UFO can translate real data to meaningful mathematical equations that could explain some hidden facts.
- UFO can work as a general-purpose regression unit to fit the given data universally without adjusting its model (1st, 2nd, 3rd, etc, order polynomial equation), mode (linear or nonlinear), significant predictors (removing insignificant ones), etc.
- Based on the last point, UFO could be used in forecasting, classification, and anomaly detection applications.
- UFO can simplify a ready-made complicated model into a compact model while preserving its accuracy.
- UFO can counterfeit a simple data to be in a very complex appearance; which, with the last point, act like encoding and decoding - just a metaphor.

The last two points are further explained in the next subsubsection.

10.2.5.8 Using UFO as a Complicator/Decomplicator

In the preceding numerical experiments, it has been seen how UFO can provide very concise mathematical equations as approximated functions. Also, it has been seen how UFO can be used to complicate (10.54) if multiple blocks are used; please, refer to Tables 10.6-10.8.

Now, let's initiate the UFO computing system with 12 blocks (i.e., $v = 12$) and then let's use it to solve the second problem; which is about the commercial properties. Table 10.13 shows three possible solutions. It is obvious that UFO can automatically produce unimaginable/unbelievable mathematical equations by increasing v. However, the problem dimensions, of the building and tuning stages, also increase; please, refer to the formulas of $\aleph$ and $\mathfrak{D}$.

Table 10.13: Regressing the Second Problem Using UFO with Twelve Blocks

$$\hat{y}(X) = 2 \times \left[\csc^{-1}\left(504.8174 - 379.1057x_1^6 - \frac{542.1394x_2^{5.5136}}{\left(510.2828x_3^{5.5096}\right)\left(687.1808x_4^{5.4975}\right)}\right)\right]^{5.4991} \times (1.6082)\left[\sec^{-1}\left(-153.6002\left(906.6041x_1^{6.4475}\right) - 147.5448x_2^{5.7355} + 389.8284x_3^{5.5479}\right.\right.$$
$$\left.\left. + \ 78.3845x_4^{5.6356}\right)\right]^{5.4902} - 1.5008 \times \left[\csc^{-1}\left(-5.4688 + \frac{\left(807.1961x_1^5\right)\left(471.3288x_2^{5.6371}\right)}{\left(53.1052x_3^{5.5072}\right)\left(286.3782x_4^5\right)}\right)\right]^5 \times (1.4523)\left[\sec^{-1}\left(-353.0839 + 162.5845x_1^{5.7528} + \frac{\left(384.6687x_2^{5.9534}\right)\left(951.8658x_3^5\right)}{703.7455x_4^{5.3784}}\right)\right]^{5.4948}$$
$$- \frac{\left[\tan^{-1}\left(397.3856 - 716.9933x_1^{5.5098} - 99.828x_2^{5.317} + 401.3661x_3^6 - 54.1446x_4^{5.4885}\right)\right]^6}{\left[\sec^{-1}\left(-509.5902\frac{833.9537x_1^{5.6878}}{18.2907x_2^{5.45}} + \frac{493.1002x_3^{5.7749}}{732.2584x_4^{5.4077}}\right)\right]^{5.4933}} + (1.5093)\left[\sec^{-1}\left(-38.7103 + 136.5152x_1^{5.4422} - \frac{\left(265.4346x_2^{5.6375}\right)\left(846.5192x_3^{5.502}\right)}{-912.3558x_4^{5.8798}}\right)\right]^{5.245}$$
$$- \frac{2\times\left[\sec^{-1}\left(713.2295 + \frac{465.413x_1^{5.5666}}{-464.9587x_2^{5.5006}} + 636.5428x_3^5 + 738.9126x_4^{5.4991}\right)\right]^{5.4181}}{1.1728\times\left[\sec^{-1}\left(-127.0536 - \left(161.0669x_1^{5.4832}\right)\left(698.0637x_2^5\right) + 684.3183x_3^{5.6789} - 684.8302x_4^6\right)\right]^{5.5426}} \times \frac{1.2739\times\left[\cot^{-1}\left(-888.8515 + 734.3779x_1^6 - \frac{553.455x_2^{5.8483}}{129.48x_3^{5.5068}} + 42.6962x_4^6\right)\right]^6}{\left[\sec^{-1}\left(\frac{\frac{331.1339}{876.1638x_1^{5.3914}}}{-916.615x_2^{5.4968}} + 923.4683x_3^{5.5308} - 240.0135x_4^5\right)\right]^{5.4012}}$$
$$\hat{y}(X) = \frac{2\times\left[\csc^{-1}\left(\frac{-172.409}{880.1683x_1^6} + \frac{270.6819x_2^{5.5136}}{653.2466x_3^{5.5096}} + 345.2217x_4^{5.4975}\right)\right]^{5.4991}}{1.6082\times\left[\sec^{-1}\left(926.3499 + \frac{973.3671x_1^{6.4475}}{832.6425x_2^{5.7355}} - 203.3868x_3^{5.5479} + 225.5691x_4^{5.6356}\right)\right]^{5.4902}} + \frac{1.5008\times\left[\csc^{-1}\left(\frac{472.0325}{-674.5834x_1^5} - 884.488x_2^{5.6371} + 675.625x_3^{5.5072} + 667.2035x_4^5\right)\right]^5}{1.4523\times\left[\sec^{-1}\left(99.7335\left(811.2763x_1^{5.7528}\right) + \frac{576.0998x_2^{5.9534}}{-996.4457x_3^5} - 740.0136x_4^{5.3784}\right)\right]^{5.4948}}$$
$$- \frac{\left[\tan^{-1}\left(-11.3156 + \frac{236.302x_1^{5.5098}}{39.5118x_2^{5.317}} + \frac{743.39x_3^6}{927.8905x_4^{5.4885}}\right)\right]^6}{\left[\sec^{-1}\left(\frac{-612.9253\left(833.9537x_1^{5.6878}\right)}{18.2907x_2^{5.45}} + \frac{22.1179x_3^{5.7749}}{-742.0407x_4^{5.4077}}\right)\right]^{5.4933}} + (1.5093)\left[\sec^{-1}\left(-38.7103 + 136.5152x_1^{5.4422} - \frac{\left(105.5413x_2^{5.6375}\right)\left(653.317x_3^{5.502}\right)}{36.4433x_4^{5.8798}}\right)\right]^{5.245}$$
$$+ \left[\sec^{-1}\left(419.5011 + 523.9903x_1^{5.5666} - 573.5831x_2^{5.5006} + \frac{363.9306x_3^5}{-197.5875x_4^{5.4991}}\right)\right]^{5.4181} \times (1.1728)\left[\sec^{-1}\left(484.6141\frac{-588.812x_1^{5.4832}}{\left(462.5735x_2^5\right)\left(240.4419x_3^{5.6789}\right)} - 243.7532x_4^6\right)\right]^{5.5426}$$
$$\times \frac{\left\{1.2739\times\left[\cot^{-1}\left(586.0486 - \left(123.068x_1^6\right)\left(536.3229x_2^{5.8483}\right) + \frac{126.1595x_3^{5.5068}}{782.4499x_4^6}\right)\right]^6\right\}\cdot\left\{\left[\sec^{-1}\left(0.76515 - 459.7159x_1^{5.3914} - 823.097x_2^{5.4968} + \frac{706.5458x_3^{5.5308}}{492.0511x_4^5}\right)\right]^{5.4012}\right\}}{2\times\left[\csc^{-1}\left(564.2835 - \left(643.2015x_1^{5.5152}\right)\left(555.4268x_2^{5.6327}\right) - \frac{78.4396x_3^5}{625.3982x_4^5}\right)\right]^5}$$
$$\hat{y}(X) = \frac{2\times\left[\csc^{-1}\left(-87.6785 - (880.1683x_1^6)(597.2705x_2^{5.5136}) + 749.5563x_3^{5.5096} + 701.251x_4^{5.4975}\right)\right]^{5.4991}}{1.6082\times\left[\sec^{-1}\left(\frac{132.6188}{-426.6437x_1^{5.4475}} + 968.1264x_2^{5.7355} - 770.5147x_3^{5.5479} - 890.6134x_4^{5.6356}\right)\right]^{5.4902}} \times (1.5008)\left[\csc^{-1}\left(-324.8393 - (859.9469x_1^5)(360.936x_2^{5.6371}) + \frac{933.6432x_3^{5.5072}}{543.3943x_4^5}\right)\right]^5$$
$$\times (1.4523)\left[\sec^{-1}\left(-99.7335 + 811.2763x_1^{5.7528} - (576.0998x_2^{5.9534})(475.7401x_3^5) - 560.2107x_4^{5.3784}\right)\right]^{5.4948} - \frac{\left[\tan^{-1}\left(-11.3156 + \frac{236.302x_1^{5.5098}}{39.5118x_2^{5.317}} + \frac{362.7755x_3^6}{19.4012x_4^{5.4885}}\right)\right]^6}{\left[\sec^{-1}\left(579.1491\frac{833.9537x_1^{5.6878}}{18.2907x_2^{5.45}} + \frac{861.0283x_3^{5.7749}}{-585.3222x_4^{5.4077}}\right)\right]^{5.4933}}$$
$$+ (1.5093)\left[\sec^{-1}\left(-38.7103 + 136.5152x_1^{5.4422} + 409.5367x_2^{5.6375} \times \frac{84.9299x_3^{5.502}}{305.9748x_4^{5.8798}}\right)\right]^{5.245} + \left[\sec^{-1}\left(419.5011 + 633.1341x_1^{5.5666} - 573.5831x_2^{5.5006} - \frac{609.863x_3^5}{146.1346x_4^{5.4991}}\right)\right]^{5.4181}$$
$$\times (1.1728)\left[\sec^{-1}\left(-36.72 + (425.6865x_1^{5.4832})(-228.083x_2^5) - 240.4419x_3^{5.6789} - 552.6083x_4^6\right)\right]^{5.5426} \times \frac{(1.2739)\left[\cot^{-1}\left(586.0486 - (66.6533x_1^6)(507.0168x_2^{5.8483}) - \frac{128.6284x_3^{5.5068}}{386.795x_4^6}\right)\right]^6}{(2)\left[\csc^{-1}\left(564.2835 - (876.5443x_1^{5.5152})(555.4268x_2^{5.6327}) - \frac{544.2768x_3^5}{247.7411x_4^5}\right)\right]^5}$$
$$\times \ \left[\sec^{-1}\left(494.7514 + 768.4806x_1^{5.3914} + 87.548x_2^{5.4968} - 807.1847x_3^{5.5308} - 898.0332x_4^5\right)\right]^{5.4012}$$

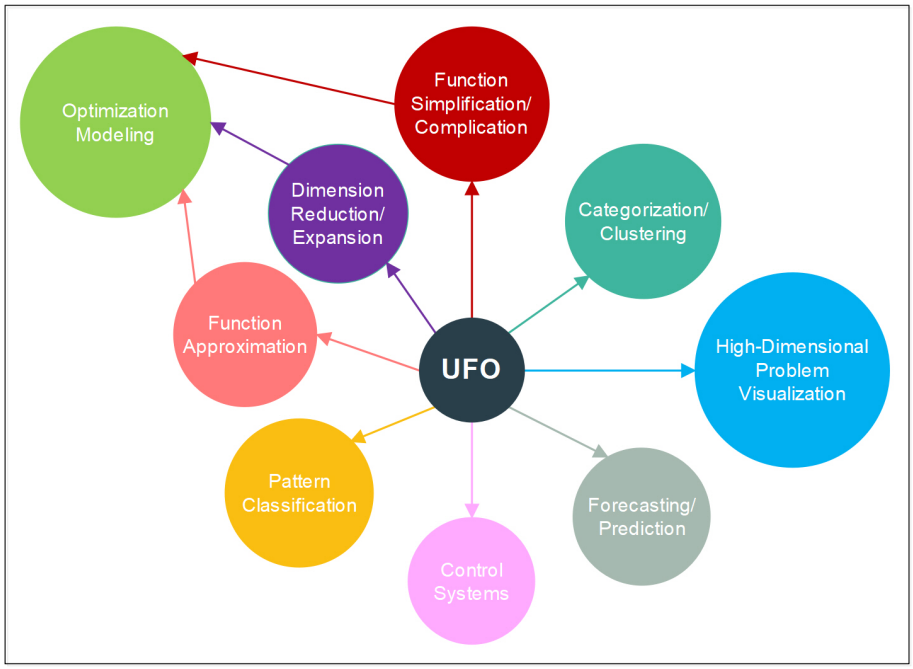


Figure 10.18: Some Possible Applications of UFO

Based on this, UFO can simplify or complicate actual responses. That is, UFO can act as a complicator if v is large and as a de-complicator/simplifier if v is small. Apparently, there is no reason to complicate functions approximated by UFO. However, perhaps there might be some future applications of this distinct feature. For example, one of the possible applications of this feature is to let UFO act as an encoder or a decoder, but everything here is in mathematical expressions. Thus, a simple function could be reformulated to be in a very complex way by setting v to a large value and then simplifying it again by setting v to a small value.

Also, *high*-dimensional objective functions could be reduced down by this feature, especially for combinatorial and mixed-integer optimization problems. This feature could be expanded to include any function. If the output of each block g_j is considered as a modified predictor, then UFO can act as a dimension expansion unit (DEU) when $v > n$ and as a dimension reduction unit (DRU) when $v < n$. Thus, *high*-dimensional functions could be reduced down to 1-dimensional functions to visualize them in 2D plots or to 2-dimensional functions to visualize them in 3D plots. This distinct feature can be effectively harnessed when UFO is hybridized with LR/NLR; as will be seen later.

10.2.5.9 Further Simplification and Complication Stages

It has been said that UFO can act as a simplifier or de-complicator if v is small. However, a further simplification unit could be embedded as an extended stage; just before displaying the UFO results. This unit can do pure mathematical simplifications to the final equations generated by UFO. For example, if UFO produces $1-\sin^2(x)$, then that equation could be further simplified to $\cos^2(x)$.

Similarly, an independent complication stage could be embedded as well. Thus, by some expansions and substitutions, a further complication process could be carried out. For the same trigonometric example given above, let's suppose that the UFO computing system generates $\cos^2(x)$. This complication unit can convert it to $1-\sin^2(x)$, then to $1-\cos^2(x)+\cos(2x)$, then to $1-\cos(2x)-\sin^2(x)+\frac{1-\tan^2(x)}{1+\tan^2(x)}$, until ending up with $1-\cos(2x)-\sin^2(x)+\frac{2\tan(x)\cot(2x)}{2-2\tan(x)cot(2x)}$. This complicated trigonometric equation is just equal to $\cos^2(x)$.

However, these extended simplification and complication units are just optional stages that could be built in the future with more advanced UFO structures.

10.2.5.10 UFO with Just Polynomial Equations

It has been said that the mode of $\{w_j, a_{0,j}, a_{k,j}, b_{k,j}, c_j\}$ could be continuous or discrete. Thus, the mathematical expression given in (10.36) can be considered as a source of endless random functions. For example, if the exponents $b_{k,j}$ and c_j are kept discrete for all the predictors $\{x_k : k = 1, 2, \cdots, n\}$ and analytical functions $\{f_j : j = 1, 2, \cdots, v\}$, respectively, then UFO can act as a generator of polynomial regression models when $f_j = 1 \times ()$, $c_j = 1$, $b_{k,j} > 0$, $\odot_{k,j} \in \{+, -, \times\}$, and $\circledcirc_{\substack{j \\ j<v}} \in \{+, -, \times\}$ $\forall\ j \in [1, v]$ and $\forall\ k \in [1, n]$.

10.3 Other UFO-Based ML Computing Systems

As said before, it will be great if UFO is hybridized with other ML computing systems. That is, the strong side of UFO can be integrated with others to have superior computing systems. For this, let's borrow the block diagram concept of UFO described by (10.36). If there are v blocks, then they can be separated from each other by removing the external arithmetic operators given in (10.38). Thus, by arranging

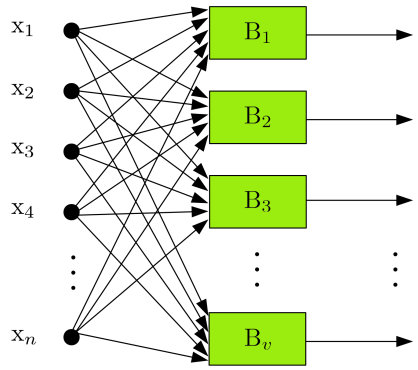


Figure 10.19: Universal Transformation Unit (UTU)

these v blocks vertically and then supplying each block by all n features, a "universal transformation unit (UTU)" can be created as shown in Figure 10.19. This unit is the basis of many hybrid ML computing systems. Some of them are covered in the next subsections.

10.3.1 Universal Transformation-based Regression (UTR)

Referring to [214], the performance of regression models could be enhanced by transforming their predictors before being sent to LR/NLR. Thus, the proposed **universal transformation-based regression** (**UTR**) is actually divided into two types:

1. UFO + LR $\rightarrow$ **universal linear regression** (**ULR**)
2. UFO + NLR $\rightarrow$ **universal nonlinear regression** (**UNR**)

The structure of the first type is shown in Figure 10.20. The same concept can be applied to UNR by just replacing the polynomial equation by any other user-defined nonlinear equation. Figure 10.21 shows our first GUI that is designed by Python with Qt library to serve as ULR.

To see how ULR is different than UFO, (10.54) is approximated using a different number of blocks. If $v = 1$, one of the precise approximate equations is:

$$\hat{y}\left(X\right) = \beta_0 + \beta_1 \cdot \left\{ 0.3345780004208794 \times \left[\csc^{-1}\left(9.899568821547554 \times \left(8.123576354948314 x^{0.742064970390371} \right) \right) \right]^{1.3571922193733161} \right\} \quad (10.72)$$

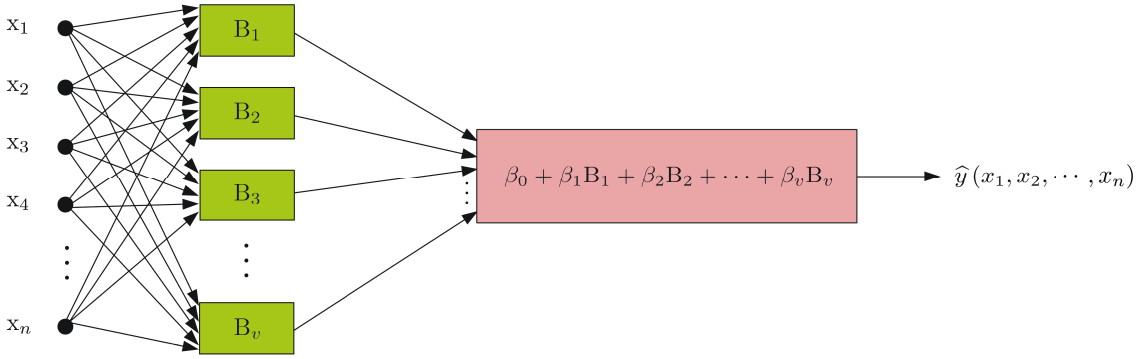


Figure 10.20: Universal Linear Regression (ULR)

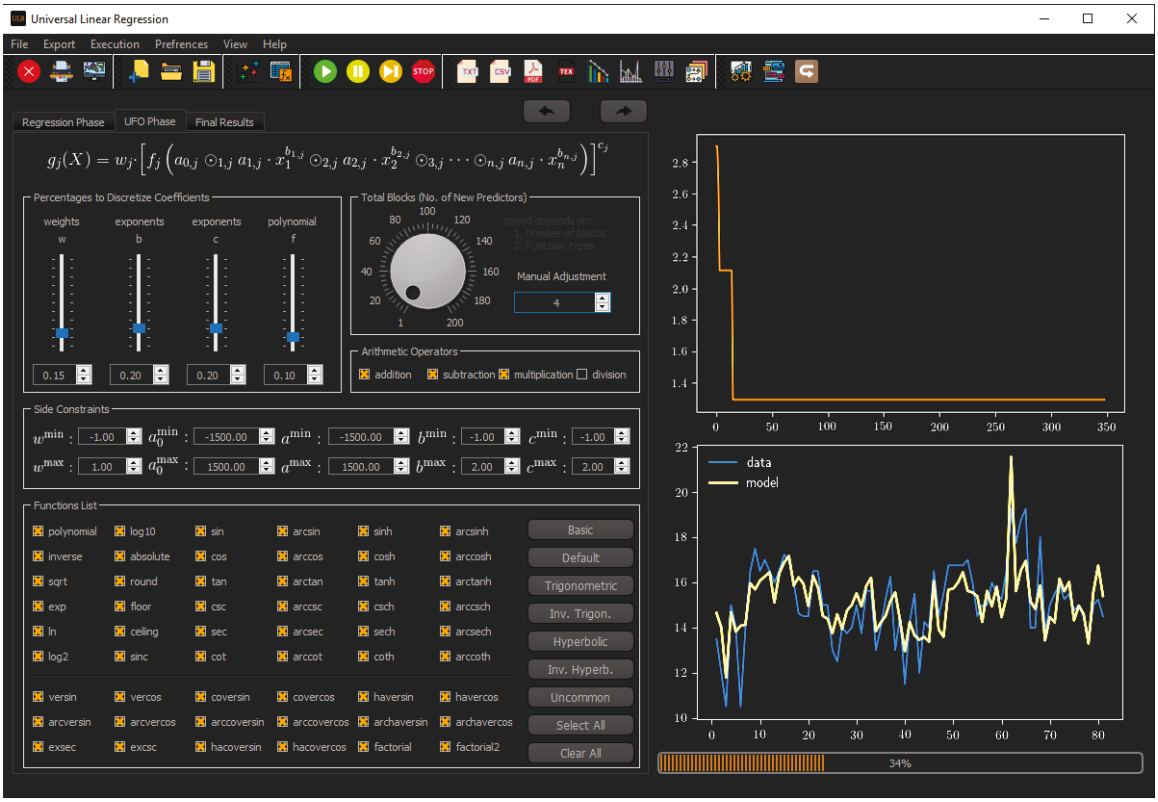


Figure 10.21: Our First Universal Linear Regression Software

where β_0 and β_1 are the regression coefficients.

The last equation represents a transformed simple linear regression model where β_0 is the intercept and β_1 is the slope, and the transformed predictor is the equation inside the curly brackets. Even if someone wants to complicate the approximation process, he/she can do that by simply increasing v to a very large value. For instance, Figure 10.22 shows one possible approximation made of 22 blocks and using many familiar and unfamiliar analytical functions. All the beta coefficients are listed as well. Now, imagine if $v = 200$ and only the inverse function is used for $\{f_1, f_2, \cdots, f_{200}\}$. The surprising thing is that ULR can still produce highly precise equations, such as the one shown in Figure 10.23.

This innovative computing system has been tested using different regression problems. It beats SVM for the 4-dimensional regression problem solved before by UFO. By referring to Table 10.13, if $v = 110$, then ULR can produce highly complicated approximate equations, such as the one shown in Figure 10.24.

10.3.2 Support Function Machine (SFM)

Because the special type of SVM, which is support vector regression (SVR), can be used in the place of LR and NLR, so a new computing system can be designed by replacing the regression box of Figure 10.20 by SVR. The structure of this novel computing system is shown in Figure 10.25. As can be seen, each transformed predictor is a function of the n actual predictors, so this hybrid system is called "**support function machine** (**SFM**)". This new ML tool is tested with many regression problems. It shows good results even with using the simplest kernel; i.e. the linear kernel. That is, UFO, which acts as UTU, can take care of the nonlinearity issue of the dataset before being sent to SVR.

There are other possible ways to hybridize UFO with SVM. For instance, (10.36) could be used as a universal kernel where $\{w_j, f_j, a_{0,j}, a_{k,j}, b_{k,j}, c_j, \odot_{k,j}\}$ are the hyperparameters that need to be tuned externally. Thus, it is a double-edged sword. It is a flexible universal kernel, but it has many parameters and thus it requires more iterations to converge to better solutions.

$$
\begin{aligned}
\hat{y}\,(X) = \beta_0 \;&+\; \beta_1 \cdot \left\{0.8356895690183792 \times \left[\text{floor}\left(1.610128216622726 \times \left(3.503322909583164 x^{0.6882510375128712}\right)\right)\right]^{1.0}\right\} \\
&+ \beta_2 \cdot \left\{0.17775121406785033 \times \left[\tan^{-1}\left(5.145996237457608 - \left(4.983308716373896 x^{1.9523538585197748}\right)\right)\right]^{1.0862817696180649}\right\} \\
&+ \beta_3 \cdot \left\{0.08969157083976165 \times \left[\text{factorial}\left(8.687896648811451 - \left(1.1487026368051967 x^{0.625191669697313}\right)\right)\right]^{0.04175109836322477}\right\} \\
&+ \beta_4 \cdot \left\{0.8863070310063199 \times \left[\sec\left(0.8402067704436067 - \left(3.5850993540974097 x^{0.429963956507321}\right)\right)\right]^{1.936136482279787}\right\} \\
&+ \beta_5 \cdot \left\{0.3903979879745616 \times \left[1 \times \left(3.8671023812756564 \div \left(0.632170501914312 x^{1.0}\right)\right)\right]^{1.0701153916387183}\right\} \\
&+ \beta_6 \cdot \left\{0.39554446084809813 \times \left[\text{hacovercos}\left(8.092388250365717 \times \left(5.312827281077195 x^{1.5609339257976238}\right)\right)\right]^{1.3959463443465352}\right\} \\
&+ \beta_7 \cdot \left\{0.9165111870464147 \times \left[\text{hacovercos}\left(8.424501877645925 + \left(2.391555020494913 x^{1.9802292967541257}\right)\right)\right]^{0.0}\right\} \\
&+ \beta_8 \cdot \left\{0.306999127340699 \times \left[\tan^{-1}\left(3.2991060378944272 \times \left(4.182831443017272 x^{1.3309540507283597}\right)\right)\right]^{0.9027550596493124}\right\} \\
&+ \beta_9 \cdot \left\{0.25826104128044747 \times \left[\text{excsc}\left(5.22876618981722 \times \left(1.1853441136582477 x^{1.960387673307723}\right)\right)\right]^{0.0}\right\} \\
&+ \beta_{10} \cdot \left\{0.6391568421682077 \times \left[\sinh\left(2.5875624779146933 - \left(5.981788906431697 x^{1.0}\right)\right)\right]^{0.012946326529686969}\right\} \\
&+ \beta_{11} \cdot \left\{0.029123007663689915 \times \left[\text{haversin}\left(1.0365037193137883 + \left(7.098317246136961 x^{0.0}\right)\right)\right]^{0.6653380680283723}\right\} \\
&+ \beta_{12} \cdot \left\{0.528024715230682 \times \left[\cosh\left(0.3309328780898535 + \left(5.051458323426767 x^{0.046545552902668375}\right)\right)\right]^{0.0}\right\} \\
&+ \beta_{13} \cdot \left\{0.9428270343896633 \times \left[\csc\left(0.9482864370627853 \times \left(0.22275494849695487 x^{2.0}\right)\right)\right]^{0.8648053102030977}\right\} \\
&+ \beta_{14} \cdot \left\{0.8672103817976009 \times \left[\exp\left(0.541720094929069 - \left(5.990643494057051 x^{0.5707745593744691}\right)\right)\right]^{1.8262302661517713}\right\} \\
&+ \beta_{15} \cdot \left\{0.4353602469433462 \times \left[\text{floor}\left(2.997335757369682 - \left(8.468989891484561 x^{1.3374940994590745}\right)\right)\right]^{1.0}\right\} \\
&+ \beta_{16} \cdot \left\{0.2609431670761281 \times \left[\log_2\left(0.8450583161342662 \div \left(0.44369729874790154 x^{1.6765829509252974}\right)\right)\right]^{1.6294965374952803}\right\} \\
&+ \beta_{17} \cdot \left\{0.18921059640160587 \times \left[\text{round}\left(5.21626662062652 \times \left(1.2626402068509135 x^{1.0}\right)\right)\right]^{1.0}\right\} \\
&+ \beta_{18} \cdot \left\{0.0 \times \left[\text{vercos}\left(2.447905513304299 - \left(6.674343631457332 x^{1.8501129090185857}\right)\right)\right]^{1.9101146169099794}\right\} \\
&+ \beta_{19} \cdot \left\{0.5173698312414654 \times \left[\text{sinc}\left(3.191401627378212 \div \left(1.5554292854571439 x^{1.368442961807131}\right)\right)\right]^{1.9076665079976993}\right\} \\
&+ \beta_{20} \cdot \left\{1.0 \times \left[\text{versin}\left(9.166296648608922 \div \left(3.921274673877603 x^{1.5525506478264033}\right)\right)\right]^{2.0}\right\} \\
&+ \beta_{21} \cdot \left\{0.4679046459336942 \times \left[\sinh\left(9.645469357702796 \div \left(4.063702976591351 x^{0.8104373769049602}\right)\right)\right]^{0.5665174377362614}\right\} \\
&+ \beta_{22} \cdot \left\{0.20975836880868048 \times \left[\cot^{-1}\left(1.0679113055159772 - \left(4.837372305180379 x^{0.12608945454895393}\right)\right)\right]^{1.0479684727482217}\right\}
\end{aligned}
$$

Final Results

ULR - Notepad

File Edit Format View Help

OLS Regression Results

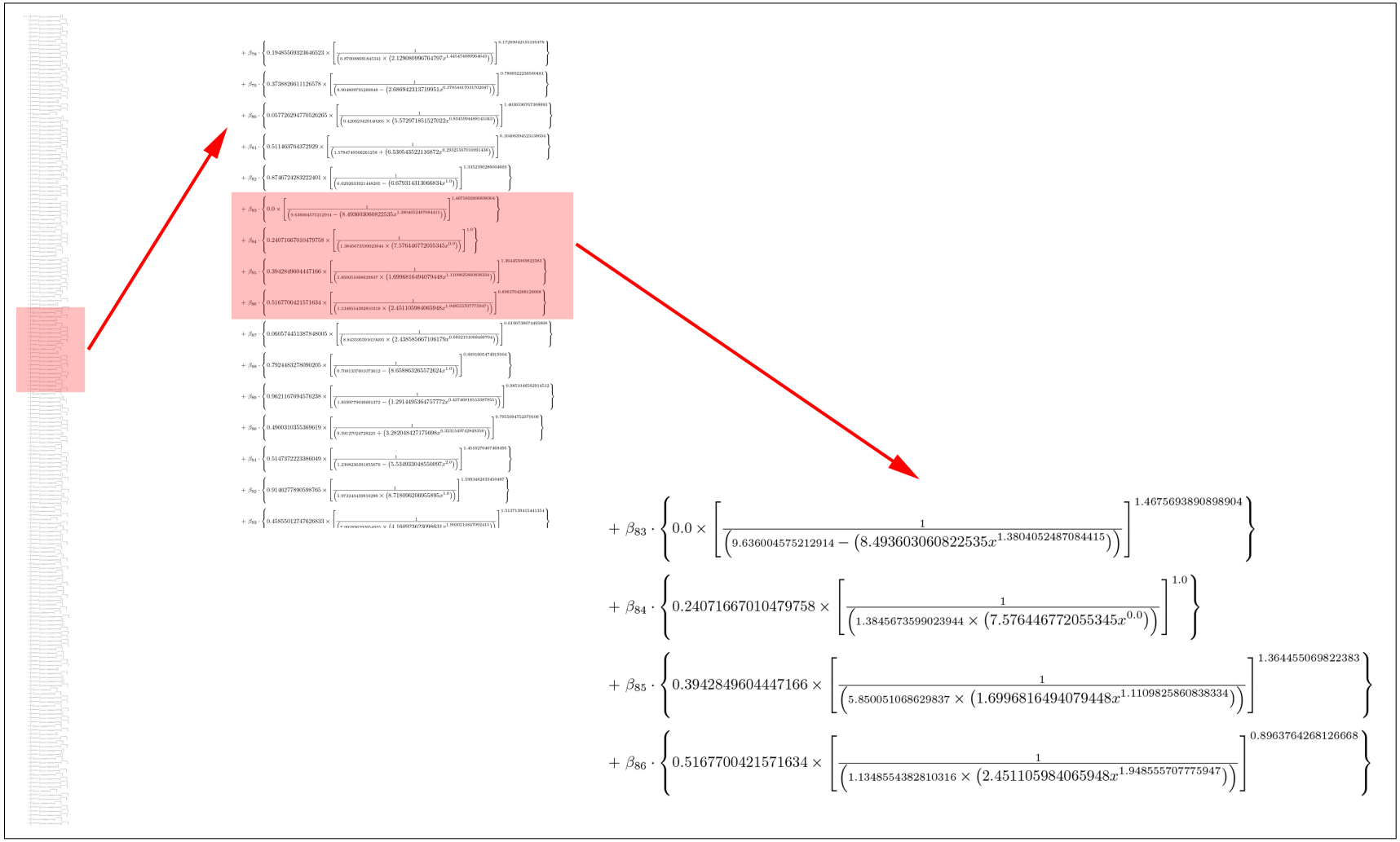
Dep. Variable:	y	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	8.964e+17
Date:	Fri, 11 Jan 2019	Prob (F-statistic):	0.00
Time:	05:47:27	Log-Likelihood:	1873.9
No. Observations:	101	AIC:	-3708.
Df Residuals:	81	BIC:	-3655.
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	2.899e-09	1.94e-09	1.496	0.139	-9.56e-10	6.76e-09
x2	-0.0002	1.36e-05	-17.732	0.000	-0.000	-0.000
x3	-10.6821	0.044	-244.871	0.000	-10.769	-10.595
x4	-6.918e-15	1e-14	-0.692	0.491	-2.68e-14	1.3e-14
x5	0.3165	0.000	790.368	0.000	0.316	0.317
x6	-5.939e-09	2.97e-09	-1.998	0.049	-1.19e-08	-2.38e-11
x7	0.1864	0.003	53.389	0.000	0.179	0.193
x8	0.1014	0.014	7.054	0.000	0.073	0.130
x9	3.344e-09	5.1e-09	0.656	0.514	-6.8e-09	1.35e-08
x10	4.222e-10	1.85e-09	0.228	0.820	-3.26e-09	4.11e-09
x11	0.0044	8.23e-05	53.390	0.000	0.004	0.005
x12	0.1074	0.002	53.388	0.000	0.103	0.111
x13	-0.0029	4.65e-05	-62.833	0.000	-0.003	-0.003
x14	-0.2941	0.019	-15.447	0.000	-0.332	-0.256
x15	-1.456e-09	2.33e-09	-0.624	0.534	-6.1e-09	3.18e-09
x16	-0.0217	0.000	-68.285	0.000	-0.022	-0.021
x17	-1.709e-08	6.65e-09	-2.569	0.012	-3.03e-08	-3.85e-09
const	-1.931e-13	4.65e-16	-415.150	0.000	-1.94e-13	-1.92e-13
x18	-1.209e-06	4.62e-07	-2.614	0.011	-2.13e-06	-2.89e-07
x19	-6.27e-10	3.56e-10	-1.759	0.082	-1.34e-09	8.2e-11
x20	6.033e-06	2.6e-07	23.227	0.000	5.52e-06	6.55e-06
x21	-25.8357	0.059	-435.988	0.000	-25.954	-25.718
x22	-1.457e-08	2.17e-09	-6.725	0.000	-1.89e-08	-1.03e-08

Omnibus:	5.895	Durbin-Watson:	0.578
Prob(Omnibus):	0.052	Jarque-Bera (JB):	8.852
Skew:	-0.128	Prob(JB):	0.0120
Kurtosis:	4.428	Cond. No.	1.39e+27

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.64e-44. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Figure 10.22: Approximating $1/x$ by ULR Using 22 Blocks



$$+ \beta_{83} \cdot \left\{ 0.0 \times \left[\frac{1}{\left(9.636004575212914 - \left(8.4936030608225 35 x^{1.3804052487084415} \right) \right)} \right]^{1.4675693890898904} \right\}$$

$$+ \beta_{84} \cdot \left\{ 0.24071667010479758 \times \left[\frac{1}{\left(1.3845673599023944 \times \left(7.576446772055345 x^{0.0} \right) \right)} \right]^{1.0} \right\}$$

$$+ \beta_{85} \cdot \left\{ 0.3942849604447166 \times \left[\frac{1}{\left(5.850051068629837 \times \left(1.6996816494079448 x^{1.1109825860838334} \right) \right)} \right]^{1.364455069822383} \right\}$$

$$+ \beta_{86} \cdot \left\{ 0.5167700421571634 \times \left[\frac{1}{\left(1.1348554382810316 \times \left(2.451105984065948 x^{1.948555707775947} \right) \right)} \right]^{0.8963764268126668} \right\}$$

Figure 10.23: Approximating $1/x$ by ULR Using 200 Blocks

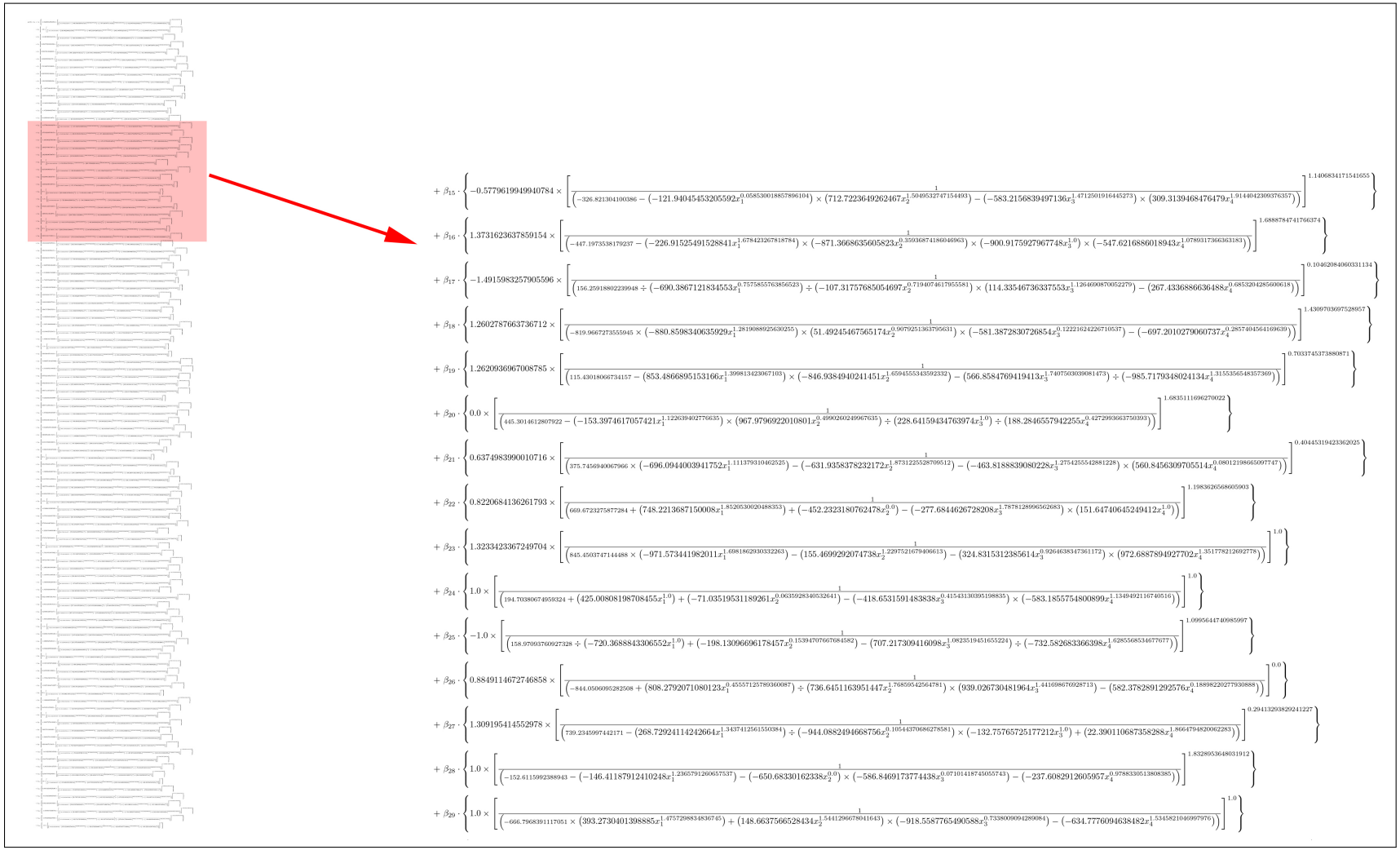


Figure 10.24: Regressing the 4-Dimensional Problem by ULR Using 110 Blocks

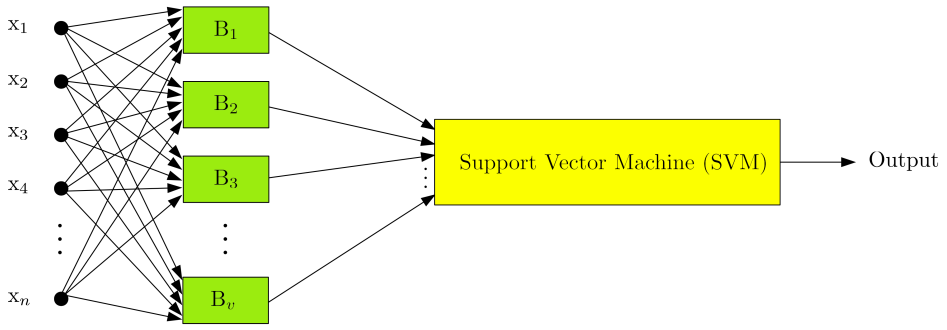


Figure 10.25: Support Function Machine (SFM)

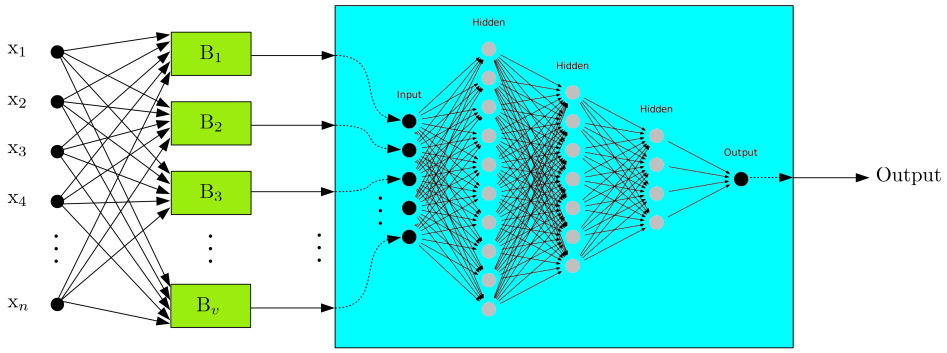


Figure 10.26: Artificial Mathematical Network (AMN)

10.3.3 Artificial Mathematical Network (AMN)

The same thing can be applied here where UTU is placed between the input layer and the first hidden layer. For this arrangement, UTU will act as a "**calibration layer**". The structure of this novel computing system is shown in Figure 10.26. Because the calibration layer contains v blocks instead of neurons and these blocks generate pure mathematical equations, so this hybrid computing system is called "**artificial mathematical network** (**AMN**)".

AMN has been tested with some datasets. The goal here is to reduce the total number of hidden layers and their associated neurons by explaining part of the nonlinearity of the dataset by the calibration layer.

There are other possible ways to hybridize UFO with ANNs. For example, (10.36) could be normalized and then used as a dynamic or controllable activation function.

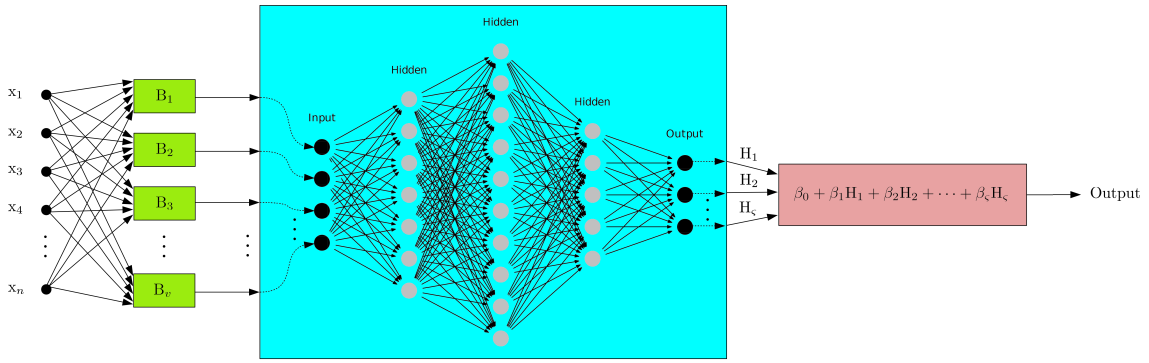


Figure 10.27: Mathematical Neural Regression (MNR)

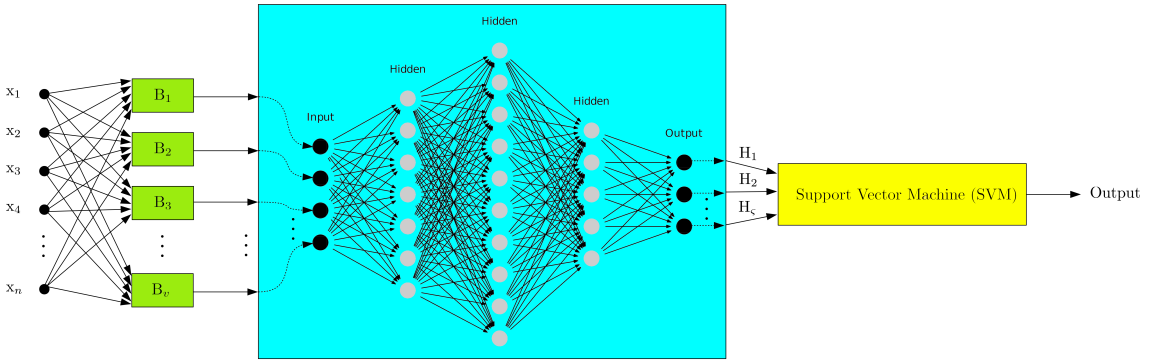


Figure 10.28: Mathematical Artificial Machine (MAM)

10.3.4 Mathematical Neural Regression (MNR)

As said earlier, LR is very fast and it can be used to fit any order polynomial curve by finding its optimal coefficients. Thus, LR can be embedded in AMN as a fine-tuning stage. If the last layer of ANN has more than one neuron, then MLR should be used instead of simple LR. The structure of this triple-hybrid computing system is shown in Figure 10.27. We call this system "**mathematical neural regression** (**MNR**)".

10.3.5 Mathematical Artificial Machine (MAM)

The same reason for replacing UTR with SFM, the regression unit of MNR can be replaced by SVR to have "**mathematical artificial machine** (**MAM**)". The structure of this advanced triple-hybrid computing system is shown in Figure 10.28.

10.3.6 Other Possible Hybridizations

The preceding subsections present some of **Al-Roomi's mathematical machines** (**AMMs**). By referring to Section 10.1, fifteen ML algorithms have been listed. Each one of them has its own strengths and weaknesses; and, of course, specific applications. We have just hybridized UFO with LR, SVM, and ANN. Thus, there are possibilities to hybridize UFO, specifically UTU, with many ML algorithms. For example, PCA could be placed before or after UTU to reduce the dimension. Also, UTU could be placed between the decision tree and its actual predictors. Furthermore, UTU could be added to symbolic regression techniques, such as the one reported in [369], so they can deal with one or two dense transformed predictors (i.e., $v = 1$ or 2) instead of n actual predictors.

10.4 Some Applications of UFO, UTR, SFM and AMN in Electric Power Systems

Based on Figure 10.18, our ML computing systems can be applied to many electric power system problems; which is impossible to include them all here. Instead, this section gives some applications of these computing systems in some selected branches of electric power systems engineering.

10.4.1 Predicting the Full-Load Power Output of a Combined-Cycle Power Plant Using UFO

For further validation, a standard benchmark regression dataset is taken from the University of California, Irvine's Machine Learning Repository, Center for Machine Learning and Intelligent Systems [117]. This multivariate dataset contains 9568 data points collected from a combined-cycle power plant (CCPP) over 6 years to predict the full-load electric power output (P_e) of the plant. The predictors or attributes here are: ambient temperature ($\breve{T}_a$), exhaust steam vacuum pressure ($\mathcal{P}_v$), relative humidity (RH), and atmospheric pressure ($\mathcal{P}_a$). Further information about this regression problem can be found in [117, 364].

Table 10.14 compares the results obtained by UFO against the results reported in [364] for different feature subset selections. As can be clearly seen from that

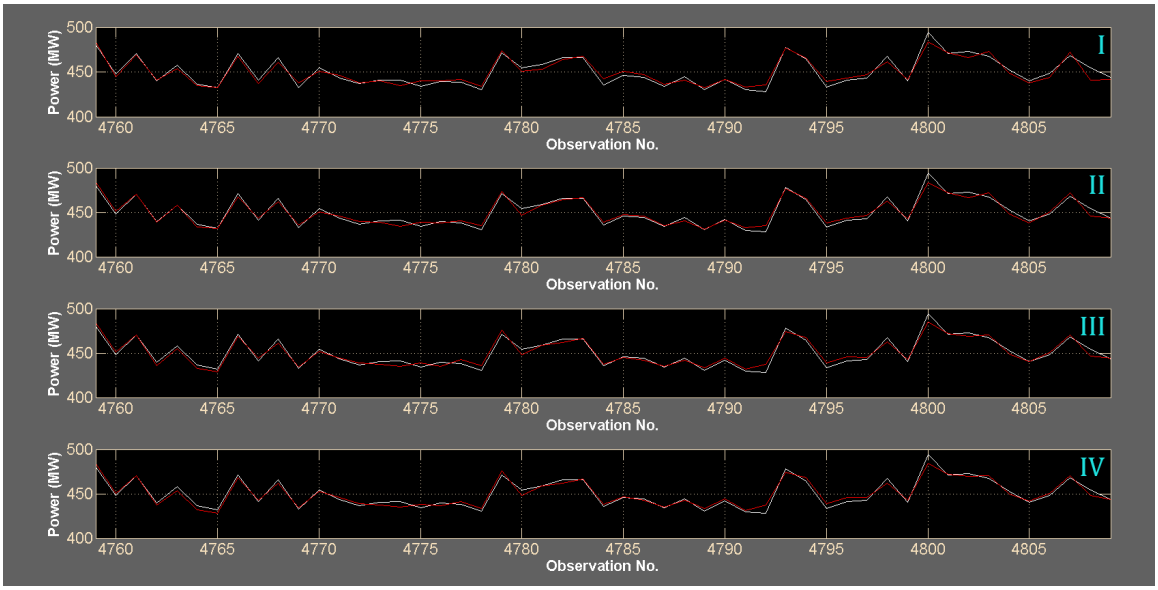


Figure 10.29: Graphical Representation of the Electric Power Predicted by UFO Using the Four Feature Subset Selections of $\{\breve{T}_a, \mathcal{P}_v, \mathcal{P}_a, \mathrm{RH}\}$

table, among 15 other ML techniques, UFO scores good position for each feature subset selection. More than that, the response $\hat{P}_e$ can be expressed in nice readable mathematical equations. These equations are shown in Table 10.15 for each feature subset selection. That is, both the explainability and prediction accuracy criteria can be achieved with UFO. Furthermore, the equations themselves can be used in other applications, such as forecasting and optimization fields.

Figure 10.29 shows the prediction accuracy of UFO for the electric power P_e based on the four feature subset selections of $\{\breve{T}_a, \mathcal{P}_v, \mathcal{P}_a, \mathrm{RH}\}$. The sample covered in that graph consists of 50 observations. It is taken from the center of the dataset, which is here located between observation no. 4759 and no. 4809. As can be obviously observed, the models have little drafts from the actual electric power reading.

10.4.2 Finding Fault Locations by SVM, ANN, ULR, SFM and AMN

This problem has been discussed in Chapter 9. It is about estimating the location of faults by utilizing the operating times of both end numerical directional overcurrent relays (DOCRs). The IEC/BS standard inverse time-current characteristic curve is used with a gradient of 3ϕ faults along the branch between bus 1 and bus 6 of the IEEE 8-bus test system shown in Figure 9.68. The original predictors are the

Table 10.14: The RMSE Performance of Different ML Techniques for the Third Regression Problem

ML Techniques		**Feature Subset Selections**[a]			
Category	Method[b]	I	II	III	IV
Functions	SLR	5.426	5.425	5.426	5.426
	LR	5.426	4.956	4.570	4.561
	LMS	5.433	4.968	4.580	4.572
	MLP	6.483	6.091	5.341	5.399
	RBF	7.501	8.757	8.695	8.487
	PR	5.426	4.956	4.570	4.561
	SMOReg	5.433	4.968	4.585	4.563
Lazy-Learning Algorithms	IBk	6.377	5.560	5.282	4.656
	K*	5.381	4.634	4.331	3.861
	LWL	8.005	7.915	8.211	8.221
Meta-Learning Algorithms	AR	5.933	5.545	5.549	5.556
	BREP	5.208	**4.026**	**3.922**	**3.779**
Rule-Based Algorithm	M5R	5.085	4.419	4.217	4.128
Tree-Based Learning Algorithms	M5P	5.086	4.359	4.178	4.087
	REP	5.229	4.339	4.291	4.211
Universal Functions Originator	UFO	**5.070**	4.567	4.306	4.270

[a] I: $\breve{T}_a$, II: $\breve{T}_a$-$\mathcal{P}_v$, III: $\breve{T}_a$-$\mathcal{P}_v$-RH, and IV: $\breve{T}_a$-$\mathcal{P}_v$-$\mathcal{P}_a$-RH.

[b] SLR: Simple Linear Regression, LR: Linear Regression, LMS: Least Median Square, MLP: Multi Layer Perceptron, RBF: Radial Basis Function Neural Network, PR: Pace Regression, SMOReg: Support Vector Poly Kernel Regression, IBk: Instance-Based/k-Nearest Neighbor Algorithm, K*: KStar Method, LWL: Locally Weighted Learning, AR: Additive Regression, BREP: Bagging Reduced-Error Pruning Tree, M5R: Model Trees Rules, M5P: Model Trees Regression, and REP: Reduced Error Pruning Trees.

Table 10.15: Regressing the Third Problem Using UFO with Two Blocks

Subset	**Mathematical Expression**	**RMSE**
I	$\hat{P}_e(\breve{T}_a) = 1.2788 \left[\sinh^{-1}\left(0.38209 + 660.6775\breve{T}_a^{-2}\right)\right]^{0.45657} \times 0.55771 \left(80.879 + 12.1642\breve{T}_a^{0.59478}\right)^{1.2479}$	5.070
II	$\hat{P}_e(\breve{T}_a, \mathcal{P}_v) = 1.0464 \left[\sinh^{-1}\left(0.14524 + 0.0057821\breve{T}_a^{1.6377} + 993.0493\mathcal{P}_v^{-2.5846}\right)\right]^{-0.090888}$	4.567
	$\times\ 1.1835 \left[\exp\left(-25.1299 + 354.3479\breve{T}_a^{0.0069709} + 243.9283\mathcal{P}_v^{-0.021615}\right)\right]^{0.010499}$	
III	$\hat{P}_e(\breve{T}_a, \mathcal{P}_v, \mathrm{RH}) = 2.0185 \left[\sinh^{-1}\left(188.3105 - 187.7175\breve{T}_a^{0.00025198} - 48.8783\mathcal{P}_v^{-2.4161} - 0.00016571\mathrm{RH}^{1.0539}\right)\right]^{1.6765}$	4.306
	$\times\ 1.8834 \left[\exp\left(242.8033 - 28.0241\breve{T}_a^{-0.29907} + (0.2798\mathcal{P}_v^{-0.5531})\left(37.5599\mathrm{RH}^{0.30605}\right)\right)\right]^{0.026446}$	
IV	$\hat{P}_e(\breve{T}_a, \mathcal{P}_v, \mathcal{P}_a, \mathrm{RH}) = 1.3376 \left[\sinh^{-1}\left(-54.9746 + 55.1402\breve{T}_a^{0.00222} - 0.00091085\mathcal{P}_v^{0.84602} + 34.6672\mathcal{P}_a^{-0.56583}\right.\right.$	**4.270**
	$\left.\left. - \ 23.3209\mathrm{RH}^{-2.2133}\right)\right]^{3.9572} \times 1.2706 \left[180.8542(1.8329\breve{T}_a^{-0.33369}) + 364.2025\mathcal{P}_v^{0.0064041}\right.$	
	$\left. - \ (30.4053\mathcal{P}_a^{-0.36839})(151.8699\mathrm{RH}^{0.0087123})\right]^{1.1532}$	

operating times of R_7 and R_{14}[5] and the target is the fault location x where the reference point is bus 1.

For the UFO sub-algorithm[6], the parameters of (10.36) have the following side constraints [37]: $w_j \in [-2, 2]$, $a_{0,j} \in [-1500, 1500]$, $a_{k,j} \in [-1500, 1500]$, $b_{k,j} \in [0, 2]$, $c_j \in [-2, 2]$, $\odot_{k,j} \in [+, -, \times, \div]$, and including all the standard analytical functions for f_j. Instead of BBO and DE, the basic random search algorithm (RSA) used in Chapter 5 is adopted here for ULR, SFM, and AMN. Thus, enough number of iterations is required. For that, 5000 iterations are used to explore the search space of UTU.

10.4.2.1 Universal Linear Regression (ULR)

As said before, UTR is a common name of two different computing systems; ULR and UNR. The structure shown in Figure 10.20 is adopted in this part of the experiment. For this mission, the software shown in Figure 10.21 is used. Using different groups of analytical functions, the location of any 3ϕ fault occurred on the faulty line of the IEEE 8-bus test system can be estimated by any one of the mathematical models listed in Table 10.16.

Similar to UFO, ULR has the ability to render its outputs as mathematical equations. However, the equation structures are different than that obtained by UFO. For example, the tuning stage of ULR works by employing LR to tune the model created by the building stage through $\{\beta_0, \beta_1, \cdots, \beta_v\}$. The external arithmetic operators between the v blocks are just "+" because the ±ve beta-coefficients are linear with the transformed predictors $\{\mathrm{B}_1, \mathrm{B}_2, \cdots, \mathrm{B}_\mathrm{v}\}$.

10.4.2.2 Support Function Machine (SFM)

For the SVM sub-algorithm, Python's scikit-learn library is used with the following hyperparameters:

- $d = 3$: the degree used with the '`poly`' kernel
- $\gamma =$'`auto`': the coefficient of the '`rbf`', '`poly`' and '`sigmoid`' kernels
- coef0 $= 0$: a term used with '`poly`' and '`sigmoid`' kernels

[5] i.e., T_7 and T_{14}.

[6] It is UTU, which is illustrated in Figure 10.19 and also called the calibration layer.

Table 10.16: Performance Evaluation of ULR in Locating 3ϕ Faults

Functions List	**MSE**	**Mathematical Model**
Basic	1.8899E-04	$\hat{x}\left(T_7, T_{14}\right) = 175.5477 + 4.4829 \cdot \left\{-1.9780 \log_2\left(778.6294\left(204.8029 T_7^{1.4664}\right)\left(1131.1296 T_{14}^{0.6994}\right)\right)\right\} + 1.209 \times 10^{11} \cdot \left\{1.2600\left[\frac{1}{1160.6968(369.6002 T_7)+1012.6474 T_{14}^{1.8653}}\right]^{1.6922}\right\}$
Default	2.1318E-05	$\hat{x}\left(T_7, T_{14}\right) = -77.1702 - 6.458 \times 10^{11} \cdot \left\{1.5623\left[\frac{1}{736.3006-\left(315.4300 T_7^{0.7710}\right)\left(815.6975 T_{14}^{0.0207}\right)}\right]^{1.8897}\right\} - 6.848 \times 10^{9} \cdot \left\{0.8964\left[\frac{1}{-733.0397+\left(1095.0451 T_7^{1.6152}\right)\left(1003.7380 T_{14}^{1.6374}\right)}\right]^{1.7957}\right\}$
Trigonometric	9.2483E-03	$\hat{x}\left(T_7, T_{14}\right) = 1362.0099 - 638.6165 \cdot \left\{2.0\left[\tan\left(418.8482 - \frac{57.7523}{693.0484 T_7^{1.5118}}\right)\right]^{0.2543}\right\} - 0.0074 \cdot \left\{0.4967\left[\tan\left(\frac{1158.6206}{384.8971 T_7^{0.0722}} + 988.4181 T_{14}^{1.0309}\right)\right]^{0.4960}\right\}$
Inverse Trigonometric	1.7488E-05	$\hat{x}\left(T_7, T_{14}\right) = -74.9112 + 1.724 \times 10^{5} \cdot \left\{0.9363 \times \left[\csc^{-1}\left(-79.4959 + \left(1202.1450 T_7^{1.7689}\right)\left(907.5649 T_{14}^{0.4859}\right)\right)\right]^{0.7807}\right\} + 9.453 \times 10^{8}\left[\csc^{-1}\left(81.0490\left(741.3958 T_7\right) + 248.1850 T_{14}^{1.2899}\right)\right]^{1.5136}$
Hyperbolic	1.1231E-02	$\hat{x}\left(T_7, T_{14}\right) = -5.109 \times 10^{7} + 4.949 \times 10^{7} \cdot \left\{1.0324\left[\operatorname{sech}\left(\frac{306.0937}{\left(802.3306 T_7^{0.8913}\right)(-466.4899 T_{14})}\right)\right]^{0.3797}\right\} + 4220.1868 \cdot \left\{2.0 \times \left[\operatorname{sech}\left(-25.2098 - \frac{435.3848 T_7^{1.5994}}{66.5691 T_{14}^{0.4897}}\right)\right]^{0.1527}\right\}$
Inverse Hyperbolic	**9.1240E-06**	$\hat{x}\left(T_7, T_{14}\right) = -117.1809 + 5494.9220 \cdot \left\{1.3839\left[\operatorname{csch}^{-1}(868.2539 + 1002.6594 T_7^{1.0334} + 691.1020 T_{14})\right]^{0.6307}\right\} - 1.198 \times 10^{8} \cdot \left\{1.1773\left[\operatorname{csch}^{-1}(-621.0659\left(1465.2040 T_7^{1.6285}\right) - 301.9881 T_{14}^{1.1051})\right]^{1.0953}\right\}$
Unfamiliar	1.7128E-03	$\hat{x}\left(T_7, T_{14}\right) = -4.588 \times 10^{4} - 1.841 \times 10^{6} \cdot \left\{1.4033\left[\operatorname{haversin}\left(\frac{93.0771}{\left(904.0293 T_7^{1.4330}\right)\left(1317.9183 T_{14}^{0.8701}\right)}\right)\right]^{0.7139}\right\} + 3.026 \times 10^{5} \cdot \left\{0.6055\left[\operatorname{hacoversin}\left(\frac{103.8404}{\left(151.5468 T_7^{1.5510}\right)\left(-1100.8869 T_{14}^{0.0842}\right)}\right)\right]^{2.0}\right\}$
All	5.8304E-05	$\hat{x}\left(T_7, T_{14}\right) = 6.656 \times 10^{6} - 3.825 \times 10^{6} \cdot \left\{1.2587\left[\coth^{-1}\left(1385.6859 T_{14}^{1.1507} - \frac{673.3372}{540.4530 T_7^{1.7669}}\right)\right]^{1.9051}\right\} + 2.053 \times 10^{6} \cdot \left\{1.8005\left[\tan^{-1}\left(-555.3394\left(265.3973 T_7^{1.6858}\right) - 529.7562\right)\right]^{1.3024}\right\}$

- $r_C = 1000$: the penalty parameter of the error term
- $\varepsilon = 0.1$: the epsilon in the epsilon-SVR model
- shrk =`True`: using the heuristic shrinking
- $\mathrm{E_{SVR}} = 0.001$: the acceptable tolerance to stop the internal SVM optimizer
- $G_{\mathrm{SVR}} = 10000$: the maximum number of iterations allowed to SVM

By referring to Figure 10.25, the original predictors are transformed by UTU as explained in Figure 10.19. Thus, UTU can also take care of the nonlinearity issue of the dataset before being sent to SVM. The results of SVM, before and after hybridizing it with UFO, are shown in Table 10.17. It can be seen from this table that the performance of SVM could be significantly enhanced by embedding UTU. For this particular engineering problem, this is true for all the standard kernel types.

10.4.2.3 Artificial Mathematical Network (AMN)

For the ANN sub-algorithm, Python's scikit-learn library is used with the following hyperparameters:

- No. of Hidden Layers: one
- No. of Neurons: 20
- Algorithm: Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS)
- Max. Iterations: 150

Similar to SFM, the universal transformation unit shown in Figure 10.19 can be used as a calibration layer. This layer can be inserted between the input layer and the first hidden layer as explained in Figure 10.26. The results of ANN, before and after hybridizing it with UFO, are shown in Table 10.18. From the preceding table, it is obvious that UTU could significantly enhance the performance of ANN. Again, for this particular engineering problem, this is true for all the four activation function types.

As said earlier, UFO could be hybridized with ML tools in different ways, and these hybrid computing systems have many applications in electric power systems.

Table 10.17: Performance Comparison of SVM and SFM in Locating 3ϕ Faults

Kernel	**SVM**		**SFM**	
Type	**MSE**	**Predictors**	**MSE**	**Predictors**
Linear	1.682112	T_7	0.047627	$\mathrm{B}_1 = \frac{2.0}{\left[\tan^{-1}\left(\frac{-1429.106630}{484.468383T_7^{1.857435}}+589.647323T_{14}^{0.563379}\right)\right]^{1.361769}}$
		T_{14}		$\mathrm{B}_2 = \frac{0.563619}{\left[\mathrm{csch}^{-1}\left(108.455201-\frac{1091.399180}{374.026931T_{14}^{1.193496}}\right)\right]^{0.686540}}$
Polynomial	13.647197	T_7	**4.196173E-03**	$\mathrm{B}_1 = \frac{-0.622320}{\left[\cot\left(-371.901338+\frac{998.502516T_7^{1.466942}}{1408.071868T_{14}^{0.086858}}\right)\right]^{1.357923}}$
		T_{14}		$\mathrm{B}_2 = \frac{1.590620}{\left[\sin\left(-1061.590080+\frac{810.197568T_7^{0.735244}}{705.355106}\right)\right]^{1.010872}}$
RBF	**0.113532**	T_7	4.869592E-03	$\mathrm{B}_1 = 1.599970\left[\mathrm{sech}\left(\frac{1264.850400}{-132.5272967T_7^{1.464794}}+1090.604637T_{14}^{0.531299}\right)\right]^{0.315255}$
		T_{14}		$\mathrm{B}_2 = -0.752684\tan\left(-1050.845742-\frac{594.275320T_7^{0.480742}}{852.688955T_{14}^{1.374335}}\right)$
Sigmoid	6.659083	T_7	0.232767	$\mathrm{B}_1 = \frac{-1.243809}{\left[\sqrt{\frac{16.092947\left(435.798002T_{14}^{0.849817}\right)}{855.705311T_7^{0.488689}}}\right]^{1.109851}}$
		T_{14}		$\mathrm{B}_2 = 0.566580\left(\mathrm{e}^{-600.672718-1042.7755967T_7^{1.636333}-404.603486T_{14}^{1,766309}}\right)^{1.039118}$

Table 10.18: Performance Comparison of ANN and AMN in Locating 3ϕ Faults

Activation	**ANN**		**AMN**	
Function	**MSE**	**Predictors**	**MSE**	**Predictors**
Identity	1.638117	T_7	9.612297E-03	$\mathrm{B}_1 = 0.518737 \left[\cot^{-1}\left(1173.452139 \times \frac{80.095814 T_7^{1.379890}}{-398.741561 T_{14}^{1.571631}}\right)\right]^{0.985347}$
		T_{14}		$\mathrm{B}_2 = \frac{-1.386326}{\left(8.792247 - \frac{325.427285 T_7^{5.833325E-05}}{1404.056577 T_{14}^{1.276365}}\right)^{1.028736}}$
Logistic	**8.306577E-04**	T_7	**3.961621E-05**	$\mathrm{B}_1 = -0.706883 \left[\mathrm{e}^{\frac{148.243617\left(60.602060 T_{14}^{1.560786}\right)}{757.145858 T_7^{0.434856}}}\right]^{0.106496}$
		T_{14}		$\mathrm{B}_2 = 1.492960 \left[\cos\left(-1430.858846 - 993.287021 T_7^{0.600390} + 795.849420 T_{14}^{0.166715}\right)\right]^{1.014171}$
TanH	2.146198E-03	T_7	9.957284E-05	$\mathrm{B}_1 = \frac{-0.957042}{\left[\mathrm{e}^{427.326875 + \left(443.329408 T_7^{1.035637}\right)\left(348.123054 T_{14}^{1.083100}\right)}\right]^{0.272274}}$
		T_{14}		$\mathrm{B}_2 = \frac{0.787860}{\left[\cot\left(1282.347128 + \frac{1484.642084}{506.971877 T_7^{0.367785}}\right)\right]^{1.448702}}$
ReLU	1.638117	T_7	2.781722E-03	$\mathrm{B}_1 = \frac{1.662760}{\left[\tan\left(481.604955 - \frac{11.859190 T_7^{0.370919}}{153.162177 T_{14}^{1.932047}}\right)\right]^{1.050553}}$
		T_{14}		$\mathrm{B}_2 = \frac{-1.231808}{\left[\csc\left(\frac{901.227196}{150.930152\left(225.084214 T_7^{1.679835}\right)}\right)\right]^{0.579001}}$

10.5 Incorporating a Global Constant Term into UFO

Including a constant term or bias to a model is very crucial. By referring to (10.36), the model generated by UFO can have a constant term for some structures of the building stage. For example:

- The external exponent c_j equals zero.
- All the internal exponents $\{b_{1,j}, b_{2,j}, \cdots, b_{n,j}\}$ equal zero with:
  - $f = 1 \times ()$ and $c_j \in [1, 2, 3, \cdots)$ or
  - $f = 1/()$ and $c_j \in (\cdots, -3, -2, -1]$.
- All the internal weights $\{a_{1,j}, a_{2,j}, \cdots, a_{n,j}\}$ equal zero and $\odot_{1,j} \in \{+, -\}$[7] with:
  - $f = 1 \times ()$ and $c_j \in [1, 2, 3, \cdots)$ or
  - $f = 1/()$ and $c_j \in (\cdots, -3, -2, -1]$.

which could be hard for some optimization algorithms, especially if they are not equipped with a round function to `round` their weights and exponents.

Also, having a constant term by the preceding strategy is impractical because all the terms of the jth block will be ineffective[8], and thus they increase the dimension and problem complexity for nothing. Add to that, if UFO is initiated with only one block (i.e., $v = 1$), then the preceding strategy works if and only if:

- $f = 1 \times ()$ and $c_j \in [1, 2, 3, \cdots)$ or
- $f = 1/()$ and $c_j \in (\cdots, -3, -2, -1]$.

which means that UFO is suppressed from doing many things, like exploring float exponents and various types of analytical functions.

To solve this problem permanently, a slight modification can be applied to Figure 10.7 to have something like the one shown in Figure 10.30. This modified structure can estimate the actual response as follows:

$$\hat{y}(X) = w_0 + g_1(X) \circledcirc_1 g_2(X) \circledcirc_2 \cdots \circledcirc_{v-1} g_v(X) \tag{10.73}$$

That is, the same original response estimated by (10.38) is used, but with adding a constant term w_0. The block B_0 can be called either the **initial block** or the **global**

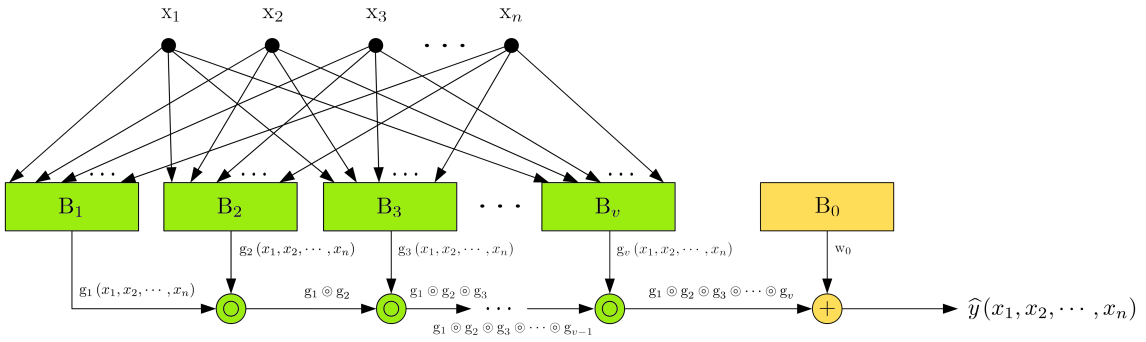


Figure 10.30: How to Incorporate a Constant Term in UFO with Only One O/P Stream and Universal Arithmetic Operators

block, and the scalar term w_0 inside it can be called either the **model bias** or the **global constant term** of UFO.

10.5.1 ULR is a Special Condition of UFO

Now, by comparing the structure given in Figure 10.30 and that given in Figure 10.20, a similarity can be made here. If the global constant term w_0 is incorporated, then UFO can act as ULR if all the external universal arithmetic operators are set as the addition operator; i.e. $\{\circledcirc_{\substack{j \\ j<v}} = + : \forall\ j = 1, 2, \cdots, v-1\}$.

This special arrangement will give us the opportunity to fine-tune the mathematical model of UFO as a **Least-Squares Regression**. It is exactly similar to LR, which can be solved quickly in the tuning stage. Thus, each g_j can be treated as a transformed predictor in UFO, which is exactly the mechanism described in Figure 10.19.

10.5.2 Approximating Many Nonlinear Benchmark Functions

In the preceding numerical analysis, we have done some function approximations using the conventional UFO shown in Figure 10.7. The special structure of UFO[9] described in Figure 10.30 is applied here to solve many nonlinear benchmark functions and then compared with SRs, ANNs, and SVMs.

[7] To avoid $a_{0,j}$ being multiplied or divided by zero.

[8] Because that scalar constant term will be represented by $\{w_j, a_{0,j}, \odot_{k,j}, a_{k,j}, b_{k,j}, c_j : \forall\ k \in [1, n]\}$, which is highly unrecommended.

[9] i.e., adding a global constant term to act as ULR.

10.5.2.1 Performance Comparison with ANNs and SVMs

In the first experiment, i.e. approximating $1/x$, the performance of UFO against other ML algorithms has been evaluated. The function used there is very simple, and it can be interpreted by many analytical functions defined in the pooling stage. That experiment aims to explore the features and capabilities of the UFO computing system and how it works. Thus, to have a fair comparison, a wide list of standard nonlinear test functions are used in this experiment. Table 10.19 gives some details about these test functions. The dataset of each test function contains 200 random points generated between the lower and upper bounds. In this experiment, ANN, SVM, and UFO are used with different configurations. For a fair comparison, only one computing machine is used in this experiment, which has the following specifications: ALIENWARE X51 Desktop, 64-bit Windows 10 OS, Intel Core i7-6700 CPU @ 3.4 GHz, and 8 GB RAM. Also, the three computing systems are initiated using the same number of function evaluations (NFE = 200). For ANN, one hidden layer of 20 neurons is used with **Adam optimizer**. The performance of ANN is evaluated using four different activation functions. Also, the performance of SVM is evaluated using four different kernels. For UFO, the performance is evaluated using five different settings of v (i.e., the number of blocks). The initial limits of $\{w, a_0, a, b, c\}$ are set as follows: $w \in [-1, 1]$, $a_0 \in [-1, 1]$, $a \in [-1, 1]$, $b \in [-1, 1]$, and $c \in [-2, 2]$. The internal optimizer of UFO can go beyond the limits. The variable f is defined with a set containing just identity, inverse, sine, and cosine analytical functions. Except for the division operator, the remaining three basic arithmetic operators ($+$, $-$, and $\times$) are defined in the pooling stage.

The error metric used in this experiment is MSE. The performance of ANN, SVM, and UFO on this metric are shown in Table 10.20 for each test function. As can be clearly seen, UFO is the winner in this competition. It can generate good results even with using just two blocks; i.e., $v = 2$. On the other side, it has been observed that the UFO computing system is slower than ANN and SVM. Table 10.21 shows the CPU time measured for each test function. There are many reasons for this phenomenon. Some of them are covered earlier in this chapter. Table 10.22 shows the total number of wins scored by UFO for each block.

Table 10.19: List of Some Standard Test Functions

No.	Function Name	Mathematical Expression	Domain
1	Dixon-Szegö's Function	$y_1 = 4x^2 - 4x^3 + x^4$	$-5 \leqslant x \leqslant 5$
2	Dixon's Function	$y_2 = x^4 - 3x^3 - 1.5x^2 + 10x$	$-5 \leqslant x \leqslant 5$
3	Rokne's Function	$y_3 = (x-1)^{10} = x^{10} - 10x^9 + 45x^8 - 120x^7 + 210x^6$ $-252x^5 + 210x^4 - 120x^3 + 45x^2 - 10x + 1$	$0 \leqslant x \leqslant 2$
4	Strongin's Function	$y_4 = 2 - \cos(x) - \cos(2x)$	$-1.5 \leqslant x \leqslant 4.5$
5	Suharev's Function	$y_5 = \sin\left(\frac{1}{x}\right)$	$0 \leqslant x \leqslant 1$
6	Marsden-Weinstein's Function	$y_6 = -2\cos(x) - \cos(2x)$	$-\frac{\pi}{2} \leqslant x \leqslant 2\pi$
7	Phillips' Function	$y_7 = -2\left(x-3\right)^2 - \mathrm{e}^{x^2/2}$	$-3 \leqslant x \leqslant 3$
8	Basso's Function	$y_8 = (-3x + 1.4)\sin(18x)$	$0 \leqslant x \leqslant 1.2$
9	Gramacy-Lee's Function	$y_9 = \frac{\sin(10\pi x)}{2x} + (x-1)^4$	$0.5 \leqslant x \leqslant 2.5$
10	Mineshaft Function No.01	$y_{10} = \cos(x) + \left\lvert(7-x)^{2/15}\right\rvert + 2\left\lvert(5-x)^{4/35}\right\rvert$	$0 \leqslant x \leqslant 10$
11	Mineshaft Function No.02	$y_{11} = \cos(x) - \mathrm{e}^{-1000(x-2)^2}$	$-10 \leqslant x \leqslant 10$
12	Timonov's Function No.01	$y_{12} = -\sin\left(x\right) - \sin\left(\frac{10}{3}x\right)$	$2.7 \leqslant x \leqslant 7.5$
13	Timonov's Function No.02	$y_{13} = -\sin\left(x\right) - \sin\left(\frac{10}{3}x\right) - \ln\left(x\right) + 0.84x - 3$	$3 \leqslant x \leqslant 7.5$
14	Timonov's Function No.03	$y_{14} = -\sin\left(x\right) - \sin\left(\frac{2}{3}x\right)$	$3.1 \leqslant x \leqslant 20.4$
15	Timonov's Function No.04	$y_{15} = \begin{cases} -\left(x-2\right)^2 & \text{if } x \leq 3 \\ -2\ln\left(x-2\right) - 1 & \text{otherwise} \end{cases}$	$0 \leqslant x \leqslant 6$
16	Lévy-Gomez's Function	$y_{16} = x - \sin(3x) + 1$	$0 \leqslant x \leqslant 6.5$
17	Cornelius-Lohner's Function	$y_{17} = (16x^2 - 24x + 5)\,\mathrm{e}^{-x}$	$1.9 \leqslant x \leqslant 3.9$
18	Bird-Like Function	$y_{18} = \frac{2x^4+x^2+2}{x^4+1}$	$-4 \leqslant x \leqslant 4$
19	Infinity or Csendes' Function	$y_{19} = \begin{cases} x^6\left[\sin\left(\frac{1}{x}\right) + 2\right] & \text{if } x \neq 0 \\ 0 & \text{otherwise} \end{cases}$	$-1 \leqslant x \leqslant 1$
20	Michalewicz's Function	$y_{20} = -\sin\left(x\right)\left[\sin\left(\frac{x^2}{\pi}\right)\right]^{20}$	$0 \leqslant x \leqslant \pi$
21	Salomon's Function	$y_{21} = 1 - \cos\left(2\pi\left\lVert x\right\rVert\right) + 0.1\left\lVert x\right\rVert \quad ; \quad \left\lVert x\right\rVert = \sqrt{x^2}$	$-100 \leqslant x \leqslant 100$
22	Vincent's Function	$y_{22} = -\sin\left[10\log\left(x\right)\right]$	$0.25 \leqslant x \leqslant 10$
23	Brown's Function	$y_{23} = \left(x_1^2\right)^{\left(x_2^2+1\right)} + \left(x_2^2\right)^{\left(x_1^2+1\right)}$	$-1 \leqslant x_1, x_2 \leqslant 4$
24	Brent's Function	$y_{24} = \left(x_1+10\right)^2 + \left(x_2+10\right)^2 + \mathrm{e}^{-x_1^2-x_2^2}$	$-10 \leqslant x_1, x_2 \leqslant 10$

Continued: List of Some Standard Test Functions

No.	Function Name	Mathematical Expression	Domain
25	Chichinadze's Function	$y_{25} = 11 - 12x_1 + x_1^2 + 10\cos\left(\frac{1}{2}\pi x_1\right) + 8\sin\left(\frac{5}{2}\pi x_1\right)$ $-0.2\frac{\sqrt{5}}{\mathrm{e}^{\frac{1}{2}\left(x_2-\frac{1}{2}\right)^2}}$	$-30 \leqslant x_1, x_2 \leqslant 30$
26	Davis' Function	$y_{26} = \left(x_1^2 + x_2^2\right)^{0.25}\left[\sin^2\left(50\left(3x_1^2 + x_2^2\right)^{0.1}\right) + 1\right]$	$-100 \leqslant x_1, x_2 \leqslant 100$
27	Easom's Function	$y_{27} = -\cos\left(x_1\right)\cos\left(x_2\right)\mathrm{e}^{\left[-(x_1-\pi)^2-(x_2-\pi)^2\right]}$	$-10 \leqslant x_1, x_2 \leqslant 10$
28	Giunta's Function	$y_{28} = 0.6 + \sum_{i=1}^{2}\left[\sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50}\sin\left(4 - \frac{64}{15}x_i\right)\right.$ $\left. - \sin\left(1 - \frac{16}{15}x_i\right)\right]$	$-1 \leqslant x_i \leqslant 1$
29	Hosaki's Function	$y_{29} = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right)x_2^2\mathrm{e}^{-x_2}$	$0 \leqslant x_1, x_2 \leqslant 10$
30	Keane's Function	$y_{30} = -\frac{\sin^2(x_1-x_2)\sin^2(x_1+x_2)}{\sqrt{x_1^2+x_2^2}}$	$0 \leqslant x_1, x_2 \leqslant 10$
31	F10: Leon's Function	$y_{31} = 100\left(x_2 - x_1^3\right)^2 + \left(1 - x_1\right)^2$	$-1.2 \leqslant x_1, x_2 \leqslant 1.2$
32	Matyas' Function	$y_{32} = 0.26\left(x_1^2 + x_2^2\right) - 0.48x_1x_2$	$-10 \leqslant x_1, x_2 \leqslant 10$
33	McCormick's Function	$y_{33} = -1.5x_1 + 2.5x_2 + \left(x_1 - x_2\right)^2 + \sin\left(x_1 + x_2\right) + 1$	$-1.5 \leqslant x_1 \leqslant 4$ $-3 \leqslant x_2 \leqslant 4$
34	Parsopoulos' Function	$y_{34} = \cos^2\left(x_1\right) + \sin^2\left(x_2\right)$	$-5 \leqslant x_1, x_2 \leqslant 5$
35	Treccani's Function	$y_{35} = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2$	$-5 \leqslant x_1, x_2 \leqslant 5$
36	Trefethen's Function	$y_{36} = \mathrm{e}^{\sin(50x_1)} + \sin\left(60\mathrm{e}^{x_2}\right) + \sin\left(70\sin\left(x_1\right)\right)$ $+\sin\left(\sin\left(80x_2\right)\right) - \sin\left(10\left(x_1 + x_2\right)\right) + \frac{1}{4}\left(x_1^2 + x_2^2\right)$	$-6.5 \leqslant x_1 \leqslant 6.5$ $-4.5 \leqslant x_2 \leqslant 4.5$
37	Tsoulos' Function	$y_{37} = x_1^2 + x_2^2 - \cos\left(18x_1\right) - \cos\left(18x_2\right)$	$-1 \leqslant x_1, x_2 \leqslant 1$
38	Zettl's Function	$y_{38} = \frac{1}{4}x_1 + \left(x_1^2 - 2x_1 + x_2^2\right)^2$	$-1 \leqslant x_1, x_2 \leqslant 5$
39	Banana Shape Function	$y_{39} = \frac{-100}{10[(x_1+1)^2-(x_2+1)^2]+x_1^2+4}$	$-1.5 \leqslant x_1 \leqslant 1.5$ $-2.5 \leqslant x_2 \leqslant 0.5$
40	Adjiman's Function	$y_{40} = \cos(x_1)\sin(x_2) - \frac{x_1}{x_2^2+1}$	$-5 \leqslant x_1, x_2 \leqslant 5$
41	Biggs' EXP2 Function	$y_{41} = \sum_{j=0}^{9}\left(\mathrm{e}^{-jx_1/10} - 5\mathrm{e}^{-jx_2/10} - \mathrm{e}^{-j/10} + 5\mathrm{e}^{-j}\right)^2$	$0 \leqslant x_1, x_2 \leqslant 20$
42	Branin's RCOS Function	$y_{42} = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	$-5 \leqslant x_1 \leqslant 10$ $0 \leqslant x_2 \leqslant 15$
43	Chen's Bird Function	$y_{43} = \frac{b}{b^2+\left(x_1^2+x_2^2-1\right)^2} + \frac{b}{b^2+\left(x_1^2+x_2^2-\frac{1}{2}\right)^2} + \frac{b}{b^2+(x_1-x_2)^2}$; $b = 0.001$	$-500 \leqslant x_1, x_2 \leqslant 500$
44	Chen's V Function	$y_{44} = \frac{b}{b^2+(x_1-0.4x_2-0.1)^2} + \frac{b}{b^2+(2x_1+x_2-1.5)^2}$; $b = 0.001$	$-500 \leqslant x_1, x_2 \leqslant 500$

Table 10.20: The MSE Performance Measured for 44 Different Test Functions Using ANN, SVM, and UFO Computing Systems

Test	Artificial Neural Network (ANN)				Support Vector Machine (SVM)				Universal Functions Originator (UFO)				
Function	Identity	Logistic	TanH	ReLU	RBF	Linear	Polynomial	Sigmoid	v = 2	v = 5	v = 10	v = 15	v = 20
y_1	8.94E+04	9.19E+04	9.12E+04	9.07E+04	7.43E+04	5.52E+04	9.01E+04	7.05E+04	**4.37E+02**	**1.05E+02**	**5.52E+00**	**7.95E-01**	**1.06E-02**
y_2	3.95E+04	4.06E+04	4.03E+04	3.99E+04	3.14E+04	3.32E+04	8.37E+04	3.52E+04	**8.73E+02**	**1.15E+02**	**6.53E+00**	**1.15E+00**	**4.12E-03**
y_3	2.61E-02	2.64E-02	2.63E-02	2.65E-02	1.07E-02	2.77E-02	2.66E-02	3.09E+01	**1.72E-03**	**1.39E-04**	**1.36E-06**	**1.41E-08**	**5.58E-10**
y_4	8.32E-01	1.29E+00	8.86E-01	1.05E+00	5.66E-03	8.52E-01	5.44E+01	1.53E+02	1.32E-01	2.93E-02	7.78E-03	**5.00E-04**	**1.17E-04**
y_5	2.68E-01	4.19E-01	2.79E-01	3.29E-01	2.19E-01	2.52E-01	3.59E-01	3.42E-01	**8.66E-04**	**9.86E-03**	**1.33E-03**	**1.14E-03**	**5.18E-04**
y_6	2.17E+00	2.10E+00	1.62E+00	1.66E+00	6.97E-03	3.30E+00	2.83E+00	2.51E+02	1.38E-01	5.57E-02	2.98E-02	1.30E-02	**2.62E-03**
y_7	2.12E+03	2.24E+03	2.24E+03	2.14E+03	2.58E+02	5.95E+02	9.77E+02	7.93E+02	**3.15E+01**	**3.07E+00**	**2.17E-01**	**2.36E-02**	**4.98E-03**
y_8	5.42E-01	5.75E-01	5.43E-01	5.51E-01	5.36E-01	5.45E-01	5.54E-01	5.63E-01	**3.13E-01**	**1.66E-01**	**5.11E-02**	**3.22E-02**	**4.41E-03**
y_9	8.83E-01	1.56E+00	1.01E+00	1.15E+00	1.39E-01	8.05E-01	1.31E+00	1.28E+02	**9.63E-02**	**6.74E-02**	**3.77E-02**	**3.24E-02**	**2.62E-02**
y_{10}	2.27E+00	2.50E+00	1.29E+00	1.94E+00	7.44E-03	3.96E-01	2.69E+03	3.69E+01	3.47E-02	1.47E-02	**5.06E-03**	**2.20E-03**	**1.80E-03**
y_{11}	5.28E-01	5.25E-01	4.74E-01	5.15E-01	7.40E-03	1.77E+01	1.29E+01	9.04E+01	2.77E-02	2.04E-02	**6.49E-03**	**6.91E-04**	**6.32E-04**
y_{12}	6.16E-01	5.12E-01	4.86E-01	5.48E-01	6.35E-03	4.93E-01	6.09E-01	5.28E-01	2.60E-01	7.55E-02	**8.01E-04**	**1.25E-05**	**5.97E-06**
y_{13}	5.46E-01	5.72E-01	4.37E-01	5.66E-01	5.97E-03	4.36E-01	7.05E+00	7.07E-01	2.57E-01	4.46E-02	**2.18E-04**	**6.06E-06**	**8.95E-07**
y_{14}	8.32E-01	8.36E-01	8.31E-01	8.44E-01	7.28E-03	1.65E+00	1.03E+00	8.49E-01	3.18E-01	8.53E-02	**5.85E-03**	**1.08E-04**	**8.47E-06**
y_{15}	1.07E+00	1.35E+00	1.15E+00	7.37E-01	5.00E-03	2.07E+00	2.01E+02	1.45E+02	3.00E-02	**2.10E-03**	**2.53E-04**	**5.31E-05**	**3.33E-05**
y_{16}	4.69E-01	1.02E+01	2.62E+00	2.96E+00	4.96E-03	5.08E-01	1.47E+01	1.50E+02	3.05E-01	2.36E-01	6.67E-02	1.01E-02	**8.13E-04**
y_{17}	2.95E-01	3.39E+00	2.12E-01	2.58E+00	5.62E-03	9.56E-02	1.35E-01	1.18E-01	**2.80E-06**	**1.32E-11**	**6.17E-15**	**3.15E-15**	**2.06E-15**
y_{18}	2.22E-01	8.86E-01	2.78E-01	1.49E+00	6.62E-03	2.53E-02	9.47E+00	2.97E+02	**5.68E-03**	**3.71E-03**	**1.16E-03**	**3.49E-04**	**4.50E-05**
y_{19}	3.25E-01	3.66E-01	3.17E-01	2.99E-01	3.95E-02	4.16E-01	3.37E-01	2.35E+00	**5.58E-03**	**1.45E-03**	**6.96E-05**	**1.22E-06**	**2.12E-08**
y_{20}	4.95E-02	5.17E-02	4.82E-02	4.52E-02	1.74E-02	5.32E-02	2.13E-01	1.24E+02	2.02E-02	**1.31E-02**	**2.81E-03**	**1.18E-03**	**3.32E-04**
y_{21}	2.73E+01	2.85E+01	2.47E+01	7.99E-01	1.83E+00	7.87E+01	4.83E+01	9.97E+00	**4.58E-01**	**4.39E-01**	**4.15E-01**	**3.94E-01**	**3.79E-01**
y_{22}	4.98E-01	4.98E-01	5.08E-01	5.01E-01	8.89E-02	1.73E+00	1.35E+00	2.31E+01	3.45E-01	2.54E-01	1.26E-01	**6.45E-02**	**1.65E-02**
y_{23}	1.10E+03	1.17E+03	1.16E+03	1.17E+03	3.12E+02	5.11E+02	8.49E+02	5.43E+02	**3.98E+01**	**2.27E+01**	**1.00E+01**	**3.14E+00**	**1.74E+00**
y_{24}	1.03E+05	1.07E+05	1.07E+05	1.03E+05	2.73E+04	1.84E+03	2.83E+04	1.02E+04	**1.30E+03**	**5.80E+02**	**4.78E+02**	**1.90E+02**	**5.71E+01**
y_{25}	2.00E+05	2.20E+05	2.20E+05	1.97E+05	1.25E+05	7.33E+04	1.05E+05	1.06E+05	**2.76E+03**	**7.69E+02**	**1.32E+03**	**2.36E+02**	**2.17E+02**
y_{26}	1.36E+02	1.17E+02	1.53E+02	1.29E+01	1.13E+01	6.48E+01	1.23E+02	1.81E+03	**1.10E+01**	**9.56E+00**	**8.74E+00**	**8.40E+00**	**7.93E+00**
y_{27}	6.09E-04	1.13E-03	2.01E-02	3.15E-02	9.86E-03	1.04E-02	1.85E-02	1.61E+03	**1.39E-04**	**1.25E-04**	**1.17E-04**	**1.08E-04**	**7.11E-05**
y_{28}	1.40E-02	1.55E-02	1.40E-02	8.09E-03	3.63E-03	1.47E-02	1.15E-02	1.61E-01	6.31E-03	4.85E-03	**3.20E-03**	**2.12E-03**	**1.21E-03**
y_{29}	2.39E+03	2.82E+03	2.74E+03	2.67E+03	2.51E+03	2.39E+03	3.38E+04	2.97E+03	**9.76E+02**	**6.99E+02**	**5.22E+02**	**3.97E+02**	**2.82E+02**
y_{30}	9.04E-02	3.52E-03	1.29E-02	2.59E-02	6.38E-03	4.40E-03	6.87E-02	3.67E+00	**1.90E-03**	**1.78E-03**	**1.67E-03**	**1.51E-03**	**1.47E-03**
y_{31}	2.04E+04	2.05E+04	2.05E+04	2.05E+04	1.27E+04	1.41E+04	1.42E+04	1.42E+04	**1.99E+03**	**1.19E+03**	**1.02E+03**	**7.37E+02**	**5.98E+02**
y_{32}	4.93E+02	4.99E+02	5.18E+02	2.91E+02	2.85E+02	4.78E+02	6.88E+02	1.79E+03	**1.29E+01**	**5.46E+00**	**3.36E+00**	**2.20E+00**	**1.61E+00**
y_{33}	6.96E+01	7.85E+01	8.05E+01	6.58E+01	1.17E+01	7.33E+01	3.57E+01	9.02E+02	**5.07E+00**	**1.62E+00**	**1.36E+00**	**6.52E-01**	**7.16E-01**
y_{34}	2.25E-01	2.24E-01	2.46E-01	2.52E-01	**6.53E-03**	3.83E-01	1.04E+00	1.28E+03	1.83E-01	1.71E-01	1.39E-01	1.13E-01	1.10E-01
y_{35}	1.13E+05	1.15E+05	1.15E+05	1.14E+05	1.00E+05	7.61E+04	6.58E+04	9.33E+04	**3.79E+03**	**6.56E+03**	**2.27E+03**	**1.48E+03**	**9.19E+02**
y_{36}	2.74E+01	2.85E+01	3.08E+01	1.43E+01	2.63E+00	2.34E+01	4.62E+01	9.08E+02	3.87E+00	2.83E+00	**2.46E+00**	**2.32E+00**	**2.16E+00**
y_{37}	1.23E+00	1.23E+00	1.23E+00	1.07E+00	9.45E-01	1.25E+00	1.26E+00	1.27E+00	**9.42E-01**	**9.08E-01**	**8.16E-01**	**7.63E-01**	**7.13E-01**
y_{38}	7.12E+04	7.38E+04	7.34E+04	7.29E+04	4.60E+04	2.19E+04	8.49E+03	5.07E+04	**6.71E+03**	**5.28E+03**	**2.29E+03**	**8.28E+02**	**8.53E+02**
y_{39}	5.87E+01	6.54E+01	6.24E+01	6.92E+01	1.42E+01	3.25E+01	3.70E+01	1.06E+02	1.56E+01	**1.26E+01**	**4.59E+00**	**4.19E+00**	**4.12E+00**
y_{40}	1.09E+00	1.08E+00	6.52E-01	8.47E-01	**3.04E-02**	3.71E+00	2.98E+00	1.28E+03	7.24E-01	5.21E-01	3.04E-01	2.70E-01	2.10E-01
y_{41}	1.57E+03	1.81E+03	1.76E+03	1.75E+03	1.73E+03	1.24E+03	3.33E+05	1.84E+03	**1.46E+01**	**4.90E+00**	**1.04E+00**	**1.58E-01**	**1.32E-02**
y_{42}	4.00E+03	5.25E+03	5.27E+03	4.39E+03	2.45E+03	2.26E+03	3.08E+04	3.31E+03	**4.86E+02**	**2.70E+02**	**2.02E+02**	**1.41E+02**	**7.18E+01**
y_{43}	7.76E-02	6.71E-04	8.75E-02	4.60E+01	1.38E-08	1.38E-08	1.38E-08	1.38E-08	**2.25E-10**	**2.06E-10**	**1.36E-10**	**8.62E-11**	**5.91E-11**
y_{44}	5.79E-01	4.91E-04	7.74E-02	6.78E+01	1.77E-09	1.77E-09	1.77E-09	1.77E-09	**5.99E-12**	**3.18E-12**	**2.89E-12**	**2.42E-12**	**2.46E-12**

10.5.2.2 Performance Comparison with SRs

In this experiment, the UFO performance is compared with some symbolic regression techniques presented in the literature. For this mission, the ten test functions listed in Table 10.23 are used. The full details about these functions are given in [369] where six genetic programming (GP) algorithms are used with different configurations. These six GP algorithms are called: 1. Standard Crossover (SC), 2. No Same Mate (NSM), 3. Semantics Aware Crossover (SAC), 4. Context Aware Crossover (CAC), 5. Soft Brood Selection (SBS), 6. Semantic Similarity-based Crossover (SSC). The number of generations is set to 50 and the population size is set to 200 (for CAC and SBS) and 500 (for SC, NSM, SAC, and SSC). To be on the safe side, the number of function evaluations (NFE) used in UFO is set to 200 for the first eight functions and to 1000 for the last two functions. Note that, the lowest NFE used in GP is not less than 10,000. Also, the same arithmetic operators $\{+, -, \times, \div\}$ and analytical functions $\{1, \exp, \log, \sin, \cos\}$ are used in UFO. Furthermore, the same computing machine used in the last experiment is also used here.

By using the same error metric adopted in [369], Table 10.24 shows how many successful hits achieved by UFO among 100 independent runs for each function. It can be seen that UFO has a very good performance in approximating these functions. Because the result given in [369] does not show the CPU time and the computing machine specifications, so the CPU performance cannot be compared. However, for a reference purpose, Table 10.25 shows the average CPU time recorded for each test function. It can be seen that the last two functions consume a huge amount of CPU time compared with the first eight functions. Apparently, someone might think that the reason comes from the higher dimension and NFE used, which is not entirely correct. Rather, the main reason comes from employing logarithmic function log, which is unsuitable for the test functions y_9 and y_{10}. This analytical function could create some infeasible points, and thus an extra CPU time is required to let the constraint-handling sub-algorithm to satisfy this infeasibility issue. This phenomenon has been discussed in Subsection 10.2.1. This claim can be proved by looking at the CPU time of y_{23} to y_{44} tabulated in Table 10.21. To make it very clear in this experiment, the logarithmic function log is removed from the pool and the last two

Table 10.21: The CPU Performance Measured for 44 Different Test Functions Using ANN, SVM, and UFO Computing Systems

Test	Artificial Neural Network (ANN)				Support Vector Machine (SVM)				Universal Functions Originator (UFO)				
Function	Identity	Logistic	TanH	ReLU	RBF	Linear	Polynomial	Sigmoid	v = 2	v = 5	v = 10	v = 15	v = 20
y_1	5.78E-02	5.88E-02	1.05E-01	5.88E-02	2.99E-03	9.97E-04	2.00E-03	6.98E-03	1.95E-01	3.30E-01	5.62E-01	7.90E-01	1.03E+00
y_2	5.78E-02	6.08E-02	1.15E-01	7.28E-02	3.00E-03	1.99E-03	2.00E-03	5.99E-03	1.95E-01	3.25E-01	5.64E-01	7.83E-01	1.02E+00
y_3	4.79E-02	6.18E-02	1.02E-01	5.88E-02	9.97E-04	1.00E-03	2.01E-03	6.98E-03	1.94E-01	3.26E-01	5.58E-01	7.74E-01	1.02E+00
y_4	4.79E-02	6.08E-02	1.05E-01	5.88E-02	9.98E-04	3.00E-03	2.00E-03	7.99E-03	1.97E-01	3.30E-01	5.60E-01	7.80E-01	1.02E+00
y_5	4.89E-02	5.98E-02	1.17E-01	5.88E-02	2.00E-03	1.99E-03	1.99E-03	6.99E-03	1.97E-01	3.24E-01	5.55E-01	7.76E-01	1.01E+00
y_6	1.10E-02	5.98E-02	1.01E-01	5.88E-02	9.91E-04	2.00E-03	2.00E-03	6.97E-03	1.96E-01	3.28E-01	5.54E-01	7.76E-01	1.02E+00
y_7	4.99E-02	6.08E-02	1.07E-01	6.68E-02	2.00E-03	1.99E-03	2.01E-03	6.98E-03	1.99E-01	3.31E-01	5.57E-01	7.74E-01	1.02E+00
y_8	4.79E-02	5.88E-02	9.97E-02	5.78E-02	2.01E-03	9.97E-04	1.99E-03	5.97E-03	1.94E-01	3.22E-01	5.51E-01	7.69E-01	1.02E+00
y_9	4.78E-02	5.98E-02	1.23E-01	5.88E-02	1.99E-03	1.99E-03	2.00E-03	6.96E-03	1.93E-01	3.24E-01	5.49E-01	7.69E-01	1.01E+00
y_{10}	4.79E-02	7.48E-02	1.02E-01	7.18E-02	9.97E-04	9.97E-04	9.97E-04	4.99E-03	1.93E-01	3.24E-01	5.59E-01	7.71E-01	1.01E+00
y_{11}	1.30E-02	5.88E-02	1.02E-01	5.88E-02	2.00E-03	9.98E-04	9.97E-04	6.00E-03	1.98E-01	3.31E-01	5.66E-01	7.93E-01	1.03E+00
y_{12}	6.01E-03	5.88E-02	2.49E-02	5.88E-02	9.97E-04	9.77E-04	9.97E-04	4.99E-03	1.92E-01	3.20E-01	5.51E-01	7.65E-01	1.01E+00
y_{13}	5.98E-03	6.38E-02	1.01E-01	5.78E-02	9.96E-04	1.99E-03	9.98E-04	4.99E-03	1.95E-01	3.25E-01	5.48E-01	7.71E-01	1.01E+00
y_{14}	4.99E-02	5.98E-02	1.08E-01	2.49E-02	1.99E-03	9.97E-04	9.81E-04	3.99E-03	1.92E-01	3.27E-01	5.47E-01	7.67E-01	1.00E+00
y_{15}	1.70E-02	6.08E-02	1.09E-01	6.08E-02	1.99E-03	9.96E-04	9.97E-04	6.00E-03	1.96E-01	3.26E-01	5.53E-01	7.70E-01	1.01E+00
y_{16}	5.58E-02	5.98E-02	1.04E-01	6.78E-02	9.97E-04	1.99E-03	1.99E-03	5.98E-03	1.92E-01	3.24E-01	5.53E-01	7.68E-01	1.04E+00
y_{17}	4.79E-02	6.08E-02	1.03E-01	5.98E-02	9.98E-04	2.00E-03	9.97E-04	5.99E-03	1.93E-01	3.25E-01	5.46E-01	7.65E-01	9.96E-01
y_{18}	4.69E-02	6.38E-02	1.10E-01	5.98E-02	1.00E-03	1.00E-03	9.98E-04	5.99E-03	1.95E-01	3.29E-01	5.63E-01	7.78E-01	1.02E+00
y_{19}	3.69E-02	6.18E-02	1.06E-01	5.98E-02	9.91E-04	1.98E-03	9.97E-04	6.98E-03	1.93E-01	3.27E-01	5.51E-01	7.74E-01	1.01E+00
y_{20}	2.19E-02	5.98E-02	1.04E-01	6.08E-02	9.97E-04	1.99E-03	9.98E-04	6.98E-03	1.93E-01	3.24E-01	5.43E-01	7.73E-01	1.01E+00
y_{21}	4.79E-02	6.18E-02	1.01E-01	6.18E-02	3.99E-03	9.98E-04	9.97E-04	4.99E-03	1.97E-01	3.31E-01	5.47E-01	7.71E-01	1.01E+00
y_{22}	2.79E-02	5.98E-02	1.60E-02	2.49E-02	2.01E-03	1.99E-03	1.99E-03	1.20E-02	2.04E-01	3.26E-01	5.44E-01	7.68E-01	1.00E+00
y_{23}	4.99E-02	6.28E-02	1.05E-01	6.48E-02	1.99E-03	9.92E-04	1.99E-03	6.98E-03	2.12E-01	3.65E-01	6.28E-01	8.96E-01	1.17E+00
y_{24}	4.99E-02	6.28E-02	1.05E-01	6.48E-02	4.00E-03	1.99E-03	9.97E-04	5.98E-03	2.09E-01	3.69E-01	6.33E-01	9.03E-01	1.18E+00
y_{25}	5.98E-02	6.08E-02	1.07E-01	6.28E-02	6.98E-03	1.99E-03	9.97E-04	4.99E-03	2.09E-01	3.69E-01	6.33E-01	9.03E-01	1.17E+00
y_{26}	5.09E-02	6.28E-02	9.87E-02	6.48E-02	4.00E-03	9.96E-04	9.98E-04	4.98E-03	2.09E-01	3.83E-01	6.33E-01	8.99E-01	1.17E+00
y_{27}	4.99E-02	6.19E-02	1.17E-01	6.38E-02	3.99E-03	9.98E-04	9.82E-04	6.99E-03	2.09E-01	3.77E-01	6.37E-01	8.98E-01	1.17E+00
y_{28}	5.68E-02	5.98E-03	1.05E-01	6.58E-02	2.01E-03	2.01E-03	2.00E-03	4.99E-03	2.09E-01	3.67E-01	6.33E-01	8.99E-01	1.18E+00
y_{29}	6.78E-02	6.18E-02	1.07E-01	6.58E-02	1.99E-03	1.99E-03	2.00E-03	6.00E-03	2.07E-01	3.67E-01	6.26E-01	8.92E-01	1.16E+00
y_{30}	4.99E-02	6.48E-02	1.09E-01	6.48E-02	9.98E-04	9.97E-04	2.00E-03	3.00E-03	2.08E-01	3.65E-01	6.30E-01	8.91E-01	1.17E+00
y_{31}	4.99E-02	6.38E-02	1.04E-01	6.58E-02	2.99E-03	1.99E-03	9.98E-04	5.99E-03	2.10E-01	3.66E-01	6.32E-01	9.05E-01	1.18E+00
y_{32}	4.99E-02	8.78E-02	1.07E-01	6.38E-02	3.99E-03	1.99E-03	1.99E-03	6.01E-03	2.08E-01	3.71E-01	6.33E-01	9.02E-01	1.18E+00
y_{33}	4.99E-02	6.18E-02	1.06E-01	6.48E-02	1.98E-03	9.97E-04	1.99E-03	7.00E-03	2.08E-01	3.67E-01	6.33E-01	8.95E-01	1.18E+00
y_{34}	4.99E-02	6.38E-02	1.04E-01	6.48E-02	2.01E-03	1.00E-03	9.98E-04	6.01E-03	2.10E-01	3.70E-01	6.32E-01	9.05E-01	1.18E+00
y_{35}	4.99E-02	6.48E-02	1.11E-01	6.48E-02	2.99E-03	1.99E-03	1.99E-03	7.98E-03	2.13E-01	3.81E-01	6.36E-01	9.02E-01	1.18E+00
y_{36}	4.99E-02	6.18E-02	1.06E-01	6.58E-02	2.99E-03	2.99E-03	1.01E-03	8.00E-03	2.12E-01	3.66E-01	6.36E-01	9.03E-01	1.18E+00
y_{37}	5.29E-02	6.78E-02	1.04E-01	6.98E-02	2.99E-03	1.99E-03	2.00E-03	6.98E-03	2.09E-01	3.70E-01	6.36E-01	9.10E-01	1.23E+00
y_{38}	5.19E-02	6.38E-02	1.05E-01	6.78E-02	2.00E-03	1.99E-03	2.00E-03	6.98E-03	2.13E-01	3.68E-01	6.31E-01	9.01E-01	1.22E+00
y_{39}	5.09E-02	6.28E-02	1.07E-01	6.48E-02	1.99E-03	1.99E-03	2.98E-03	7.99E-03	2.12E-01	3.68E-01	6.30E-01	8.97E-01	1.17E+00
y_{40}	5.09E-02	6.28E-02	1.20E-01	6.58E-02	2.00E-03	1.99E-03	9.96E-04	6.98E-03	2.09E-01	3.66E-01	6.30E-01	9.01E-01	1.18E+00
y_{41}	5.08E-02	6.78E-02	1.10E-01	6.58E-02	5.98E-03	1.99E-03	9.98E-04	3.99E-03	2.07E-01	3.62E-01	6.27E-01	8.94E-01	1.16E+00
y_{42}	5.19E-02	6.38E-02	1.18E-01	6.38E-02	2.99E-03	2.00E-03	2.00E-03	5.99E-03	2.11E-01	3.70E-01	6.28E-01	8.96E-01	1.17E+00
y_{43}	5.59E-02	6.18E-02	8.78E-02	6.38E-02	9.98E-04	9.97E-04	9.97E-04	9.98E-04	2.11E-01	3.71E-01	6.38E-01	9.17E-01	1.20E+00
y_{44}	5.09E-02	6.28E-02	1.01E-01	6.38E-02	9.97E-04	9.99E-04	9.96E-04	9.98E-04	2.10E-01	3.76E-01	6.30E-01	9.02E-01	1.17E+00

Table 10.22: Total Wins Scored by UFO Against ANN and SVM for Different Blocks

Number of Blcoks (v)	2	5	10	15	20
Total Wins (Out of 44)	28	31	38	40	**41**

Table 10.23: Symbolic Regression Benchmark Functions

Function	**Fitcase**
$y_1 = x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
$y_2 = x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
$y_3 = x^5 + x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
$y_4 = x^6 + x^5 + x^4 + x^3 + x^2 + x$	20 random points $\subseteq [-1, 1]$
$y_5 = \sin(x^2)\cos(x) - 1$	20 random points $\subseteq [-1, 1]$
$y_6 = \sin(x) + \sin(x + x^2)$	20 random points $\subseteq [-1, 1]$
$y_7 = \log(x + 1) + \log(x^2 + 1)$	20 random points $\subseteq [-1, 1]$
$y_8 = \sqrt{x}$	20 random points $\subseteq [-1, 1]$
$y_9 = \sin(x_1) + \sin(x_2^2)$	100 random points $\subseteq [-1, 1] \times [-1, 1]$
$y_{10} = 2\sin(x_1)\cos(x_2)$	100 random points $\subseteq [-1, 1] \times [-1, 1]$

functions given in Table 10.23 are evaluated again. Given 100 independent trials or runs, the average CPU time recorded for y_9 is 4.31126 seconds and for y_{10} is 4.27870 seconds. More ideas about how to accelerate the CPU time of UFO are given earlier.

10.5.3 Applying Piecewise Regression in UFO

In LR, the nonlinearity issue could be tackled by using what is called **piecewise regression**[10]. This concept could be applied to UFO to express the response as a piecewise linear/nonlinear equation. That is, UFO could be used to express the response of each segment mathematically by using multiple blocks, analytical functions,

[10]It is also called **broken-stick regression** and **segmented regression**. It is briefly discussed in Chapter 9/Section 9.4.

Table 10.24: Number of Successful Runs Out of 100 Runs Scored by SRs and UFO

Method	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
SC	48	22	7	4	20	35	35	16	7	18
NSM	48	16	4	4	19	36	40	28	4	17
SAC2	53	25	7	4	17	32	25	13	4	4
SAC3	56	19	6	2	21	23	25	12	3	8
SAC4	53	17	11	1	20	23	29	14	3	8
SAC5	53	17	11	1	19	27	30	12	3	8
CAC1	34	19	7	7	12	22	25	9	1	15
CAC2	34	20	7	7	13	23	25	9	2	16
CAC4	35	22	7	8	12	22	26	10	3	16
SBS31	43	15	9	6	31	28	31	17	13	33
SBS32	42	26	7	8	36	27	44	30	17	27
SBS34	51	21	10	9	34	33	46	25	26	33
SBS41	41	22	9	5	31	34	38	25	19	33
SBS42	50	22	17	10	41	32	51	24	24	33
SBS44	40	25	16	9	35	43	42	27	33	34
SSC8	66	28	22	10	48	56	59	21	25	47
SSC12	67	33	14	12	47	47	66	38	37	51
SSC16	55	39	20	11	46	44	67	29	30	59
SSC20	58	27	10	9	52	48	63	26	39	51
UFO	**100**	**100**	**100**	**97**	**100**	**100**	**100**	**100**	**100**	**100**

and arithmetic operators. The number of segments and break-points could be pre-defined before running UFO. Alternatively, the optimal values of these two settings could be determined heuristically. These are just some possible ideas that could be tried in the future.

Table 10.25: Best, Worst, Mean, Median, and Standard Deviation of Fitness and CPU Time Scored by UFO Against SRs

Fitness	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_3}$	$\mathbf{y_4}$	$\mathbf{y_5}$	$\mathbf{y_6}$	$\mathbf{y_7}$	$\mathbf{y_8}$	$\mathbf{y_9}$	$\mathbf{y_{10}}$
Best	2.21E-04	4.30E-04	2.03E-04	7.91E-04	1.66E-04	8.46E-05	3.22E-07	6.08E-11	6.08E-03	1.31E-03
Worst	6.94E-03	3.34E-03	2.25E-03	1.57E-02	4.87E-03	1.71E-03	1.02E-05	1.10E-07	9.39E-03	8.78E-03
Mean	2.67E-03	1.52E-03	8.21E-04	4.34E-03	2.17E-03	6.32E-04	2.52E-06	1.11E-08	7.97E-03	5.43E-03
Median	2.68E-03	1.37E-03	7.48E-04	4.03E-03	2.09E-03	6.31E-04	2.18E-06	6.36E-09	8.02E-03	5.36E-03
StdDev	1.33E-03	6.67E-04	3.98E-04	2.39E-03	1.09E-03	3.13E-04	1.62E-06	1.67E-08	8.46E-04	1.40E-03

CPU Time (s)	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_3}$	$\mathbf{y_4}$	$\mathbf{y_5}$	$\mathbf{y_6}$	$\mathbf{y_7}$	$\mathbf{y_8}$	$\mathbf{y_9}$[a]	$\mathbf{y_{10}}$[b]
Best	1.91486	1.90189	1.83011	1.92487	1.88695	1.91089	0.87168	0.87568	237.2119	235.0178
Worst	2.62698	2.57711	2.58708	2.91122	2.64094	2.56616	1.18384	1.19381	306.0041	306.8119
Mean	2.30448	2.22030	2.22787	2.24806	2.27829	2.24336	0.99652	1.02285	274.2710	274.3978
Median	2.30483	2.20560	2.26894	2.23653	2.28090	2.25049	0.98786	1.02226	276.1613	276.1773
StdDev	0.14412	0.14169	0.15573	0.15622	0.13676	0.13589	0.05753	0.05722	15.3216	13.6593

[a] If log is removed, then: Best = 4.02423 s, Worst = 4.69444 s, Mean = 4.31126 s, Median = 4.34537 s, and StdDev = 0.14379 s.
[b] If log is removed, then: Best = 3.97635 s, Worst = 4.68947 s, Mean = 4.27870 s, Median = 4.32093 s, and StdDev = 0.15387 s.

Chapter 11

Conclusion and Scope of Future Work

11.1 Conclusion

In this research work, different hybrid optimization algorithms are proposed. The basis of all these hybridizations is biogeography-based optimization (BBO). It is hybridized with simulated annealing (SA), differential evolution (DE), sequential quadratic programming (SQP), and linear programming (LP). The Metropolis criterion of SA can enhance the migration stage of BBO, while DE can enhance its exploration level. Based on the objective function, whether it is linear or quadratic, LP or SQP can be applied as a fine-tuning stage. If any one of these two classical gradient-based optimization algorithms is applied, then the elite solutions are fine-tuned and then recycled again in the population by taking the place of the worst individuals. This process can effectively accelerate finding very good feasible solutions without using high population size. These hybrid optimization algorithms are applied to solve many benchmark functions, economic load dispatch (ELD) problems, and optimal relay coordination (ORC) problems. Their performance, against many other algorithms presented in the literature, proves their superiority. They can converge to the global or near-global solutions within a few numbers of iterations and small population size.

Because meta-heuristic optimization algorithms can escape from trapping into local optima, so BBO is hybridized with a feed-forward artificial neural network (ANN) to optimize its hyperparameters before being used to forecast some Nova Scotian heat consumption profiles. The same strategy is applied to a support vector machine (SVM) where a multi-stage random search algorithm (RSA) is employed to optimize its hyperparameters before being used to forecast the peak-load of Nova Scotia during Winter 2018-2019. These two hybrid approaches show that the performance of ANNs and SVMs can be effectively enhanced by optimizing their hyperparameters. This process can be done even with a very basic stochastic algorithm like RSA.

Although modern optimization algorithms and machine learning (ML) tools are great add-ons, the most important stage in modern electric power systems is power flow (PF) analysis. It is agreed that PF is a backbone of almost all the power system studies, such as ELD, unit commitment (UC), optimal power flow (OPF), fault analysis (FA), contingency analysis (CA), state estimation (SE), optimal relay coordination (ORC), power system stability (PSS), automation and control (A&C), etc. Thus, all these stages are affected by incorrect or inaccurate readings that come from the PF stage. However, this crucial analysis also depends on the quality and accuracy of the model used to represent real transmission lines. Thus, even going with highly precise/accurate PF solvers, like Newton-Raphson (NR) method, any weak representation of transmission lines will lead to significant errors in PF, which eventually reflected on all other power system analysis. In this dissertation, a new lumped-parameter model called the M-model is proposed. The nominal circuit of this model provides highly accurate results compared with all the known nominal circuits (short line circuit, Γ-circuit, Ɪ-circuit, T-circuit, and Π-circuit). The M-model is applied to solve the difference in values between ideal and sag transmission line parameters due to the effects of surrounding weather conditions (temperature, humidity, pressure, etc), system frequency, load current, and cable design/status/age. This realization phase reveals some innovative methods to estimate the inductance, capacitance, and conductance temperature coefficients of sag transmission lines, which are not covered in the literature yet.

A part of this realization is applied to solve some problems in PF, FA, ELD, ORC, PSS, and power losses. The results show that the realization phase is very important where some critical deviations in cable temperature and system frequency could lead to non-optimal settings, infeasible solutions, or/and unstable operations. This relaxed realization model is called the temperature/frequency-based model; or just the TFB-model. Compared with the few available alternative models presented in the literature, the TFB-model can preserve the generality of the conventional mathematical models used in many power system studies where the simplicity, feasibility, optimality, and processing speed criteria are kept without any change.

Besides these realization and integration works, many innovative ideas have been presented as an extension to the main work of the dissertation. These new studies

could be applied to many power system applications; including PF, SE, ELD, ORC, energy and load forecasting, smart grids, power system protection, and fault location.

During the journey of this dissertation, it has been discovered that many mathematical formulations, used in electric power systems, are modeled based on linear regression (LR) and nonlinear regression (NLR) analysis. For simple datasets, LR can be used smoothly without facing any accuracy issue. Also, NLR can be used for datasets that have simple nonlinear patterns, so their user-defined functions, initial points, and side constraints can be easily set. For complex datasets, ANNs and SVMs are preferred where the accuracy and precision criteria are very important. However, ANNs and SVMs are black-boxes where the explainability and interpretability criteria are not satisfied. Thus, the last part of this dissertation presents a novel ML computing system called universal functions originator (UFO). This computing system can generate pure mathematical equations universally by varying everything (arithmetic operators, analytical functions, biases, weights of independent variables, weights of dependent variables, exponents of independent variables, and exponents of dependent variables). UFO can act as a general-purpose regression unit where both LR and NLR models can be automatically generated without any external intervention. This is done with the help of two independent stages of optimization algorithms. The first one acts as a global mixed-integer optimization algorithm to build the initial mathematical models, while the other acts as a local gradient-based optimization algorithm to fine-tune their biases, weights, and exponents. By hybridizing UFO with other ML computing systems, to act as a universal transformation unit (UTU), some new frameworks are proposed. For example, it is hybridized with LR to act as a universal linear regression (ULR), with SVM to act as a support functions machine (SFM), and with ANN to act as an artificial mathematical network (AMN). These four new computing systems have lots of useful applications in many electric power systems. In this dissertation, they are applied to solve some practical problems; particularly, in predicting full-load power output of a combined-cycle power plant (CCPP) and in locating short-circuit faults in a mesh network. They show very impressive results compared with classical ML computing systems, which encourage us to continue improving their mechanisms to be more mature.

11.2 Scope of Future Work

Each chapter contains multiple topics and each topic raises multiple questions. Thus, the door is open wide to do further researches. The following points cover the most important works that could be conducted in the future:

- Nowadays, the literature has many new meta-heuristic optimization algorithms that are waiting for someone to discover their potentials and strengths. Thus, there is an opportunity to do many modifications and hybridizations to get highly superior optimization algorithms that can satisfy or compromise multiple performance criteria (accuracy, speed, simplicity, etc). Also, during doing this Ph.D. dissertation, four completely new meta-heuristic optimization algorithms have been designed. They are not included here because their performance against other optimization algorithms has not been evaluated yet.

- The performance of the hybrid optimization algorithms designed by BBO, SA, and SQP (i.e., the MpBBO-SQP algorithm) can be enhanced by conducting some sensitivity analysis to find its optimal settings. Moreover, it is good to test MpBBO and MpBBO-SQP with adaptive cooling strategies suggested in the literature for SA. Furthermore, the mutation and migration rates of BBO can be replaced with other more advanced rates presented in the literature. Also, the oppositional and blended versions of BBO (i.e., OBBO and BBBO) can be considered as new phases for getting more advanced hybrid optimization algorithms.

- In the TFB-model, only the effects of cable temperature and system frequency are considered. Thus, the other effects could also be considered to minimize the overall error. Furthermore, the formulas developed for the inductance, capacitance, and conductance temperature coefficients could be implemented to see their performance. Their accuracy could be enhanced by including higher-order regression terms. Moreover, the novel techniques proposed to calculate the actual distributed parameters of sag transmission lines could be used in different power system studies.

- There is no test system developed for our proposed realistic multi-fuel-based

ELD model that contains multiple generating units connected to some common busbars. Thus, there is a need to provide that so other researchers could contribute to this very challenging ELD problem. The same thing can be applied for the spinning reserve thermal generating units that are kept running without connected to the gird. Also, the new ML computing systems presented in this dissertation could be used to search for highly accurate nonlinear multi-fuel cost functions instead of sticking to LR models.

- For the transformation-based linear fuel-cost function, the other possible transformations could be investigated as well. Both UFO and ULR could be involved here to extract many transformed linear models. Also, there is a need to check the performance of this new approach in some test systems and then comparing its results with those of classical quadratic and cubic models, so the overall performance can be evaluated.

- The optimization-free economic load dispatcher designed based on the sliced fuel-cost curve (SFCC) strategy could be extended with more information and numerical analysis to show how this new technique works with real-world problems. Also, it could be hybridized with combinatorial optimization algorithms. Thus, the convergence rate can be accelerated by giving more weights to the suspected spots of the map extracted from all the feasible solutions of SFCC.

- The solution accuracy of the optimization/modeling-free economic load dispatcher (OMF-ELD) can be improved by involving many other data of power stations (like temperature, pressure, relative humidity, sensors condition, current efficiency, emission rates, etc). Also, that solution could be improved if the dataset is grouped per similar months or seasons. Add to that, by effectively utilizing the capability of modern automation and energy management systems, the mechanism of OMF-ELD could be extended to cover the UC part; or, in other words, to act as OMF-UC.

- For the strategy proposed to trade non-electric energy sources locally between entities, it is recommended to heavily integrate many topics and disciplines such as heat transfer, fluid transport, thermodynamics, chemical engineering,

and C&I engineering. The study also needs to be extended with more details about how to establish this unfamiliar local energy trading market for the next-generation smart and micro-grids. Also, it needs to show the ways to practically design smart pipeline networks and operate public logs warehouses (PLW), and how to find the best configurations for both the seller-side and the buyer-side.

- For the multi-technology ORC model, it is known that each relay technology has many manufacturers, and each manufacturer has different models. Thus, to make the results more realistic, a slight difference in the settings of each relay technology can be considered here. Thus, that realistic ORC model can be extended to deal with this real situation.

- The double primary relay strategy (DPRS) can be extended by applying it to a few circuit breakers instead of covering all of them. Also, a combination of numerical, static, electromechanical relays as well as fuses and distance relays could be considered together in one network, which is a more realistic problem. Multiple North American and European TCCC standards could also be considered for more advanced and *high*-dimensional ORC problems. Add to that, user-defined TCCCs could be considered as well. However, going with multiple TCCCs requires special care to avoid facing infeasible solutions.

- The technique proposed to locate faults, from the operating times measured from the two-end directional overcurrent relays (DOCRs) of each faulty line, can be upgraded to consider multi-terminal lines. This special topology requires some essential modifications to the main structure of the current technique. Also, the operating times of back-up DOCRs could be utilized to increase the prediction accuracy.

- For the concept of applying linear heat sensors (LHS) as very cheap protective devices, the two important things that need to be investigated are the type and thickness of their polymer insulations. These two factors can control the interrupting current and the detection speed.

- To increase the accuracy of ANNs, used for estimating power flow quantities

(including directions, magnitudes, and losses), the highly advanced ANN topologies could be considered here.

- The same thing can be applied to increase the forecasting accuracy of ANNs. Also, the past energy readings can be recycled again to the input stream by using feedback neural networks. Add to that, the other missing variables, such as holidays and social events, could be considered to increase the explanation level of data variability.

Due to the deep loss of my late supervisor, Prof. Mohamed E. El-Hawary, tens of new researches that were under study and preparation are postponed and kept as handwritten notes. Many of them are multidisciplinary studies that contain highly innovative ideas and have never been addressed in the literature. Also, many of them can be converted into hardware and software products. These researches mainly focus on energy harvesting, power system protection, power system operation, demand forecasting, state estimation, control systems, and artificial intelligence.

Appendices

A. Linearizing the IEC/BS Model of DOCRs by Fixing Time Multiplier Settings

The operating time of the ith DOCR can be calculated at the kth location as follows:

$$T_{i,k} = TMS_i \times \frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} \tag{A.1}$$

By taking the reciprocal of both sides:

$$\frac{1}{T_{i,k}} = \frac{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1}{\beta\ TMS_i} \tag{A.2}$$

Distributing the exponent α and multiplying both sides by $\beta\ TMS_i$:

$$\frac{\beta\ TMS_i}{T_{i,k}} = \frac{I_{i,k}^{\alpha}}{PS_i^{\alpha}} - 1 = \frac{I_{i,k}^{\alpha} - PS_i^{\alpha}}{PS_i^{\alpha}} \tag{A.3}$$

By re-taking the reciprocal of both sides:

$$\frac{T_{i,k}}{\beta\ TMS_i} = \frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}} \tag{A.4}$$

Multiplying both sides by $\beta\ TMS_i$:

$$\therefore\ T_{i,k} = \beta\ TMS_i \left(\frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}}\right) = \boxed{\vartheta_i \widetilde{PS}_{i,k}} \tag{A.5}$$

where ϑ_i and $\widetilde{PS}_{i,k}$ are respectively equal to:

$$\boxed{\vartheta_i = \beta\ TMS_i} \tag{A.6}$$

$$\boxed{\widetilde{PS}_{i,k} = \frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}}} \tag{A.7}$$

To extract PS_i from (A.7), first, both sides should be multiplied by $\left(I_{i,k}^{\alpha} - PS_i^{\alpha}\right)$:

$$\widetilde{PS}_{i,k}\left(I_{i,k}^{\alpha} - PS_i^{\alpha}\right) = PS_i^{\alpha} \tag{A.8}$$

By applying the distributive property, the parentheses can be removed:

$$\widetilde{PS}_{i,k} I_{i,k}^{\alpha} - \widetilde{PS}_{i,k} PS_i^{\alpha} = PS_i^{\alpha} \tag{A.9}$$

Collecting the terms containing PS_i^{α} in one side:

$$\widetilde{PS}_{i,k} I_{i,k}^{\alpha} = PS_i^{\alpha} + \widetilde{PS}_{i,k} PS_i^{\alpha} = PS_i^{\alpha} \left(1 + \widetilde{PS}_{i,k}\right) \tag{A.10}$$

Therefore,

$$\begin{aligned} PS_i &= \left[\frac{\widetilde{PS}_{i,k} I_{i,k}^{\alpha}}{1 + \widetilde{PS}_{i,k}}\right]^{1/\alpha} = \frac{\left(\widetilde{PS}_{i,k} I_{i,k}^{\alpha}\right)^{1/\alpha}}{\left(1 + \widetilde{PS}_{i,k}\right)^{1/\alpha}} \\ &= \frac{\left(\widetilde{PS}_{i,k}\right)^{1/\alpha} \left(I_{i,k}^{\alpha}\right)^{1/\alpha}}{\left(1 + \widetilde{PS}_{i,k}\right)^{1/\alpha}} = \frac{(I_{i,k})^{\alpha/\alpha} \sqrt[\alpha]{\widetilde{PS}_{i,k}}}{\sqrt[\alpha]{1 + \widetilde{PS}_{i,k}}} \end{aligned}$$

$$\therefore \quad \boxed{PS_i = I_{i,k} \times \sqrt[\alpha]{\frac{\widetilde{PS}_{i,k}}{1 + \widetilde{PS}_{i,k}}}} \tag{A.11}$$

B. Linearizing the ANSI/IEEE Model of DOCRs by Fixing Time Multiplier Settings

The operating time of the ith DOCR can be calculated at the kth location as follows:

$$T_{i,k} = TMS_i \times \left[\frac{\beta}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} + \gamma \right] \tag{B.1}$$

Moving the gamma term to the left-side:

$$T_{i,k} - \gamma\ TMS_i = \frac{\beta\ TMS_i}{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1} \tag{B.2}$$

By taking the reciprocal of both sides:

$$\frac{1}{T_{i,k} - \gamma\ TMS_i} = \frac{\left(\frac{I_{i,k}}{PS_i}\right)^{\alpha} - 1}{\beta\ TMS_i} \tag{B.3}$$

Distributing the exponent α and multiplying both sides by $\beta\ TMS_i$:

$$\frac{\beta\ TMS_i}{T_{i,k} - \gamma\ TMS_i} = \frac{I_{i,k}^{\alpha}}{PS_i^{\alpha}} - 1 = \frac{I_{i,k}^{\alpha} - PS_i^{\alpha}}{PS_i^{\alpha}} \tag{B.4}$$

By re-taking the reciprocal of both sides:

$$\frac{T_{i,k} - \gamma\ TMS_i}{\beta\ TMS_i} = \frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}} \tag{B.5}$$

Multiplying both sides by $\beta\ TMS_i$:

$$T_{i,k} - \gamma\ TMS_i = \beta\ TMS_i \left(\frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}} \right) \tag{B.6}$$

Moving the gamma term to the right-side:

$$\therefore\ T_{i,k} = \beta\ TMS_i \left(\frac{PS_i^{\alpha}}{I_{i,k}^{\alpha} - PS_i^{\alpha}} \right) + \gamma\ TMS_i = \boxed{\vartheta_i \widetilde{PS}_{i,k} + \xi_i} \tag{B.7}$$

where ϑ_i and $\widetilde{PS}_{i,k}$ are respectively given in (A.6)-(A.7), and the last term ξ_i is equal to:

$$\boxed{\xi_i = \gamma\ TMS_i} \tag{B.8}$$

C. Finding the Number of Generations Assigned to Each Stage of RSA

Let's consider the following infinite sequence:

$$h_\infty = \left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \cdots \right\} \tag{C.1}$$

Then, the infinite geometric series can be mathematically expressed as follows:

$$s_\infty = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots \tag{C.2}$$

which can also be expressed as:

$$s_\infty = \sum_{j=1}^{\infty} \left(\frac{1}{2}\right)^j \tag{C.3}$$

where s_∞ approaches one as j approaches infinity (i.e., $s_\infty \to 1$ as $j \to \infty$). Thus, if the total number of iterations G_{RSA} assigned to the random search algorithm (RSA) is distributed among infinite stages, then (C.3) becomes:

$$\begin{aligned} G_{\text{RSA}} &= G_{\text{RSA}} \sum_{j=1}^{\infty} \left(\frac{1}{2}\right)^j \\ &= \frac{G_{\text{RSA}}}{2} + \frac{G_{\text{RSA}}}{4} + \frac{G_{\text{RSA}}}{8} + \frac{G_{\text{RSA}}}{16} + \cdots \end{aligned} \tag{C.4}$$

For ς stages, G_{RSA} can be approximated as follows:

$$G_{\text{RSA}} \approx G_{\text{RSA}} \sum_{j=1}^{\varsigma} \left(\frac{1}{2}\right)^j \tag{C.5}$$

The symbol $\approx$ is used because $\varsigma \neq \infty$. Thus, to equalize both sides, the left hand side is multiplied by s_ς as follows:

$$s_\varsigma G_{\text{RSA}} = G_{\text{RSA}} \sum_{j=1}^{\varsigma} \left(\frac{1}{2}\right)^j \tag{C.6}$$

where s_ς is the ς terms sum of (C.3), which can be calculated as:

$$s_\varsigma = \sum_{j=1}^{\varsigma} \left(\frac{1}{2}\right)^j \tag{C.7}$$

Expanding the last equation and taking $r = \frac{1}{2}$ yields:

$$s_\varsigma = r + r^2 + r^3 + \cdots + r^{\varsigma-1} + r^\varsigma \tag{C.8}$$

Multiplying both sides by r yields:

$$rs_{\varsigma} = r^2 + r^3 + r^4 + \cdots + r^{\varsigma} + r^{\varsigma+1} \tag{C.9}$$

Subtracting (C.9) from (C.8) yields:

$$(1-r)\, s_{\varsigma} = r - r^{\varsigma+1} \tag{C.10}$$

Therefore, s_{ς} can be directly determined for any non-zero positive integer value of ς as follows:

$$s_{\varsigma} = \frac{r - r^{\varsigma+1}}{1-r} = \frac{r\,(1-r^{\varsigma})}{1-r} \tag{C.11}$$

Returning r back to $\frac{1}{2}$:

$$\therefore\ s_{\varsigma} = \frac{\frac{1}{2}\left[1-\left(\frac{1}{2}\right)^{\varsigma}\right]}{1-\frac{1}{2}} = \frac{\cancel{\frac{1}{2}}\left(1-\frac{1}{2^{\varsigma}}\right)}{\cancel{\frac{1}{2}}} = \boxed{1-2^{-\varsigma}} \tag{C.12}$$

Substituting (C.12) in (C.6) for s_{ς} yields:

$$G_{\mathrm{RSA}} = \frac{G_{\mathrm{RSA}}}{s_{\varsigma}} \sum_{j=1}^{\varsigma} \left(\frac{1}{2}\right)^{j} = \frac{G_{\mathrm{RSA}}}{1-2^{-\varsigma}} \sum_{j=1}^{\varsigma} \left(\frac{1}{2}\right)^{j} \tag{C.13}$$

Now, let's assume that the total number of generations are distributed among ς stages as:

$$G_{\mathrm{RSA}} = \chi_1 + \chi_2 + \cdots + \chi_j + \cdots + \chi_{\varsigma} \tag{C.14}$$

To find the specific number of iterations assigned to χ_j, then by referring to (C.4):

$$\therefore\ \chi_j = 2^{-j} s_{\varsigma}^{-1} G_{\mathrm{RSA}} \tag{C.15}$$

To avoid getting a float number, the final value is rounded as follows:

$$\therefore\ \boxed{\chi_j = \left\lfloor 2^{-j} s_{\varsigma}^{-1} G_{\mathrm{RSA}} \right\rceil} \tag{C.16}$$

As a numerical example, assume $G_{\mathrm{RSA}} = 1500$ and $\varsigma = 4$, then:

$$s_4 = 1 - 2^{-4} = \frac{15}{16}$$

$$G_{\mathrm{RSA}} = \chi_1 + \chi_2 + \chi_3 + \chi_4 = \left[1500\left(\frac{16}{15}\right)\right]\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16}\right) = 1500$$

D. Deriving the Short-Length Transmission Line Model

This model can be solved by multiplying the per the unit length series impedance z by the total length $\mathfrak{L}$ [326]. From the simplified circuit shown in Figure D.1, if Kirchhoff's current law (KCL) is applied[1]:

$$i(x,t) = i(x+\Delta x,t) \tag{D.1}$$

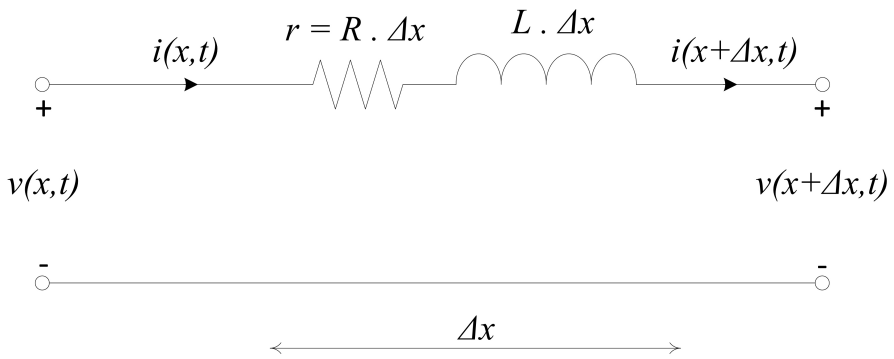


Figure D.1. Transmission Line Length Section Δx (Short-Line Model)

Applying Kirchhoff's voltage law (KVL):

$$v(x,t) = R\ \Delta x\ i(x,t) + L\Delta x\frac{\partial i(x,t)}{\partial t} + v(x+\Delta x,t) \tag{D.2}$$

$$\frac{v(x+\Delta x,t) - v(x,t)}{\Delta x} = -\left[R\ i(x,t) + L\frac{\partial i(x,t)}{\partial t}\right] \tag{D.3}$$

As $\Delta x \ \rightarrow\ 0$:

$$-\frac{\partial v}{\partial x} = R\ i + L\frac{\partial i}{\partial t} \tag{D.4}$$

The above equation can be easily solved by converting it to a complex form in order to deal with one independent variable x:

$$Z = (r + j\omega L)\ \mathfrak{L} = R + jX \tag{D.5}$$

where r is the per-phase resistance per unit length Δx, and $\mathfrak{L}$ is the total length of the transmission line.

It is known that the voltage is transmitted from the sender V_S to the receiver V_R through the powerlines from $\mathfrak{L}_{min}$ to $\mathfrak{L}_{max}$, and thus a portion of the electricity is lost

[1] By considering no effect of t "transient condition on the per-unit-length inductance".

across the line. The circuit representation is shown in Figure 6.1. If a small z is used to denote the per unit length impedance, then:

$$-\int_{V_S}^{V_R} dV = z\, I \int_{\mathfrak{L}_{min}}^{\mathfrak{L}_{max}} dx \tag{D.6}$$

$$-\Big[V\Big]_{V_S}^{V_R} = z\, I \Big[x\Big]_{\mathfrak{L}_{min}}^{\mathfrak{L}_{max}} \tag{D.7}$$

$$V_s - V_R = z\, I\, (\mathfrak{L}_{max} - \mathfrak{L}_{min}) \tag{D.8}$$

$$\because\ \mathfrak{L} = \mathfrak{L}_{max} - \mathfrak{L}_{min} \ \rightarrow\ \text{the total length} \tag{D.9}$$

$$\therefore\ V_s - V_R = z\, I\, \mathfrak{L} \tag{D.10}$$

$$\because\ Z = z\, \mathfrak{L} \ \rightarrow\ \text{the total impedance} \tag{D.11}$$

$$\therefore\ \boxed{V_S = V_R + ZI} \tag{D.12}$$

Based on (D.1):

$$\therefore\ \boxed{I = I_S = I_R} \tag{D.13}$$

Equations (D.12) and (D.13) can be reformulated again in order to have a 2×2 matrix, as a two-port network, as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{D.14}$$

The determinant of this matrix can be calculated as follows:

$$AD - \cancel{BC} = 1 \tag{D.15}$$

Thus, the previous two-port network can be re-expressed to act as a measurement toolbox for the receiving-end of the transmission line[a]:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & -Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{D.16}$$

[a] Logically and Practically, V_S and I_S are supposed to be known by the electricity producer, while V_R and I_R are supposed to be unknown based on the condition of the transmission line.

E. Deriving the Medium-Length Transmission Line Model (Nominal Γ-Representation)

To derive the model of the nominal circuit shown in Figure 6.3, first KVL is applied in the right closed-loop:

$$\boxed{V_S = V_R + ZI_R} \tag{E.1}$$

Applying KCL at the top-left node:

$$I_S = I_C + I_R \tag{E.2}$$

Applying Ohm's law to that node:

$$I_C = YV_S \tag{E.3}$$

Substituting (E.3) in (E.2) for I_C yields:

$$I_S = YV_S + I_R \tag{E.4}$$

Substituting (E.1) in (E.4) for V_S yields:

$$I_S = Y\left(V_R + ZI_R\right) + I_R$$

$$\therefore \quad \boxed{I_S = YV_R + \left(1 + ZY\right)I_R} \tag{E.5}$$

The two-port network of this model can be built by expressing (E.1) and (E.5) in a matrix form as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ Y & 1+ZY \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{E.6}$$

The determinant of this matrix can be calculated as follows:

$$\begin{aligned} AD - BC &= (1)\left(1 + ZY\right) - (Z)\left(Y\right) \\ &= 1 + \cancel{ZY} - \cancel{ZY} = 1 \end{aligned} \tag{E.7}$$

Thus, if $\{V_S, I_S\}$ are known and $\{V_R, I_R\}$ are unknown, then the two-port network becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+ZY & -Z \\ -Y & 1 \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{E.8}$$

F. Deriving the Medium-Length Transmission Line Model (Nominal T-Representation)

To derive the model of the nominal circuit shown in Figure 6.4, first KCL is applied at the top-right node:

$$I_S = I_C + I_R \tag{F.1}$$

Applying Ohm's law to that node:

$$I_C = YV_R \tag{F.2}$$

Substituting (F.2) in (F.1) for I_C yields:

$$\therefore \quad \boxed{I_S = YV_R + I_R} \tag{F.3}$$

Applying KVL in the left closed-loop:

$$V_S = ZI_S + V_R \tag{F.4}$$

Substituting (F.3) in (F.4) for I_S yields:

$$V_S = Z\left(YV_R + I_R\right) + V_R$$

$$\therefore \quad \boxed{V_S = \left(1 + ZY\right)V_R + ZI_R} \tag{F.5}$$

The two-port network of this model can be built by expressing (F.3) and (F.5) in a matrix form as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+ZY & Z \\ Y & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{F.6}$$

Again, the determinant is unity, which can be calculated as follows:

$$\begin{aligned} AD - BC &= \left(1+ZY\right)\left(1\right) - \left(Z\right)\left(Y\right) \\ &= 1 + \cancel{ZY} - \cancel{ZY} = 1 \end{aligned} \tag{F.7}$$

Thus, if $\{V_S, I_S\}$ are known and $\{V_R, I_R\}$ are unknown, then the two-port network becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & -Z \\ -Y & 1+ZY \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{F.8}$$

G. Deriving the Medium-Length Transmission Line Model (Nominal T-Representation)

To derive the model of the nominal circuit shown in Figure 6.5, first KCL is applied at the midpoint:

$$I_S = I_C + I_R \tag{G.1}$$

Applying KVL in the left closed-loop:

$$V_S = \frac{Z}{2} I_S + V_C \tag{G.2}$$

Applying KVL in the right closed-loop:

$$V_C = \frac{Z}{2} I_R + V_R \tag{G.3}$$

Substituting (G.3) in (G.2) for V_C yields:

$$V_S = \frac{Z}{2} I_S + \frac{Z}{2} I_R + V_R \tag{G.4}$$

Applying Ohm's law to the midpoint:

$$I_C = Y V_C \tag{G.5}$$

Substituting (G.3) in (G.5) for V_C yields:

$$I_C = \frac{ZY}{2} I_R + Y V_R \tag{G.6}$$

Substituting (G.6) in (G.1) for I_C yields:

$$I_S = I_R + \frac{ZY}{2} I_R + Y V_R$$

$$\therefore \boxed{I_S = Y V_R + \left(1 + \frac{YZ}{2}\right) I_R} \tag{G.7}$$

Substituting (G.7) in (G.4) for I_S yields:

$$\begin{aligned} V_S &= \frac{Z}{2}\left[Y V_R + \left(1 + \frac{ZY}{2}\right) I_R\right] + \frac{Z}{2} I_R + V_R \\ &= \frac{ZY}{2} V_R + \frac{Z}{2} I_R + \frac{Z^2 Y}{4} I_R + \frac{Z}{2} I_R + V_R \end{aligned}$$

$$\therefore \boxed{V_S = \left(1 + \frac{ZY}{2}\right) V_R + Z\left(1 + \frac{ZY}{4}\right) I_R} \tag{G.8}$$

The two-port network of this model can be built by formulating (G.7) and (G.8) in a matrix form as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & Z\left(1+\frac{ZY}{4}\right) \\ Y & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{G.9}$$

The determinant can be determined as follows:

$$\begin{aligned} AD - BC &= \left(1+\frac{ZY}{2}\right)\left(1+\frac{ZY}{2}\right) - ZY\left(1+\frac{ZY}{4}\right) \\ &= 1 + \cancel{ZY} + \cancel{\frac{Z^2Y^2}{4}} - \cancel{ZY} - \cancel{\frac{Z^2Y^2}{4}} = 1 \end{aligned} \tag{G.10}$$

Thus, if V_R and I_R are the two unknowns required to be calculated, then the two-port network becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & -Z\left(1+\frac{ZY}{4}\right) \\ -Y & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{G.11}$$

H. Deriving the Medium-Length Transmission Line Model (Nominal Π-Representation)

To derive the model of the nominal circuit shown in Figure 6.6, first KCL is applied at the top-right node:

$$I_L = I_{C_2} + I_R \tag{H.1}$$

Applying Ohm's law to that point again:

$$I_{C_2} = \frac{Y}{2} V_R \tag{H.2}$$

Substituting (H.2) in (H.1) for I_{C_2} yields:

$$I_L = \frac{Y}{2} V_R + I_R \tag{H.3}$$

Applying KVL in the middle closed-loop:

$$V_S = V_R + Z I_L \tag{H.4}$$

Substituting (H.3) in (H.4) for I_L yields:

$$V_S = V_R + Z \left(I_R + \frac{Y}{2} V_R \right)$$

$$\therefore \quad \boxed{V_S = \left(1 + \frac{ZY}{2} \right) V_R + Z I_R} \tag{H.5}$$

Applying KCL at the top-left node:

$$I_S = I_L + I_{C_1} \tag{H.6}$$

Applying Ohm's law to the preceding node:

$$I_{C_1} = \frac{Y}{2} V_S \tag{H.7}$$

Substituting (H.3) and (H.7) in (H.6) for I_L and I_{C_1}, respectively, yields:

$$I_S = \frac{Y}{2} V_S + \frac{Y}{2} V_R + I_R \tag{H.8}$$

Substituting (H.5) in (H.8) for V_S yields:

$$\begin{aligned} I_S &= I_R + \frac{Y}{2} V_R + \frac{Y}{2} \left[\left(1 + \frac{ZY}{2} \right) V_R + Z I_R \right] \\ &= I_R + \frac{Y}{2} V_R + \frac{Y}{2} \left(1 + \frac{ZY}{2} \right) V_R + \frac{ZY}{2} I_R \\ &= \frac{Y}{2} \left(2 + \frac{ZY}{2} \right) V_R + \left(1 + \frac{ZY}{2} \right) I_R \end{aligned}$$

$$\therefore \quad \boxed{I_S = Y\left(1+\frac{ZY}{4}\right)V_R + \left(1+\frac{ZY}{2}\right)I_R} \tag{H.9}$$

The two-port network of this model can be constructed from (H.5) and (H.9) as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & Z \\ Y\left(1+\frac{ZY}{4}\right) & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{H.10}$$

The determinant of this matrix is unity, which can be determined as follows:

$$\begin{aligned} AD - BC &= \left(1+\frac{ZY}{2}\right)\left(1+\frac{ZY}{2}\right) - ZY\left(1+\frac{ZY}{4}\right) \\ &= 1 + \cancel{ZY} + \cancel{\frac{Z^2Y^2}{4}} - \cancel{ZY} - \cancel{\frac{Z^2Y^2}{4}} = 1 \end{aligned} \tag{H.11}$$

Thus, if V_R and I_R are the two unknowns required to be calculated, then the two-port network becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{2} & -Z \\ -Y\left(1+\frac{ZY}{4}\right) & 1+\frac{ZY}{2} \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{H.12}$$

I. Deriving the Medium-Length Transmission Line Model (Nominal M-Representation)

To derive the model of the nominal circuit shown in Figure 6.8, first KCL is applied at node B:

$$I_{CB} = I_{B0} + I_R \tag{I.1}$$

Applying Ohm's law to that point again:

$$I_{B0} = \frac{Y}{3} V_R \tag{I.2}$$

Substituting (I.2) in (I.1) for I_{B0} yields:

$$I_{CB} = \frac{Y}{3} V_R + I_R \tag{I.3}$$

Applying KCL at node C:

$$I_{AC} = I_{C0} + I_{CB} \tag{I.4}$$

Applying Ohm's law to that point again:

$$I_{C0} = \frac{Y}{3} V_C \tag{I.5}$$

Applying KVL in loop CB:

$$V_C = \frac{Z}{2} I_{CB} + V_R \tag{I.6}$$

Substituting (I.3) in (I.6) for I_{CB} yields:

$$V_C = \frac{Z}{2}\left(\frac{Y}{3} V_R + I_R\right) + V_R$$

$$\therefore\ V_C = \left(1 + \frac{ZY}{6}\right) V_R + \frac{Z}{2} I_R \tag{I.7}$$

Substituting (I.7) in (I.5) for V_C yields:

$$I_{C0} = \frac{Y}{3}\left(1 + \frac{ZY}{6}\right) V_R + \frac{ZY}{6} I_R \tag{I.8}$$

Substituting (I.8) and (I.3) in (I.4) for I_{C0} and I_{CB}, respectively, yields:

$$I_{AC} = \frac{Y}{3}\left(1 + \frac{ZY}{6}\right) V_R + \frac{ZY}{6} I_R + \frac{Y}{3} V_R + I_R$$

$$\therefore\ I_{AC} = \frac{Y}{3}\left(2 + \frac{ZY}{6}\right) V_R + \left(1 + \frac{ZY}{6}\right) I_R \tag{I.9}$$

Applying KCL at node A:

$$I_S = I_{A0} + I_{AC} \tag{I.10}$$

Applying Ohm's law to node A:

$$I_{A0} = \frac{Y}{3} V_S \tag{I.11}$$

Substituting (I.11) and (I.9) in (I.10) for I_{A0} and I_{AC}, respectively, yields:

$$I_S = \frac{Y}{3} V_S + \frac{Y}{3}\left(2 + \frac{ZY}{6}\right) V_R + \left(1 + \frac{ZY}{6}\right) I_R \tag{I.12}$$

Applying KVL in loop AC:

$$V_S = \frac{Z}{2} I_{AC} + V_C \tag{I.13}$$

Substituting (I.9) and (I.7) in (I.13) for I_{AC} and V_C, respectively, yields:

$$\begin{aligned} V_S &= \frac{ZY}{6}\left(2 + \frac{ZY}{6}\right) V_R + \frac{Z}{2}\left(1 + \frac{ZY}{6}\right) I_R + \left(1 + \frac{ZY}{6}\right) V_R + \frac{Z}{2} I_R \\ &= \frac{ZY}{3} V_R + \frac{Z^2Y^2}{36} V_R + \frac{Z}{2} I_R + \frac{Z^2Y}{12} I_R + V_R + \frac{ZY}{6} V_R + \frac{Z}{2} I_R \end{aligned}$$

$$\therefore \boxed{V_S = \left[1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right)\right] V_R + Z\left(1 + \frac{ZY}{12}\right) I_R} \tag{I.14}$$

Substituting (I.14) in (I.12) for V_S yields:

$$\begin{aligned} I_S &= \frac{Y}{3}\left\{\left[1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right)\right] V_R + Z\left(1 + \frac{ZY}{12}\right) I_R\right\} + \frac{Y}{3}\left(2 + \frac{ZY}{6}\right) V_R \\ &\quad + \left(1 + \frac{ZY}{6}\right) I_R \\ &= \left[\frac{Y}{3} + \frac{ZY^2}{18}\left(3 + \frac{ZY}{6}\right)\right] V_R + \frac{ZY}{3}\left(1 + \frac{ZY}{12}\right) I_R + \frac{2}{3} Y V_R + \frac{ZY^2}{18} V_R \\ &\quad + I_R + \frac{ZY}{6} I_R \\ &= \frac{Y}{3} V_R + \frac{ZY^2}{6} V_R + \frac{Z^2Y^3}{108} V_R + \frac{2}{3} Y V_R + \frac{ZY^2}{18} V_R + \frac{ZY}{3} I_R + \frac{Z^2Y^2}{36} I_R \\ &\quad + I_R + \frac{ZY}{6} I_R \\ &= \left(Y + \frac{2ZY^2}{9} + \frac{Z^2Y^3}{108}\right) V_R + \left(1 + \frac{ZY}{2} + \frac{Z^2Y^2}{36}\right) I_R \end{aligned}$$

$$\therefore \boxed{I_S = \left[Y + \frac{ZY^2}{9}\left(2 + \frac{ZY}{12}\right)\right] V_R + \left[1 + \frac{ZY}{6}\left(3 + \frac{ZY}{6}\right)\right] I_R} \tag{I.15}$$

From (I.14) and (I.15), the two-port network of this model can be constructed as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1+\frac{ZY}{6}\left(3+\frac{ZY}{6}\right) & Z\left(1+\frac{ZY}{12}\right) \\ Y+\frac{ZY^2}{9}\left(2+\frac{ZY}{12}\right) & 1+\frac{ZY}{6}\left(3+\frac{ZY}{6}\right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{I.16}$$

The determinant of this matrix is unity, which can be determined by simplifying $AD - BC$.

Thus, if V_R and I_R are the two unknowns required to be calculated, then the two-port network becomes:

$$\begin{aligned} \begin{bmatrix} V_R \\ I_R \end{bmatrix} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \\ &= \begin{bmatrix} 1+\frac{ZY}{6}\left(3+\frac{ZY}{6}\right) & -Z\left(1+\frac{ZY}{12}\right) \\ -Y-\frac{ZY^2}{9}\left(2+\frac{ZY}{12}\right) & 1+\frac{ZY}{6}\left(3+\frac{ZY}{6}\right) \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \end{aligned} \tag{I.17}$$

J. Deriving the Long-Length Transmission Line Model (Nominal Π-Section)

To derive the distributed parameter model with a Π-section similar to that shown in Figure 6.7, first KCL is applied at node P [29]:

$$i(x+\Delta x,t) = i_{G_1} + i_{C_1} + i_L \tag{J.1}$$

Applying Ohm's law to node P:

$$i_{G_1} = \frac{G}{2}\Delta x\ v(x+\Delta x,t) \tag{J.2}$$

$$i_{C_1} = \frac{C}{2}\Delta x\frac{\partial v(x+\Delta x,t)}{\partial t} \tag{J.3}$$

Applying KCL at node Q:

$$i_L = i_{G_2} + i_{C_2} + i(x,t) \tag{J.4}$$

Applying Ohm's law to node Q:

$$i_{G_2} = \frac{G}{2}\Delta x\ v(x,t) \tag{J.5}$$

$$i_{C_2} = \frac{C}{2}\Delta x\frac{\partial v(x,t)}{\partial t} \tag{J.6}$$

Substituting (J.5) and (J.6) in (J.4) for i_{G_2} and i_{C_2}, respectively, yields:

$$i_L = \frac{G}{2}\Delta x\ v(x,t) + \frac{C}{2}\Delta x\frac{\partial v(x,t)}{\partial t} + i(x,t) \tag{J.7}$$

Now, substituting (J.2), (J.3) and (J.7) in (J.1) for i_{G_1}, i_{C_1} and i_L, respectively, yields:

$$\begin{aligned} i(x+\Delta x,t) = {} & \frac{G}{2}\Delta x\ v(x+\Delta x,t) + \frac{C}{2}\Delta x\frac{\partial v(x+\Delta x,t)}{\partial t} + \frac{G}{2}\Delta x\ v(x,t) \\ & + \frac{C}{2}\Delta x\frac{\partial v(x,t)}{\partial t} + i(x,t) \end{aligned} \tag{J.8}$$

Re-arranging (J.8) gives:

$$\begin{aligned} \frac{i(x+\Delta x,t) - i(x,t)}{\Delta x} = {} & \frac{G}{2}\left[v(x+\Delta x,t) + v(x,t)\right] \\ & + \frac{C}{2}\left[\frac{\partial v(x+\Delta x,t)}{\partial t} + \frac{\partial v(x,t)}{\partial t}\right] \end{aligned} \tag{J.9}$$

As $\Delta x \rightarrow 0$, the left-hand side becomes a partial derivative of i with respect to x:

$$\frac{\partial i(x,t)}{\partial x} = Gv(x,t) + C\frac{\partial v(x,t)}{\partial t} \quad \text{(the 1st Telegrapher's equation)} \tag{J.10}$$

Now, to find the 2nd Telegrapher's equation, let's focus on the behavior of the voltage across different locations of the circuit. Applying KVL in the branch between node P and node Q (i.e., the middle closed-loop):

$$v(x+\Delta x,t) = R\Delta x\ i_L + L\Delta x\frac{\partial i_L}{\partial t} + v(x,t) \tag{J.11}$$

Re-arranging (J.11) gives:

$$\frac{v(x+\Delta x,t) - v(x,t)}{\Delta x} = R\ i_L + L\frac{\partial i_L}{\partial t} \tag{J.12}$$

As $\Delta x \rightarrow 0$, the left-hand side becomes a partial derivative of v with respect to x:

$$\frac{\partial v(x,t)}{\partial x} = R\ i_L + L\frac{\partial i_L}{\partial t} \tag{J.13}$$

To have $\frac{\partial i_L}{\partial t}$, let's differentiate (J.7) with respect to t as follows:

$$\frac{\partial i_L}{\partial t} = \frac{G}{2}\Delta x\frac{\partial v(x,t)}{\partial t} + \frac{C}{2}\Delta x\frac{\partial^2 v(x,t)}{\partial t^2} + \frac{\partial i(x,t)}{\partial t} \tag{J.14}$$

Substituting (J.7) and (J.14) in (J.13) for i_L and $\frac{\partial i_L}{\partial t}$, respectively, yields:

$$\begin{aligned}\frac{\partial v(x,t)}{\partial x} &= R\left[\frac{G}{2}\cancel{\Delta x}\ v(x,t) + \frac{C}{2}\cancel{\Delta x}\frac{\partial v(x,t)}{\partial t} + i(x,t)\right] \\ &\quad + L\left[\frac{G}{2}\cancel{\Delta x}\frac{\partial v(x,t)}{\partial t} + \frac{C}{2}\cancel{\Delta x}\frac{\partial^2 v(x,t)}{\partial t^2} + \frac{\partial i(x,t)}{\partial t}\right]\end{aligned} \tag{J.15}$$

Because $\Delta x \rightarrow 0$, all the terms containing Δx are removed:

$$\frac{\partial v(x,t)}{\partial x} = Ri(x,t) + L\frac{\partial i(x,t)}{\partial t} \quad \text{(the 2nd Telegrapher's equation)} \tag{J.16}$$

Now, let's differentiate (J.10) and (J.16) with respect to t and x, respectively:

$$\frac{\partial^2 i(x,t)}{\partial x \partial t} = G\frac{\partial v(x,t)}{\partial t} + C\frac{\partial^2 v(x,t)}{\partial t^2} \tag{J.17}$$

$$\frac{\partial^2 v(x,t)}{\partial x^2} = R\frac{\partial i(x,t)}{\partial x} + L\frac{\partial^2 i(x,t)}{\partial t \partial x} \tag{J.18}$$

Substituting (J.10) and (J.17) in (J.18) for $\frac{\partial i}{\partial x}$ and $\frac{\partial^2 i}{\partial t \partial x}$, respectively, yields:

$$\frac{\partial^2 v(x,t)}{\partial x^2} = R\left[Gv(x,t) + C\frac{\partial v(x,t)}{\partial t}\right] + L\left[G\frac{\partial v(x,t)}{\partial t} + C\frac{\partial^2 v(x,t)}{\partial t^2}\right] \tag{J.19}$$

Expanding (J.19) gives the 1st wave equation:

$$\frac{\partial^2 v(x,t)}{\partial x^2} = RGv(x,t) + (RC + GL)\frac{\partial v(x,t)}{\partial t} + LC\frac{\partial^2 v(x,t)}{\partial t^2} \tag{J.20}$$

To have the 2nd wave equation (i.e., i is the unknown variable), let's differentiate (J.10) and (J.16) again, but now with respect to x and t, respectively:

$$\frac{\partial^2 i(x,t)}{\partial x^2} = G\frac{\partial v(x,t)}{\partial x} + C\frac{\partial^2 v(x,t)}{\partial t \partial x} \tag{J.21}$$

$$\frac{\partial^2 v(x,t)}{\partial x \partial t} = R\frac{\partial i(x,t)}{\partial t} + L\frac{\partial^2 i(x,t)}{\partial t^2} \tag{J.22}$$

Substituting (J.16) and (J.22) in (J.21) for $\frac{\partial v}{\partial x}$ and $\frac{\partial^2 v}{\partial t \partial x}$, respectively, yields:

$$\frac{\partial^2 i(x,t)}{\partial x^2} = G\left[Ri(x,t) + L\frac{\partial i(x,t)}{\partial t}\right] + C\left[R\frac{\partial i(x,t)}{\partial t} + L\frac{\partial^2 i(x,t)}{\partial t^2}\right] \tag{J.23}$$

Expanding (J.23) gives the 2nd wave equation:

$$\frac{\partial^2 i(x,t)}{\partial x^2} = RGi(x,t) + (RC + GL)\frac{\partial i(x,t)}{\partial t} + LC\frac{\partial^2 i(x,t)}{\partial t^2} \tag{J.24}$$

As can be clearly seen from (J.20) and (J.24), both v and i have the same equation. Therefore, the general form of the wave equation is:

$$\frac{\partial^2 u(x,t)}{\partial x^2} = RGu(x,t) + (RC + GL)\frac{\partial u(x,t)}{\partial t} + LC\frac{\partial^2 u(x,t)}{\partial t^2} \tag{J.25}$$

Replacing the differential operators by subscript symbols:

$$u_{xx} = RGu + (RC + GL)\,u_t + LCu_{tt} \tag{J.26}$$

Re-arranging the terms of (J.26) and dividing them by LC:

$$u_{tt} + \frac{(RC + GL)}{LC}u_t = \frac{1}{LC}u_{xx} - \frac{RG}{LC}u \tag{J.27}$$

Adding $\Phi(x,t)$ as an external source makes (J.27) in a more general form, and the wave equation becomes inhomogeneous:

$$u_{tt} + \frac{RC+GL}{LC}u_t = \frac{1}{LC}u_{xx} - \frac{RG}{LC}u + \Phi(x,t) \tag{J.28}$$

Time-Domain to Phasor-Domain Transformation (Converting the Telegrapher's Equations from PDEs to ODEs):

Equations (J.10) and (J.16) can be solved analytically, which is preferred by mathematicians. However, the solution can be easily obtained by converting them from being partial differential equations (PDEs) to ordinary differential equations (ODEs), which is preferred by electrical power engineers and can be seen in many textbooks [49,125,149,155,326]. To apply that, let's consider v and i are steady-state sinusoidal solutions where (J.10) and (J.16) can be differentiated with respect to time t. For having a direct solution, the dependent variables v and i are expressed in a polar coordinate as follows:

$$V = \bar{V}_p \angle \omega t = \bar{V}_p e^{j\omega t} = \bar{V}_p \cos(\omega t) + j\bar{V}_p \sin(\omega t) \tag{J.29}$$

$$I = \bar{I}_p \angle \omega t = \bar{I}_p e^{j\omega t} = \bar{I}_p \cos(\omega t) + j\bar{I}_p \sin(\omega t) \tag{J.30}$$

The variables v and i can be represented by any parts of the previous two equations. If the imaginary parts are selected for v and i:

$$v(t) = \Im\{V\} = V_p \sin(\omega t + \theta_v) \tag{J.31}$$

$$i(t) = \Im\{I\} = I_p \sin(\omega t + \theta_i) \tag{J.32}$$

where V_p and I_p are respectively the peak voltage and current, and they can be determined either from their sinusoidal plots or if the effective or root-mean square values V_{rms} and I_{rms} are known.

From (J.29) and (J.30), if $\bar{V}_p$ and $\bar{I}_p$ are given, then the peak voltage and current can be directly obtained as follows:

$$\bar{V}_p = \left|\bar{V}_p\right| e^{j\theta_v} = V_p e^{j\theta_v} \;\Rightarrow\; \therefore\; V_p = \left|\bar{V}_p\right| \tag{J.33}$$

$$\bar{I}_p = \left|\bar{I}_p\right| e^{j\theta_i} = I_p e^{j\theta_i} \;\Rightarrow\; \therefore\; I_p = \left|\bar{I}_p\right| \tag{J.34}$$

By referring to Figure 6.7, the per-unit-length impedance z and admittance y are calculated as follows:

$$z = R + j\omega L \tag{J.35}$$

$$y_1 = y_2 = \frac{y}{2} = \frac{G}{2} + j\frac{\omega C}{2} \tag{J.36}$$

Now, to convert PDEs to ODEs, just use z and y instead of R, L, G and C as follows:
Applying KCL at node P:

$$I(x + \Delta x) = I_{y_1} + I_L \tag{J.37}$$

Applying Ohm's law to node P:

$$I_{y_1} = \frac{y}{2}\Delta x \; V(x + \Delta x) \tag{J.38}$$

Applying KCL at node Q:

$$I_L = I_{y_2} + I(x) \tag{J.39}$$

Applying Ohm's law to node P:

$$I_{y_2} = \frac{y}{2}\Delta x \; V(x) \tag{J.40}$$

Substituting (J.40) in (J.39) for I_{y_2} yields:

$$I_L = \frac{y}{2}\Delta x \; V(x) + I(x) \tag{J.41}$$

Now, substituting (J.38) and (J.41) in (J.37) for I_{y_1} and I_L, respectively, yields:

$$I(x + \Delta x) = \frac{y}{2}\Delta x \; V(x + \Delta x) + \frac{y}{2}\Delta x \; V(x) + I(x) \tag{J.42}$$

Re-arranging (J.42) gives:

$$\frac{I(x + \Delta x) - I(x, t)}{\Delta x} = \frac{y}{2}\left[V(x + \Delta x) + V(x)\right] \tag{J.43}$$

As $\Delta x \rightarrow 0$, the left-hand side becomes an ordinary derivative of I with respect to x[a]:

$$\frac{dI(x)}{dx} = y \; V(x) \quad \text{(the 1}^{\text{st}}\text{ ODE Telegrapher's equation)} \tag{J.44}$$

[a] It is similar to (J.10) but without the time variable t.

For the 2nd ODE Telegrapher's equation, let's apply KVL in the middle closed-loop of Figure 6.7:

$$V(x+\Delta x) = z\ \Delta x\ I_L + V(x) \tag{J.45}$$

Substituting (J.41) in (J.45) for I_L, yields:

$$\begin{aligned} V(x+\Delta x) &= z\ \Delta x\ \left[\frac{y}{2}\Delta x\ V(x) + I(x)\right] + V(x) \\ &= \frac{zy}{2}V(x)\,(\Delta x)^2 + z\ I(x)\Delta x + V(x) \end{aligned} \tag{J.46}$$

Re-arranging (J.46) gives:

$$\frac{V(x+\Delta x) - V(x)}{\Delta x} = \frac{z\ y}{2}V(x)\cancel{\Delta x} + z\ I(x) \tag{J.47}$$

As $\Delta x \ \rightarrow\ 0$, the left-hand side becomes an ordinary derivative of v with respect to x, and all the terms multiplied by Δx are removed:

$$\frac{dV(x)}{dx} = z\ I(x) \quad \text{(the 2nd ODE Telegrapher's equation)} \tag{J.48}$$

By differentiating (J.48) with respect to x and substituting for $\frac{dI(x)}{dx}$ from (J.44) gives:

$$\frac{d^2V(x)}{dx^2} = z\frac{dI(x)}{dx} = z\ y\ V(x) \tag{J.49}$$

Let:

$$\gamma_v^2 = z\ y\ \rightarrow\ \gamma_v\text{: propagation constant with respect to voltage } [\text{m}^{-1}] \tag{J.50}$$

Therefore, the following 2nd order ODE differential equation can be obtained:

$$\frac{d^2V(x)}{dx^2} - \gamma_v^2 V(x) = 0 \tag{J.51}$$

The solution of the above equation can be easily obtained as follows:

$$V(x) = A_1 e^{\gamma_v x} + A_2 e^{-\gamma_v x} \tag{J.52}$$

Again, by differentiating (J.44) with respect to x and substituting for $\frac{dV(x)}{dx}$ from (J.48) gives:

$$\frac{d^2I(x)}{dx^2} = y\frac{dV(x)}{dx} = y\ z\ I(x) \tag{J.53}$$

Let:

$$\gamma_i^2 = y\ z\ \rightarrow\ \gamma_i\text{: propagation constant with respect to current } [\mathrm{m}^{-1}] \tag{J.54}$$

Therefore, the following 2nd order ODE differential equation will result:

$$\frac{d^2 I(x)}{dx^2} - \gamma_i^2 I(x) = 0 \tag{J.55}$$

As with (J.52), the solution of the above equation can be easily obtained as follows:

$$I(x) = A_3 e^{\gamma_i x} + A_4 e^{-\gamma_i x} \tag{J.56}$$

The discrimination between γ_v and γ_i is very important when dealing with matrices of $[z]$ and $[y]$; as when the 3ϕ system is under abnormal operation (i.e, unsteady-state or unstable condition):

$$[\gamma_v] \neq [\gamma_i]\ \rightarrow\ [z]\,[y] \neq [y]\,[z] \tag{J.57}$$

This note can be found in transient power systems analysis references, such as that summarized in [49]. But, because this study considers a stable 3ϕ system, so the entire system can be represented as a 1ϕ system using the positive sequence network. This gives the following simplification:

$$\therefore\ \gamma = \gamma_v = \gamma_i = \sqrt{z\ y} = \sqrt{y\ z}\ \rightarrow\ \text{propagation constant } [\mathrm{m}^{-1}] \tag{J.58}$$

and γ can also be calculated as follows:

$$\gamma = \bar{\alpha} + j\bar{\beta} = \sqrt{z\ y} = \sqrt{(R + j\omega L)\,(G + j\omega C)} \tag{J.59}$$

where the real part, $\bar{\alpha}$, is known as the attenuation constant, and the imaginary part, $\bar{\beta}$, is known as the phase constant [326].

If (J.58) is the case, then no need to take the second derivative of $I(x)$ to solve the current telegraph equation. Instead, it can be obtained directly by differentiating (J.52) and then substituting it into (J.48) for $\frac{dV}{dx}$. To do that, first (J.48) is re-arranged for $I(x)$ as follows:

$$I(x) = \frac{1}{z}\frac{dV(x)}{dx} \tag{J.60}$$

Differentiating (J.52) with respect to x gives:

$$\frac{dV(x)}{dx} = A_1\gamma e^{\gamma x} - A_2\gamma e^{-\gamma x} \tag{J.61}$$

Substituting (J.61) in (J.60) for $\frac{dV(x)}{dx}$ yields:

$$I(x) = \frac{\gamma}{z}\left(A_1 e^{\gamma x} - A_2 e^{-\gamma x}\right) \tag{J.62}$$

From (J.59):

$$\frac{\gamma}{z} = \frac{\sqrt{z\,y}}{z} = \frac{\cancel{\sqrt{z}}\sqrt{y}}{\cancel{\sqrt{z}}\sqrt{z}} = \frac{\sqrt{y}}{\sqrt{z}} = \sqrt{\frac{y}{z}} \tag{J.63}$$

Substituting (J.63) in (J.62) for $\frac{\gamma}{z}$ yields:

$$\therefore\ I(x) = \sqrt{\frac{y}{z}}\left(A_1 e^{\gamma x} - A_2 e^{-\gamma x}\right) = \frac{1}{Z_c}\left(A_1 e^{\gamma x} - A_2 e^{-\gamma x}\right) \tag{J.64}$$

where Z_c is called the wave or characteristic impedance of the transmission line [125], which is given by:

$$Z_c = \sqrt{\frac{z}{y}} \tag{J.65}$$

and thus the characteristic admittance Y_c is the inverse of Z_c [49]:

$$Y_c = \frac{1}{Z_c} = \sqrt{\frac{y}{z}} \tag{J.66}$$

Therefore, comparing the constants of (J.64) with that obtained earlier in (J.56) gives the following:

$$A_3 = \sqrt{\frac{y}{z}}A_1 = \frac{A_1}{Z_c} \tag{J.67}$$

$$A_4 = -\sqrt{\frac{y}{z}}A_2 = \frac{-A_2}{Z_c} \tag{J.68}$$

Now, the first boundary condition[2] is used in (J.52) and (J.64) to find A_1 and A_2 as follows:

$$V_R = V(0) = A_1 \cancel{e^{0}} + A_2 \cancel{e^{-0}} = A_1 + A_2 \tag{J.69}$$

$$I_R = I(0) = \frac{A_1}{Z_c}\cancel{e^{0}} - \frac{A_2}{Z_c}\cancel{e^{-0}} = \frac{A_1}{Z_c} - \frac{A_2}{Z_c} \tag{J.70}$$

These two equations have two unknowns, which can be easily determined by substitution, elimination, or augmented matrix. If the last method is used, then:

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \frac{1}{Z_c} & \frac{-1}{Z_c} \end{bmatrix}^{-1} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & Z_c \\ 1 & -Z_c \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} \frac{V_R + Z_c I_R}{2} \\ \frac{V_R - Z_c I_R}{2} \end{bmatrix} \tag{J.71}$$

Substituting (J.71) in (J.52) and (J.64) for A_1 and A_2, respectively, yields:

$$V(x) = \frac{V_R + Z_c I_R}{2} e^{\gamma x} + \frac{V_R - Z_c I_R}{2} e^{-\gamma x} \tag{J.72}$$

$$V(x) = \frac{\frac{V_R}{Z_c} + I_R}{2} e^{\gamma x} - \frac{\frac{V_R}{Z_c} - I_R}{2} e^{-\gamma x} \tag{J.73}$$

To have a two-port network, V_R and I_R should be separated as follows:

$$V(x) = \left(\frac{e^{\gamma x} + e^{-\gamma x}}{2}\right) V_R + Z_c \left(\frac{e^{\gamma x} - e^{-\gamma x}}{2}\right) I_R \tag{J.74}$$

$$I(x) = \frac{1}{Z_c}\left(\frac{e^{\gamma x} - e^{-\gamma x}}{2}\right) V_R + \left(\frac{e^{\gamma x} + e^{-\gamma x}}{2}\right) I_R \tag{J.75}$$

Recalling that:

$$\sinh(\theta) = \frac{e^{\theta} - e^{-\theta}}{2} \tag{J.76}$$

$$\cosh(\theta) = \frac{e^{\theta} + e^{-\theta}}{2} \tag{J.77}$$

Thus, (J.74) and (J.75) can be transformed from exponential forms to hyperbolic forms as follows:

$$V(x) = \cosh(\gamma x)\, V_R + Z_c \sinh(\gamma x)\, I_R \tag{J.78}$$

$$I(x) = \frac{1}{Z_c}\sinh(\gamma x)\, V_R + \cosh(\gamma x)\, I_R \tag{J.79}$$

After finding A_1 and A_2, the sending-end voltage and current can be determined using the second boundary condition (i.e., $x = \mathfrak{L}$) as follows:

$$V_S = V(\mathfrak{L}) = \cosh(\gamma \mathfrak{L})\, V_R + Z_c \sinh(\gamma \mathfrak{L})\, I_R \tag{J.80}$$

$$I_S = I(\mathfrak{L}) = \frac{1}{Z_c}\sinh(\gamma \mathfrak{L})\, V_R + \cosh(\gamma \mathfrak{L})\, I_R \tag{J.81}$$

[2] i.e., $x = 0$, which happens at the receiving-end of the transmission line.

Re-expressing (J.80) and (J.81) in a matrix form gives:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \cosh(\gamma\mathfrak{L}) & Z_c \sinh(\gamma\mathfrak{L}) \\ \frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{J.82}$$

As considered in the preceding transmission line models, the voltage and current at the sending terminal are supposed to be known by the measurement devices mounted on the generator bus. Thus, it is better to re-express (J.82) in terms of V_R and I_R as follows:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \frac{1}{AD - BC} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{J.83}$$

The first and second terms of the determinant are:

$$AD = \cosh^2(\gamma\mathfrak{L}) \tag{J.84}$$

$$BC = \sinh^2(\gamma\mathfrak{L}) \tag{J.85}$$

One of the hyperbolic relations is:

$$\cosh^2(\gamma\mathfrak{L}) - \sinh^2(\gamma\mathfrak{L}) = 1 \tag{J.86}$$

This means that the determinant is unity:

$$AD - BC = 1 \tag{J.87}$$

Therefore, (J.83) becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \cosh(\gamma\mathfrak{L}) & -Z_c \sinh(\gamma\mathfrak{L}) \\ \frac{-1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix} \tag{J.88}$$

K. Deriving the Equivalent T-Model for the Long Line's ABCD Parameters

To find the equivalent T-model, first the ABCD parameters of the long line model, given in (6.9), are equated with that of the T-model, given in (6.7), as follows:

$$\begin{bmatrix} \cosh(\gamma\mathfrak{L}) & Z_c \sinh(\gamma\mathfrak{L}) \\ \frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} = \begin{bmatrix} 1+\frac{\hat{Z}\hat{Y}}{2} & \hat{Z}\left(1+\frac{\hat{Z}\hat{Y}}{4}\right) \\ \hat{Y} & 1+\frac{\hat{Z}\hat{Y}}{2} \end{bmatrix} \tag{K.1}$$

Comparing between As or Ds:

$$\cosh(\gamma\mathfrak{L}) = 1 + \frac{\hat{Z}\hat{Y}}{2} \tag{K.2}$$

By separating $\frac{\hat{Z}}{2}$:

$$\frac{\hat{Z}}{2} = \frac{1}{\hat{Y}}\left[\cosh(\gamma\mathfrak{L}) - 1\right] \tag{K.3}$$

Comparing between Cs:

$$\frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) = \hat{Y} \tag{K.4}$$

By referring to (J.58), (J.63), (J.65) and (J.66):

$$Z_c = \frac{1}{Y_c} = \sqrt{\frac{z}{y}} = \frac{z}{\sqrt{yz}} = \frac{z}{\gamma}\cdot\frac{\mathfrak{L}}{\mathfrak{L}} = \frac{Z}{\gamma\mathfrak{L}} \tag{K.5}$$

$$Y_c = \frac{1}{Z_c} = \sqrt{\frac{y}{z}} = \frac{y}{\sqrt{zy}} = \frac{y}{\gamma}\cdot\frac{\mathfrak{L}}{\mathfrak{L}} = \frac{Y}{\gamma\mathfrak{L}} \tag{K.6}$$

Thus, (K.4) becomes:

$$\therefore\ \hat{Y} = \frac{\gamma\mathfrak{L}}{Z}\sinh(\gamma\mathfrak{L}) = \frac{Y}{\gamma\mathfrak{L}}\sinh(\gamma\mathfrak{L}) = \boxed{Y\frac{\sinh(\gamma\mathfrak{L})}{\gamma\mathfrak{L}}} \tag{K.7}$$

Substituting (K.7) in (K.3) for $\hat{Y}$ yields:

$$\frac{\hat{Z}}{2} = \frac{\cosh(\gamma\mathfrak{L})-1}{\left(\frac{\gamma\mathfrak{L}}{Z}\right)\sinh(\gamma\mathfrak{L})} = \frac{Z}{\gamma\mathfrak{L}}\left[\frac{\cosh(\gamma\mathfrak{L})-1}{\sinh(\gamma\mathfrak{L})}\right] \tag{K.8}$$

Re-calling:

$$\tanh\left(\frac{\theta}{2}\right) = \frac{\cosh(\theta)-1}{\sinh(\theta)} \tag{K.9}$$

Substituting (K.9) in (K.8) yields:

$$\frac{\hat{Z}}{2} = \frac{Z}{\gamma\mathfrak{L}}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right) = \boxed{\frac{Z}{2}\cdot\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)}} \tag{K.10}$$

L. Deriving the Equivalent Π-Model for the Long Line's ABCD Parameters

To find the equivalent Π-model, first the ABCD parameters of the long line model, given in (6.9), are equated with that of the Π-model, given in (6.8), as follows:

$$\begin{bmatrix} \cosh(\gamma\mathfrak{L}) & Z_c \sinh(\gamma\mathfrak{L}) \\ \frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} = \begin{bmatrix} 1+\frac{\hat{Z}\hat{Y}}{2} & \hat{Z} \\ \hat{Y}\left(1+\frac{\hat{Z}\hat{Y}}{4}\right) & 1+\frac{\hat{Z}\hat{Y}}{2} \end{bmatrix} \tag{L.1}$$

Comparing between As or Ds:

$$\cosh(\gamma\mathfrak{L}) = 1 + \frac{\hat{Z}\hat{Y}}{2} \tag{L.2}$$

By separating $\frac{\hat{Y}}{2}$:

$$\frac{\hat{Y}}{2} = \frac{1}{\hat{Z}}\left[\cosh(\gamma\mathfrak{L}) - 1\right] \tag{L.3}$$

Comparing between Bs:

$$Z_c \sinh(\gamma\mathfrak{L}) = \hat{Z} \tag{L.4}$$

Substituting (K.5) in (L.4) for Z_c yields:

$$\therefore \quad \boxed{\hat{Z} = Z\frac{\sinh(\gamma\mathfrak{L})}{\gamma\mathfrak{L}}} \tag{L.5}$$

Substituting (L.5) in (L.3) for $\hat{Z}$ yields:

$$\frac{\hat{Y}}{2} = \frac{\cosh(\gamma\mathfrak{L}) - 1}{\left(\frac{Z}{\gamma\mathfrak{L}}\right)\sinh(\gamma\mathfrak{L})} = \frac{\cosh(\gamma\mathfrak{L}) - 1}{\left(\frac{\gamma\mathfrak{L}}{Y}\right)\sinh(\gamma\mathfrak{L})} = \frac{Y}{\gamma\mathfrak{L}} \cdot \frac{\cosh(\gamma\mathfrak{L}) - 1}{\sinh(\gamma\mathfrak{L})} \tag{L.6}$$

Applying the half argument formula given in (K.9) to the hyperbolic terms of (L.6) yields:

$$\therefore \quad \frac{\hat{Y}}{2} = \frac{Y}{\gamma\mathfrak{L}}\tanh\left(\frac{Y}{\gamma 2}\right) = \boxed{\frac{Y}{2} \cdot \frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)}} \tag{L.7}$$

M. Deriving the Equivalent M-Model for the Long Line's ABCD Parameters

The equivalent M-model can be constructed by equating the ABCD parameters of (6.9) with that of (6.12) as follows:

$$\begin{bmatrix} \cosh(\gamma\mathfrak{L}) & Z_c\sinh(\gamma\mathfrak{L}) \\ \frac{1}{Z_c}\sinh(\gamma\mathfrak{L}) & \cosh(\gamma\mathfrak{L}) \end{bmatrix} = \begin{bmatrix} 1+\frac{\hat{Z}\hat{Y}}{6}\left(3+\frac{\hat{Z}\hat{Y}}{6}\right) & \hat{Z}\left(1+\frac{\hat{Z}\hat{Y}}{12}\right) \\ \hat{Y}+\frac{\hat{Z}\hat{Y}^2}{9}\left(2+\frac{\hat{Z}\hat{Y}}{12}\right) & 1+\frac{\hat{Z}\hat{Y}}{6}\left(3+\frac{\hat{Z}\hat{Y}}{6}\right) \end{bmatrix} \tag{M.1}$$

Comparing between As or Ds:

$$\begin{aligned} \cosh(\gamma\mathfrak{L}) &= 1+\frac{\hat{Z}\hat{Y}}{6}\left(3+\frac{\hat{Z}\hat{Y}}{6}\right) = 1+\frac{\hat{Z}\hat{Y}}{2}+\frac{\hat{Z}^2\hat{Y}^2}{36} \\ 36\cosh(\gamma\mathfrak{L}) &= 36+18\hat{Z}\hat{Y}+\hat{Z}^2\hat{Y}^2 \end{aligned} \tag{M.2}$$

Comparing between Bs:

$$\begin{aligned} Z_c\sinh(\gamma\mathfrak{L}) &= \hat{Z}\left(1+\frac{\hat{Z}\hat{Y}}{12}\right) \\ 12Z_c\sinh(\gamma\mathfrak{L}) &= 12\hat{Z}+\hat{Z}^2\hat{Y} \end{aligned} \tag{M.3}$$

From (M.3):

$$\hat{Z}^2 = \frac{1}{\hat{Y}}\left[12Z_c\sinh(\gamma\mathfrak{L})-12\hat{Z}\right] = \frac{12}{\hat{Y}}\left[Z_c\sinh(\gamma\mathfrak{L})-\hat{Z}\right] \tag{M.4}$$

Substituting (M.4) in (M.2) for $\hat{Z}^2$ yields:

$$36\cosh(\gamma\mathfrak{L}) = 36+18\hat{Z}\hat{Y}+\frac{12\hat{Y}^{\not 2}}{\not{\hat{Y}}}\left[Z_c\sinh(\gamma\mathfrak{L})-\hat{Z}\right]$$

$$\begin{aligned} 36\cosh(\gamma\mathfrak{L})-36 &= 18\hat{Z}\hat{Y}+12\hat{Y}Z_c\sinh(\gamma\mathfrak{L})-12\hat{Y}\hat{Z} \\ &= 6\hat{Z}\hat{Y}+12\hat{Y}Z_c\sinh(\gamma\mathfrak{L}) \end{aligned}$$

$$^{6}\not{36}\left[\cosh(\gamma\mathfrak{L})-1\right] = \not{6}\left[\hat{Z}+2Z_c\sinh(\gamma\mathfrak{L})\right]\hat{Y}$$

$$\therefore\ \hat{Y} = \frac{6\left[\cosh(\gamma\mathfrak{L})-1\right]}{2Z_c\sinh(\gamma\mathfrak{L})+\hat{Z}} \tag{M.5}$$

Substituting (M.5) in (M.3) for $\hat{Y}$ yields:

$$^{2}\not{12}Z_c\sinh(\gamma\mathfrak{L}) =^{2} \not{12}\hat{Z}+\frac{\not{6}\left[\cosh(\gamma\mathfrak{L})-1\right]\hat{Z}^2}{2Z_c\sinh(\gamma\mathfrak{L})+\hat{Z}}$$

$$\left[2Z_c \sinh\left(\gamma\mathfrak{L}\right) - 2\hat{Z}\right]\left[2Z_c \sinh\left(\gamma\mathfrak{L}\right) + \hat{Z}\right] = \left[\cosh\left(\gamma\mathfrak{L}\right) - 1\right]\hat{Z}^2$$

$$4Z_c^2 \sinh^2\left(\gamma\mathfrak{L}\right) + 2Z_c \sinh\left(\gamma\mathfrak{L}\right)\hat{Z} - 4Z_c \sinh\left(\gamma\mathfrak{L}\right)\hat{Z} - 2\hat{Z}^2 = \cosh\left(\gamma\mathfrak{L}\right)\hat{Z}^2 - \hat{Z}^2$$

$$4Z_c^2 \sinh^2\left(\gamma\mathfrak{L}\right) - 2Z_c \sinh\left(\gamma\mathfrak{L}\right)\hat{Z} = \left[\cosh\left(\gamma\mathfrak{L}\right) + 1\right]\hat{Z}^2$$

$$\left[\cosh\left(\gamma\mathfrak{L}\right) + 1\right]\hat{Z}^2 + 2Z_c \sinh\left(\gamma\mathfrak{L}\right)\hat{Z} - 4Z_c^2 \sinh^2\left(\gamma\mathfrak{L}\right) = 0$$

$$\hat{Z}^2 + 2Z_c \frac{\sinh\left(\gamma\mathfrak{L}\right)}{\cosh\left(\gamma\mathfrak{L}\right) + 1}\hat{Z} - 4Z_c^2 \frac{\sinh^2\left(\gamma\mathfrak{L}\right)}{\cosh\left(\gamma\mathfrak{L}\right) + 1} = 0 \tag{M.6}$$

Re-calling:

$$\tanh\left(\frac{\theta}{2}\right) = \frac{\sinh\left(\theta\right)}{\cosh\left(\theta\right) + 1} \tag{M.7}$$

$$2\sinh^2\left(\frac{\theta}{2}\right) = \frac{\sinh^2\left(\theta\right)}{\cosh\left(\theta\right) + 1} \tag{M.8}$$

Thus, (M.6) becomes:

$$\hat{Z}^2 + 2Z_c \tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\hat{Z} - 8Z_c^2 \sinh^2\left(\frac{\gamma\mathfrak{L}}{2}\right) = 0 \tag{M.9}$$

Substituting (K.5) in (M.9) for Z_c with some arrangements yields:

$$\hat{Z}^2 + Z\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)}\hat{Z} - 2Z^2\frac{\sinh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)^2} = 0 \tag{M.10}$$

As can be clearly seen, it is a quadratic equation. Thus, for $a\hat{Z}^2 + b\hat{Z} + C = 0$, there are two possible solutions:

$$\hat{Z} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \begin{cases} \frac{-b+\sqrt{b^2-4ac}}{2a}, & \text{solution no.1} \ \checkmark \\ \frac{-b-\sqrt{b^2-4ac}}{2a}, & \text{solution no.2} \ \boldsymbol{\times} \end{cases} \tag{M.11}$$

By testing both solutions with realistic parameters and $\mathfrak{L} = 1 \rightarrow 10,000$ km, it has been found that the first solution always gives positive resistance and inductive reactance while the other one always gives negative values. Based on that, the second solution is rejected, and thus:

$$\begin{aligned} \hat{Z} &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{-Z\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)} + \sqrt{\frac{\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)Z^2}{\left(\frac{\gamma\mathfrak{L}}{2}\right)^2} + \frac{\sinh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)8Z^2}{\left(\frac{\gamma\mathfrak{L}}{2}\right)^2}}}{2} \\ &= \frac{-Z}{\gamma\mathfrak{L}}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right) + \frac{Z}{\gamma\mathfrak{L}}\sqrt{\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right) + 8\sinh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)} \\ &= \left(\frac{-Z}{\gamma\mathfrak{L}}\right)\left[\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right) - \sqrt{\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right) + 8\sinh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)}\right] \end{aligned} \tag{M.12}$$

Applying (M.7) and (M.8) to (M.12) gives:

$$\begin{aligned}\hat{Z} &= \left(\frac{-Z}{\gamma\mathfrak{L}}\right)\left\{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)-\sqrt{\frac{\sinh^2(\gamma\mathfrak{L})}{[\cosh(\gamma\mathfrak{L})+1]^2}+4\frac{\sinh^2(\gamma\mathfrak{L})[\cosh(\gamma\mathfrak{L})+1]}{[\cosh(\gamma\mathfrak{L})+1]^2}}\right\}\\ &= \left(\frac{-Z}{\gamma\mathfrak{L}}\right)\left\{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)-\frac{\sinh(\gamma\mathfrak{L})}{\cosh(\gamma\mathfrak{L})+1}\sqrt{1+4[\cosh(\gamma\mathfrak{L})+1]}\right\}\\ &= \left(\frac{-Z}{\gamma\mathfrak{L}}\right)\left[\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)-\frac{\sinh(\gamma\mathfrak{L})}{\cosh(\gamma\mathfrak{L})+1}\sqrt{1+4\frac{\sinh(\gamma\mathfrak{L})}{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}}\right]\end{aligned} \tag{M.13}$$

Applying (K.9) and (M.7) to (M.13) gives:

$$\begin{aligned}\hat{Z} &= \left(\frac{-Z}{\gamma\mathfrak{L}}\right)\left[\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)-\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\sqrt{1+4\frac{\sinh(\gamma\mathfrak{L})}{\frac{\cosh(\gamma\mathfrak{L})-1}{\sinh(\gamma\mathfrak{L})}}}\right]\\ &= \frac{-Z}{\gamma\mathfrak{L}}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{1+4\frac{\sinh^2(\gamma\mathfrak{L})}{\cosh(\gamma\mathfrak{L})-1}}\right]\end{aligned} \tag{M.14}$$

Re-calling:

$$\frac{\sinh^2(\theta)}{\cosh(\theta)-1}=2\cosh^2\left(\frac{\theta}{2}\right) \tag{M.15}$$

Thus, (M.15) becomes:

$$\therefore\ \hat{Z}=\frac{-Z}{\gamma\mathfrak{L}}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{1+8\cosh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)}\right] \tag{M.16}$$

Re-calling:

$$\cosh(\theta)+1=2\cosh^2\left(\frac{\theta}{2}\right) \tag{M.17}$$

Thus, (M.17) becomes:

$$\therefore\ \hat{Z}=\frac{-Z}{\gamma\mathfrak{L}}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh(\gamma\mathfrak{L})}\right] \tag{M.18}$$

$$\therefore\ \boxed{\frac{\hat{Z}}{2}=\frac{-Z}{4}\cdot\frac{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)}\left[1-\sqrt{5+4\cosh(\gamma\mathfrak{L})}\right]} \tag{M.19}$$

By using (K.5) and (K.6), the term $\frac{Z}{\gamma\mathfrak{L}}$ of (M.18) can be replaced with $\frac{\gamma\mathfrak{L}}{Y}$ as follows:

$$\hat{Z}=\frac{-\gamma\mathfrak{L}}{Y}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh(\gamma\mathfrak{L})}\right] \tag{M.20}$$

Multiplying (M.20) by itself to find $\hat{Z}^2$:

$$\begin{aligned}\hat{Z}^2 &= \left\{\frac{-\gamma\mathfrak{L}}{Y}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]\right\}^2 \\ &= \left(\frac{\gamma\mathfrak{L}}{Y}\right)^2\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]^2\end{aligned}$$

where

$$\begin{aligned}\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]^2 &= 1-2\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+5+4\cosh(\gamma\mathfrak{L}) \\ &= 2\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]\end{aligned}$$

$$\therefore\ \hat{Z}^2 = 2\left(\frac{\gamma\mathfrak{L}}{Y}\right)^2\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right] \tag{M.21}$$

Re-expressing (M.4) by taking $\hat{Y}$ as the subject of the equation gives:

$$\hat{Y} = \frac{12}{\hat{Z}^2}\left[Z_c\sinh\left(\gamma\mathfrak{L}\right)-\hat{Z}\right] = \frac{12}{\hat{Y}}\left[Z_c\sinh\left(\gamma\mathfrak{L}\right)-\hat{Z}\right] \tag{M.22}$$

Substituting (M.20) and (M.21) in (M.22) for $\hat{Z}$ and $\hat{Z}^2$, respectively, yields:

$$\therefore\ \hat{Y} = \frac{12Z_c\sinh\left(\gamma\mathfrak{L}\right)+12\left(\frac{\gamma\mathfrak{L}}{Y}\right)\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]}{2\left(\frac{\gamma\mathfrak{L}}{Y}\right)^2\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]} \tag{M.23}$$

Substituting (K.6) in (M.23) for Z_c and dividing both sides of the equation by 3:

$$\begin{aligned}\frac{\hat{Y}}{3} &= \frac{{}^{2}\cancel{12}\left\{\cancel{\left(\frac{\gamma\mathfrak{L}}{Y}\right)}\sinh\left(\gamma\mathfrak{L}\right)+\cancel{\left(\frac{\gamma\mathfrak{L}}{Y}\right)}\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]\right\}}{\cancel{6}\left(\frac{\gamma\mathfrak{L}}{Y}\right)^{\cancel{2}}\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]} \\ &= \frac{2\left\{\sinh\left(\gamma\mathfrak{L}\right)+\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]\right\}}{\left(\frac{\gamma\mathfrak{L}}{Y}\right)\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]}\end{aligned} \tag{M.24}$$

Applying (M.7) to (M.24) gives:

$$\begin{aligned}\frac{\hat{Y}}{3} &= \frac{2\left\{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[\cosh\left(\gamma\mathfrak{L}\right)+1\right]+\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]\right\}}{\left(\frac{\gamma\mathfrak{L}}{Y}\right)\tanh^2\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]} \\ &= \frac{2\left(\frac{Y}{\gamma\mathfrak{L}}\right)\cancel{\tanh\left(\frac{\gamma\mathfrak{L}}{2}\right)}\left\{\left[\cosh\left(\gamma\mathfrak{L}\right)+1\right]+\left[1-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}\right]\right\}}{\tanh^{\cancel{2}}\left(\frac{\gamma\mathfrak{L}}{2}\right)\left[3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})\right]}\end{aligned}$$

$$\therefore\ \boxed{\frac{\hat{Y}}{3} = Y\frac{\coth\left(\frac{\gamma\mathfrak{L}}{2}\right)}{\left(\frac{\gamma\mathfrak{L}}{2}\right)}\left[\frac{2-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+\cosh(\gamma\mathfrak{L})}{3-\sqrt{5+4\cosh\left(\gamma\mathfrak{L}\right)}+2\cosh(\gamma\mathfrak{L})}\right]} \tag{M.25}$$

N. Deriving the Equation of Permeability of Air

To express the permeability of air μ_{air} as a function of temperature, pressure, humidity and vacuum wave-number, the relationship between μ_0 and the permittivity of free space ε_0 is governed by the following formula [253]:

$$c_0 = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \tag{N.1}$$

where c_0 is the speed of light in free space or vacuum.
Re-arranging (N.1) gives:

$$\mu_0 = \frac{1}{c_0^2\ \varepsilon_0} \tag{N.2}$$

By analogy, in air, (N.2) could be realized as:

$$\mu_{\text{air}} = \frac{1}{c_{\text{air}}^2\ \varepsilon_{\text{air}}} \tag{N.3}$$

where c_{air} is the speed of light in air, which can be calculated as follows:

$$c_{\text{air}} = \frac{c_o}{n_{\text{air}}} \tag{N.4}$$

where n_{air} is the refractive index of air.
Two possible approaches could be used to obtain n_{air}:

1. Approximating it to a fixed number, which is around $n_{\text{air}} = 1.000293$ [204].
2. Expressing it as a function of temperature and pressure.

If the first approach is adopted, then the speed of light in air can be easily obtained by (N.4) as follows:

$$c_{\text{air}} = \frac{c_o}{n_{\text{air}}} = \frac{299792458}{1.000293} = 299704644.53915 \text{ m/s}$$

The permittivity of air can be mathematically expressed as a function of temperature, pressure and relative humidity as follows [176]:

$$\varepsilon_{\text{air}} = \varepsilon_0 \left[1 + \frac{1.5826}{\breve{T}_{\text{abs}}} \left(\mathcal{P}_{\text{ma}} + \frac{0.36\mathcal{P}_{\text{sw}}}{\breve{T}_{\text{abs}}}\text{RH}\right) \times 10^{-6}\right] \quad \text{(F/m)} \tag{N.5}$$

where $\mathcal{P}_{\text{ma}}$ and $\mathcal{P}_{\text{sw}}$ are respectively the pressure of moist air (in Pa) and the pressure of saturated water vapour (in Pa). The symbols $\breve{T}_{\text{abs}}$ and RH are the absolute temperature (in K) and the relative humidity (in %). Finally, ε_0 is the permittivity of free space, which can be approximated to $\varepsilon_0 \approx 8.854187817620389 \times 10^{-12}$ F/m [253].

Thus, substituting (N.5) in (N.3) for ε_{air} yields:

$$\therefore \quad \boxed{\mu_{\text{air}} = \frac{1}{795308.5963455703\left[1+\frac{1.5826}{\breve{T}_{\text{abs}}}\left(\mathcal{P}_{\text{ma}}+\frac{0.36\mathcal{P}_{\text{sw}}}{\breve{T}_{\text{abs}}}\text{RH}\right)\times 10^{-6}\right]}} \tag{N.6}$$

The more complicated way to find μ_{air} is by considering n_{air} as a function, and thus it is not constant anymore. To do that, first, the dispersion formula for standard air is expressed as follows [120]:

$$\left(n_{\text{air}}-1\right)_s \times 10^8 = 8342.13 + \frac{2406030}{130-\nu^2} + \frac{15997}{38.9-\nu^2} \tag{N.7}$$

where the subscript s means "standard air" and ν is the vacuum wave-number measured in μm^{-1}.

The preceding equation is independent of atmospheric conditions. A more realized dispersion formula is given as follows:

$$\left(n_{\text{air}}-1\right)_{\breve{T}_a,\mathcal{P}_a} = \left(n_{\text{air}}-1\right)_s \times \frac{0.00138823\mathcal{P}_a}{1+0.003671\breve{T}_a} \tag{N.8}$$

where $\breve{T}_a$ and $\mathcal{P}_a$ are respectively the ambient temperature (in °C) and the atmospheric pressure (in torr).

Based on the fact that standard air contains about 0.0003 parts of CO_2 per volume and the refractivity of CO_2 is about 50% higher than that of standard air, so the refractivity of CO_2-content air is more realized. The effect of carbon dioxide in air can be considered by using the following formula [120]:

$$\frac{\left(n_{\text{air}_{CO_2}}-n_{\text{air}_s}\right)}{\left(n_{\text{air}}-1\right)_s} = 0.5291 + 0.00360\nu \tag{N.9}$$

Because the effect of CO_2 is very small, so it is neglected in [120]. On the opposite side, the water vapour effect is more significant, and thus it is considered by using the following formula:

$$n_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}} - n_{\text{air}_{\breve{T}_a,\mathcal{P}_a}} = -\mathfrak{f}\left(5.722-0.0457\nu^2\right)\times 10^{-8} \tag{N.10}$$

The last equation is used to calculate the difference in refractive index of moist air (that contains $\mathfrak{f}$ torr of water vapour) and dry air at total pressure and equal temperature [120].

If the effect of CO_2 is also considered, then we have to modify the preceding formula. Let's assume that there is a linear relationship between $n_{\text{air}_{CO_2}}$ and n_{air_s}. Then, all we need is to replace the term $(n_{\text{air}}-1)_s$ with $(n_{\text{air}}-1)_{CO_2}$. To do that, first, re-arrange (N.9) by multiplying both sides by $(n_{\text{air}}-1)_s$ and then shift n_{air_s} to the right-side to have the following equation:

$$n_{\text{air}_{CO_2}} = (n_{\text{air}}-1)_s\,(0.5291+0.00360\nu)+n_{\text{air}_s} \tag{N.11}$$

Subtracting both sides by 1 and then multiplying each side by 10^8:

$$\begin{aligned}(n_{\text{air}}-1)_{CO_2}\times 10^8 &= [(n_{\text{air}}-1)_s\,(0.5291+0.00360\nu)+n_{\text{air}_s}-1]\times 10^8\\ &= (n_{\text{air}}-1)_s\,(0.5291+0.00360\nu)\times 10^8+(n_{\text{air}}-1)_s\times 10^8\\ &= (1.5291+0.00360\nu)\,(n_{\text{air}}-1)_s\times 10^8\end{aligned} \tag{N.12}$$

Substituting (N.7) in (N.12) for $(n_{\text{air}}-1)_s\times 10^8$ and multiplying both sides by 10^{-8} yields:

$$(n_{\text{air}}-1)_{CO_2} = (1.5291+0.00360\nu)\left[8342.13+\frac{2406030}{130-\nu^2}+\frac{15997}{38.9-\nu^2}\right]\times 10^{-8} \tag{N.13}$$

Now, by replacing $(n_{\text{air}}-1)_s$ with $(n_{\text{air}}-1)_{CO_2}$ in (N.8):

$$(n_{\text{air}}-1)_{\breve{T}_a,\mathcal{P}_a} = (n_{\text{air}}-1)_{CO_2}\times\frac{0.00138823\mathcal{P}_a}{1+0.003671\breve{T}_a} \tag{N.14}$$

Substituting (N.13) in (N.14) for $(n_{\text{air}}-1)_{CO_2}$ and shifting -1 to the right-side yields:

$$\begin{aligned}\therefore\ n_{\text{air}_{\breve{T}_a,\mathcal{P}_a}} &= 1+(1.5291+0.00360\nu)\left(\frac{0.00138823\mathcal{P}_a}{1+0.003671\breve{T}_a}\right)\\ &\quad\left[8342.13+\frac{2406030}{130-\nu^2}+\frac{15997}{38.9-\nu^2}\right]\times 10^{-8}\end{aligned} \tag{N.15}$$

Substituting (N.15) in (N.10) and solving it for $n_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}}$ yields:

$$\begin{aligned}\therefore\ n_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}} &= 1+\left[\frac{0.00138823\mathcal{P}_a\,(1.5291+0.00360\nu)\left(8342.13+\frac{2406030}{130-\nu^2}+\frac{15997}{38.9-\nu^2}\right)}{1+0.003671\breve{T}_a}\right.\\ &\quad\left.-\mathfrak{f}\left(5.722-0.0457\nu^2\right)\right]\times 10^{-8}\end{aligned} \tag{N.16}$$

Thus, based on any dynamic change in atmospheric condition, the precise value of $n_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}}$ can be obtained and then applied to the following equation:

$$\therefore \quad \boxed{\mu_{\text{air}} = \frac{4\pi \times 10^{-7} n^2_{\text{air}_{\breve{T}_a,\mathcal{P}_a,\mathfrak{f}}}}{1 + \frac{1.5826}{\breve{T}_{\text{abs}}}\left(\mathcal{P}_{\text{ma}} + \frac{0.36\mathcal{P}_{\text{sw}}}{\breve{T}_{\text{abs}}}\text{RH}\right) \times 10^{-6}}} \tag{N.17}$$

It has to be careful with the units used in the last equation where torr for $\mathcal{P}_a$ and $\mathfrak{f}$, Pa for $\mathcal{P}_{\text{ma}}$ and $\mathcal{P}_{\text{sw}}$, °C for $\breve{T}_a$, K for $\breve{T}_{\text{abs}}$, μm^{-1} for ν, and % for RH. To have unified units for temperature and pressure, then the following unit conversions can be used:

- Pressure Unit Conversion → between Pa[3] and torr[4]:

$$P\ [\text{Pa}] = 133.3223684211P\ [\text{torr}] \tag{N.18}$$

- Temperature Unit Conversion → between °C and K:

$$T\ [^\circ\text{C}] = 273.15 + T\ [\text{K}] \tag{N.19}$$

[3]It is also called N/m^2.
[4]It is also called mmHg.

O. Deriving the Equations of the Incident Voltage and Current and the Reflected Voltage and Current in the Time-Domain

From Figure 6.22, applying Kirchhoff's voltage law (KVL) with subtracting $V\big|_{x+\Delta x}$ from $V\big|_{x}$ and dividing them by Δx:

$$\frac{V(x+\Delta x,\omega,\breve{T})-V(x,\omega,\breve{T})}{\Delta x}=z(x,\omega,\breve{T})I(x,\omega,\breve{T}) \tag{O.1}$$

As $\Delta x \to 0$:

$$\frac{\partial V(x,\omega,\breve{T})}{\partial x}=z(x,\omega,\breve{T})I(x,\omega,\breve{T}) \tag{O.2}$$

Also, from Kirchhoff's current law (KCL):

$$\frac{\partial I(x,\omega,\breve{T})}{\partial x}=y(x,\omega,\breve{T})V(x,\omega,\breve{T}) \tag{O.3}$$

Differentiating (O.2) with respect to x and substituting from (O.3):

$$\begin{aligned}\frac{\partial^2 V(x,\omega,\breve{T})}{\partial x^2} &= \frac{\partial z(x,\omega,\breve{T})}{\partial x}I(x,\omega,\breve{T})+z(x,\omega,\breve{T})\frac{\partial I(x,\omega,\breve{T})}{\partial x}\\ \frac{\partial^2 V(x,\omega,\breve{T})}{\partial x^2} &= \frac{\partial z(x,\omega,\breve{T})}{\partial x}I(x,\omega,\breve{T})+z(x,\omega,\breve{T})y(x,\omega,\breve{T})V(x,\omega,\breve{T})\end{aligned} \tag{O.4}$$

From (6.65), the term $\partial z(x,\omega,\breve{T})/\partial x$ can be simplified as:

$$\begin{aligned}\frac{\partial z(x,\omega,\breve{T})}{\partial x} &= \lim_{\Delta x\to 0}\frac{z(x+\Delta x,\omega,\breve{T})-z(x,\omega,\breve{T})}{\Delta x} && \text{(O.5)}\\ &\approx \lim_{k\to 1}\Big\{\big[r_{k+1}(\omega,T_{k+1})+jx_{l,k+1}(\omega,T_{k+1})\big]\\ &\quad -\big[r_k(\omega,T_k)+jx_{l,k}(\omega,T_k)\big]\Big\} && \text{(O.6)}\\ &\approx 0 \text{ (for simplification}^{5}\text{)}\end{aligned}$$

Also, considering (6.83)-(6.86), (O.4) can be simplified to:

$$\frac{\partial^2 V(x,\omega,\breve{T})}{\partial x^2}-\gamma^2(\omega,\breve{T})V(x,\omega,\breve{T})=0 \tag{O.7}$$

The solution to the above PDE equation is:

$$V(x,\omega,\breve{T})\approx A_1(\omega,\breve{T})e^{\gamma(\omega,\breve{T})x}+A_2(\omega,\breve{T})e^{-\gamma(\omega,\breve{T})x} \tag{O.8}$$

Differentiating (O.8) with respect to x and substituting it in (O.2) after considering (K.5):

$$I(x,\omega,\breve{T})\approx\frac{1}{Z_c(\omega,\breve{T})}\Big[A_1(\omega,\breve{T})e^{\gamma(\omega,\breve{T})x}-A_2(\omega,\breve{T})e^{-\gamma(\omega,\breve{T})x}\Big] \tag{O.9}$$

At the receiving-end, the parameters A_1 and A_2 can be determined from (J.71) as:

$$A_1(\omega,\breve{T}) = \frac{1}{2}\Big[V_R(\omega,\breve{T}) + Z_c(\omega,\breve{T})I_R(\omega,\breve{T})\Big] \tag{O.10}$$

$$A_2(\omega,\breve{T}) = \frac{1}{2}\Big[V_R(\omega,\breve{T}) - Z_c(\omega,\breve{T})I_R(\omega,\breve{T})\Big] \tag{O.11}$$

Based on (6.85), $\gamma(\omega,\breve{T})$ is a complex parameter:

$$\gamma(\omega,\breve{T}) = \bar{\alpha}(\omega,\breve{T}) + j\bar{\beta}(\omega,\breve{T}) \tag{O.12}$$

where the real part is called the *attenuation parameter* and the imaginary part is called the *phase parameter*.

Substituting (O.12) in (O.8) and (O.9) yields:

$$V(x,\omega,\breve{T}) = A_1(\omega,\breve{T})e^{\bar{\alpha}(\omega,\breve{T})x}e^{j\bar{\beta}(\omega,\breve{T})x} + A_2(\omega,\breve{T})e^{-\bar{\alpha}(\omega,\breve{T})x}e^{-j\bar{\beta}(\omega,\breve{T})x} \tag{O.13}$$

$$I(x,\omega,\breve{T}) = B_1(\omega,\breve{T})e^{\bar{\alpha}(\omega,\breve{T})x}e^{j\bar{\beta}(\omega,\breve{T})x} - B_2(\omega,\breve{T})e^{-\bar{\alpha}(\omega,\breve{T})x}e^{-j\bar{\beta}(\omega,\breve{T})x} \tag{O.14}$$

where $B_1(\omega,\breve{T})$ and $B_2(\omega,\breve{T})$ are equal to $A_1(\omega,\breve{T})/Z_c(\omega,\breve{T})$ and $A_2(\omega,\breve{T})/Z_c(\omega,\breve{T})$, respectively.

Now, by extracting the instantaneous v and i using either the imaginary part $\Im$ or the real part $\Re$, (O.13) and (O.14) can be transformed from their phasor-domain to the time-domain as follows:

$$\begin{aligned} v(x,t,\omega,\breve{T}) &= \sqrt{2}\,\Re\Big\{A_1(\omega,\breve{T})e^{\bar{\alpha}(\omega,\breve{T})x}e^{j\left[\omega t+\bar{\beta}(\omega,\breve{T})x\right]}\Big\} \\ &\quad + \sqrt{2}\,\Re\Big\{A_2(\omega,\breve{T})e^{-\bar{\alpha}(\omega,\breve{T})x}e^{j\left[\omega t-\bar{\beta}(\omega,\breve{T})x\right]}\Big\} \end{aligned} \tag{O.15}$$

$$\begin{aligned} i(x,t,\omega,\breve{T}) &= \sqrt{2}\,\Re\Big\{B_1(\omega,\breve{T})e^{\bar{\alpha}(\omega,\breve{T})x}e^{j\left[\omega t+\bar{\beta}(\omega,\breve{T})x\right]}\Big\} \\ &\quad - \sqrt{2}\,\Re\Big\{B_2(\omega,\breve{T})e^{-\bar{\alpha}(\omega,\breve{T})x}e^{j\left[\omega t-\bar{\beta}(\omega,\breve{T})x\right]}\Big\} \end{aligned} \tag{O.16}$$

From the exponential functions, it is obvious that as x increases (i.e., away from the receiving-end), the first terms of both (O.15) and (O.16) increase and the second terms decrease. Thus, (O.15) and (O.16) can be generalized as:

$$v(x,t,\omega,\breve{T}) = v_1(x,t,\omega,\breve{T}) + v_2(x,t,\omega,\breve{T}) \tag{O.17}$$

$$i(x,t,\omega,\breve{T}) = i_1(x,t,\omega,\breve{T}) + i_2(x,t,\omega,\breve{T}) \tag{O.18}$$

where v_1 and v_2 are respectively called the *incident voltage* and the *reflected voltage*, while i_1 and i_2 are respectively called the *incident current* and the *reflected current*.

To find the expressions of these variables, (O.15) and (O.16) should be expanded by considering the following complex parameters:

$$A_1(\omega, \breve{T}) = \psi_1(\omega, \breve{T}) + j\varphi_1(\omega, \breve{T}) \quad \text{(O.19)}$$

$$A_2(\omega, \breve{T}) = \psi_2(\omega, \breve{T}) + j\varphi_2(\omega, \breve{T}) \quad \text{(O.20)}$$

$$B_1(\omega, \breve{T}) = \psi_3(\omega, \breve{T}) + j\varphi_3(\omega, \breve{T}) \quad \text{(O.21)}$$

$$B_2(\omega, \breve{T}) = \psi_4(\omega, \breve{T}) + j\varphi_4(\omega, \breve{T}) \quad \text{(O.22)}$$

To get (6.96), the first real term of (O.15) is solved after being expanded by substituting (O.19). Similar thing for (6.97)-(6.99).

P. Deriving the M-Model of Transmission Lines with Sag

To derive the M-model of the circuit shown in Figure 6.32, first apply KCL at node B:

$$I_{CB} = I_{B0} + I_R \tag{P.1}$$

Applying Ohm's law to that point again:

$$I_{B0} = Y_S V_R \tag{P.2}$$

Substituting (P.2) in (P.1) for I_{B0} yields:

$$I_{CB} = Y_S V_R + I_R \tag{P.3}$$

Applying KCL at node C:

$$I_{AC} = I_{C0} + I_{CB} \tag{P.4}$$

Applying Ohm's law to that point again:

$$I_{C0} = Y_C V_C \tag{P.5}$$

Applying KVL in loop CB:

$$V_C = Z_R I_{CB} + V_R \tag{P.6}$$

Substituting (P.3) in (P.6) for I_{CB} yields:

$$\begin{aligned} V_C &= Z_R \left(Y_S V_R + I_R\right) + V_R \\ \therefore\ V_C &= \left(1 + Z_R Y_S\right) V_R + Z_R I_R \end{aligned} \tag{P.7}$$

Substituting (P.7) in (P.5) for V_C yields:

$$I_{C0} = \left(Y_C + Z_R Y_C Y_S\right) V_R + Z_R Y_C I_R \tag{P.8}$$

Substituting (P.8) and (P.3) in (P.4) for I_{C0} and I_{CB}, respectively, yields:

$$\begin{aligned} I_{AC} &= \left(Y_C + Z_R Y_C Y_S\right) V_R + Z_R Y_C I_R + Y_S V_R + I_R \\ \therefore\ I_{AC} &= \left(Y_C + Y_S + Z_R Y_C Y_S\right) V_R + \left(1 + Z_R Y_C\right) I_R \end{aligned} \tag{P.9}$$

Applying KCL at node A:

$$I_S = I_{A0} + I_{AC} \tag{P.10}$$

Applying Ohm's law to node A:

$$I_{A0} = Y_S V_S \tag{P.11}$$

Substituting (P.11) and (P.9) in (P.10) for I_{A0} and I_{AC}, respectively, yields:

$$I_S = Y_S V_S + (Y_C + Y_S + Z_R Y_C Y_S)\, V_R + (1 + Z_R Y_C)\, I_R \tag{P.12}$$

Applying KVL in loop AC:

$$V_S = Z_L I_{AC} + V_C \tag{P.13}$$

Substituting (P.9) and (P.7) in (P.13) for I_{AC} and V_C, respectively, yields:

$$\begin{aligned} V_S &= (Z_L Y_C + Z_L Y_S + Z_L Z_R Y_C Y_S)\, V_R + (Z_L + Z_L Z_R Y_C)\, I_R + (1 + Z_R Y_S)\, V_R + Z_R I_R \\ &= (1 + Z_L Y_C + Z_R Y_S + Z_L Y_S + Z_L Z_R Y_C Y_S)\, V_R + (Z_L + Z_R + Z_L Z_R Y_C)\, I_R \end{aligned}$$

Therefore,

$$\boxed{\begin{aligned} V_S = {} & [1 + Z_L Y_C + Y_S\,(Z_L + Z_R + Z_L Z_R Y_C)]\, V_R \\ & + \ (Z_L + Z_R + Z_L Z_R Y_C)\, I_R \end{aligned}} \tag{P.14}$$

Substituting (P.14) in (P.12) for V_S yields:

$$\begin{aligned} I_S &= Y_S\,(1 + Z_L Y_C + Z_R Y_S + Z_L Y_S + Z_L Z_R Y_C Y_S)\, V_R + Y_S\,(Z_L + Z_R + Z_L Z_R Y_C)\, I_R \\ &\quad + \ (Y_C + Y_S + Z_R Y_C Y_S)\, V_R + (1 + Z_R Y_C)\, I_R \\ &= \left(Y_S + Z_L Y_C Y_S + Z_R Y_S^2 + Z_L Y_S^2 + Z_L Z_R Y_C Y_S^2 + Y_C + Y_S + Z_R Y_C Y_S\right) V_R \\ &\quad + \ (1 + Z_R Y_C + Z_L Y_S + Z_R Y_S + Z_L Z_R Y_C Y_S)\, I_R \end{aligned}$$

Therefore,

$$\boxed{\begin{aligned} I_S = {} & \left\{Y_C + Y_S\,[2 + Y_C\,(Z_L + Z_R)] + Y_S^2\,(Z_L + Z_R + Z_L Z_R Y_C)\right\} V_R \\ & + \ [1 + Z_R Y_C + Y_S\,(Z_L + Z_R + Z_L Z_R Y_C)]\, I_R \end{aligned}} \tag{P.15}$$

From (P.14) and (P.15), the two-port network of this model can be constructed as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{P.16}$$

where the expressions of the elements $\{A, B, C, D\}$ are:

$$\begin{aligned} A &= 1 + Z_L Y_C + Y_S \left(Z_L + Z_R + Z_L Z_R Y_C\right) \\ B &= Z_L + Z_R + Z_L Z_R Y_C \\ C &= Y_C + Y_S \left[2 + Y_C \left(Z_L + Z_R\right)\right] + Y_S^2 \left(Z_L + Z_R + Z_L Z_R Y_C\right) \\ D &= A \end{aligned}$$

The determinant of this matrix is unity, which can be determined by simplifying $AD - BC$.

Thus, if V_R and I_R are the two unknowns required to be calculated, then the two-port network becomes:

$$\begin{bmatrix} V_R \\ I_R \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} V_S \\ I_S \end{bmatrix}$$

Q. Data of the IEEE 42-Bus Test System (IEEE Std. 399-1997)

The 42-bus test system[6] is shown in Figure 8.5. It is considered as one of the largest ORC test systems available in the literature. This test system is protected by a mixture of 97 directional and non-directional OCRs. That is, using (4.2), the dimension is 194 (i.e., 97 variables of type TMS and 97 variables of type PS) when only one unified TCCC is used for all the relays. If multiple TCCCs are used, then the dimension increases to 485 variables. Also, if the double primary relay strategy (DPRS) is considered, then the dimension increases for both one unified TCCC and multiple TCCCs. For the extreme case (i.e., 194 relays: 97 main-1 relays and 97 main-2 relays), the dimension increases from being 194 to 388 variables if only one unified TCCC is used for all the relays. For multiple TCCCs, the dimension increases from being 485 to 970. Please, refer to Table 8.7.

The same thing for the design constraints where the extreme case of this ORC problem (i.e., applying DPRS to all 194 relays of this test system) has the following constraints:

- 114 inequality constraints for (8.18)
- 194 inequality constraints for (8.21)
- 194 inequality constraints for (8.22)
- 194 side constraints for (8.23)
- 194 side constraints for (8.25)

The standard ANSI CT-rations are listed in Table Q.1. The overload currents and CT-ratios of all the primary relay sets are tabulated in Table Q.2. The three-phase (3ϕ) short-circuit analysis is carried out using the DIgSILENT PowerFactory software. The ANSI standard calculation is applied here with zero fault impedance. The short-circuit currents seen by all the primary/backup (P/B) relay sets are tabulated in Table Q.3. More details about this test system are given in [21, 25, 40, 178].

[6] This test system is known as the IEEE Std. 399-1997 system [178].

Table Q.1: Standard ANSI C37.110-1996 CT-Ratios

No.	**Primary**	**Secondary**	**Ratio**
1	50	5	10
2	75	5	15
3	100	5	20
4	150	5	30
5	200	5	40
6	250	5	50
7	300	5	60
8	400	5	80
9	500	5	100
10	600	5	120
11	800	5	160
12	1000	5	200
13	1200	5	240
14	1500	5	300
15	2000	5	400
16	3000	5	600
17	4000	5	800

Table Q.2: Overload Currents and CT-Ratios Used for the IEEE 42-Bus Test System

Primary Relay	Over-Load Current (kA)	CTR	Primary Relay	Over-Load Current (kA)	CTR
1	0.753	800	50	0.23533299	250
2	0.7844	800	51	0.23533299	250
3	0.7844	800	52	0.094133196	100
4	1.152	1200	53	0.541265877	600
5	1.152	1200	54	1.353164693	1500
6	0.062755464	75	55	1.353164693	1500
7	0.3765	400	56	0.108253175	150
8	0.3765	400	57	0.062755464	75
9	0.3765	400	58	0.359109092	400
10	0.3765	400	59	1.804219591	2000
11	0.188266392	200	60	0.062755464	75
12	0.941331961	1000	61	0.15688866	200
13	0.941331961	1000	62	1.804219591	2000
14	0.188266392	200	63	0.902109796	1000
15	1.152	1200	64	0.047066598	50
16	1.152	1200	65	1.353164693	1500
17	0.4725	500	66	1.804219591	2000
18	0.615	800	67	0.615	800
19	0.4725	500	68	0.615	800
20	0.4725	500	69	0.4725	500
21	0.4725	500	70	0.4725	500
22	0.4725	500	71	0.360843918	400
23	0.4725	500	72	0.085700431	100
24	0.4725	500	73	2.255274489	3000
25	0.4725	500	74	0.346097893	400
26	1.969824442	2000	75	0.017140086	50
27	0.4725	500	76	0.571074723	600
28	0.4725	500	77	0.087472806	100
29	0.4725	500	78	0.180421959	200
30	0.285272165	300	79	0.087472806	100
31	0.4725	500	80	0.180421959	200
32	1.5	1500	81	0.857004306	1000
33	0.4725	500	82	0.405949408	500
34	0.4725	500	83	0.649519053	800
35	0.09021098	100	84	0.599903014	600
36	2.706329387	3000	85	0.087480474	100
37	0.094133196	100	86	0.085700431	100
38	0.094133196	100	87	0.085700431	100
39	2.706329387	3000	88	0.180421959	200
40	0.094133196	100	89	0.072168784	75
41	0.094133196	100	90	0.086452177	100
42	0.4725	500	91	0.198464155	200
43	0.08747885	100	92	0.225527449	250
44	0.094133196	100	93	0.83896211	1000
45	2.706329387	3000	94	0.086452177	100
46	2.706329387	3000	95	0.08747885	100
47	2.706329387	3000	96	0.09021098	100
48	0.4725	500	97	0.847198765	1000
49	0.4725	500	—	—	—

Table Q.3: Three-Phase Fault Currents of the IEEE 42-Bus Test System

Primary Relay	Current (kA)	Backup Relays	Current (kA)	Primary Relay	Current (kA)	Backup Relays	Current (kA)
1	8.367	—	—	40	11.613	27	11.613
2	4.084	—	—	41	13.081	23	13.081
3	5.835	—	—	42	13.163	20	12.997
4	9.821	3	5.835	43	21.056	62	19.803
4	9.821	15	3.876	44	13.197	20	12.997
5	9.821	3	5.835	45	28.984	44	1.0081
5	9.821	16	3.876	46	28.963	41	1.0074
6	13.581	3	5.835	47	28.686	40	0.9978
7	9.115	1	8.367	48	13.07	17	12.923
7	9.115	10	0.748	49	13.321	17	12.923
8	9.113	1	8.367	50	12.526	21	12.526
8	9.113	9	0.746	51	12.812	22	12.722
9	0.767	13	3.835	52	13.391	22	12.722
10	0.768	12	3.84	53	6.046	52	1.0515
11	6.82	8	6.82	54	13.415	51	2.333
12	7.528	2	4.084	55	13.361	50	2.3237
13	7.495	4	2.873	56	13.281	24	13.142
13	7.495	5	2.873	57	13.451	24	13.142
14	6.822	7	6.822	58	3.659	56	1.103
15	11.004	4	2.873	59	30.761	57	1.0699
15	11.004	14	1.2764	60	12.892	49	12.892
16	11.004	5	2.873	61	12.778	48	12.778
16	11.004	14	1.2764	62	19.803	60	0.6888
17	13.364	2	4.084	63	9.156	61	1.5923
17	13.364	11	1.2762	64	12.748	19	12.748
18	12.014	2	4.084	65	15.046	64	0.5233
18	12.014	11	1.2762	66	29.053	6	1.0105
19	13.818	2	4.084	67	35.125	46	28.963
19	13.818	11	1.2762	68	35.125	46	28.963
20	13.542	2	4.084	69	34.656	46	28.963
20	13.542	11	1.2762	70	33.409	45	28.984
21	13.359	2	4.084	71	31.497	66	29.053
21	13.359	11	1.2762	72	30.235	66	29.053
22	13.118	4	2.873	73	30.761	59	30.761
22	13.118	5	2.873	74	3.659	58	3.659
22	13.118	14	1.2764	75	6.046	53	6.046
23	13.698	4	2.873	76	11.69	18	11.69
23	13.698	5	2.873	77	28.487	39	24.941
23	13.698	14	1.2764	78	29.081	39	24.941
24	13.429	4	2.873	79	32.418	36	28.873
24	13.429	5	2.873	80	33.012	36	28.873
24	13.429	14	1.2764	81	15.033	54	13.415
25	13.511	4	2.873	82	16.828	54	13.415
25	13.511	5	2.873	83	15.749	55	13.361
25	13.511	14	1.2764	84	14.904	55	13.361
26	35.288	59	30.761	85	15.046	65	15.046
27	13.132	25	12.937	86	29.038	34	29.038
28	35.343	36	28.873	87	32.348	28	32.348
29	13.109	25	12.937	88	31.627	47	28.686
30	31.986	32	31.986	89	32.853	47	28.686
31	33.338	47	28.686	90	30.648	31	30.648
32	34.57	47	28.686	91	31.782	69	31.782
33	12.749	42	12.749	92	31.392	46	28.963
34	31.412	39	24.941	93	33.158	46	28.963
35	23.847	62	19.803	94	30.736	70	30.736
36	28.873	37	1.0043	95	32.321	45	28.984
37	12.579	33	12.579	96	32.526	45	28.984
38	12.963	29	12.964	97	9.156	63	9.156
39	24.941	38	0.8675	—	—	—	—

R. Data of the Turbo-Generator with 2oo3

The cleaned dataset used in Chapter 9/Section 9.5, which is published in [36], is tabulated below. The active power is in megawatt (MW) and the gas consumption is in normal cubic meter per hour (Nm^3/h).

Table R.1: The Cleaned Dataset of the Real GT Used with the 2oo3 Voting Logic

No.	MW	C (Nm^3/h)	No.	MW	C (Nm^3/h)	No.	MW	C (Nm^3/h)	No.	MW	C (Nm^3/h)
1	3.1	11113	41	29.6	17871	81	48.7	23164	121	58	25761
2	6.3	11835	42	30	17832	82	49.2	23300	122	58.5	25839
3	6.8	12207	43	31.3	18125	83	49.2	23339	123	59.6	25839
4	11.4	13300	44	31.9	18222	84	50.1	23476	124	60.4	26328
5	12.5	13476	45	32.5	18359	85	49.9	23515	125	59.9	26328
6	12.7	13593	46	32.8	18535	86	49.9	23515	126	59.6	26503
7	13	13730	47	34	18710	87	49.9	23515	127	59.9	26464
8	14.6	13964	48	34	18906	88	49.9	23515	128	60.2	26464
9	14.6	14199	49	35.6	19257	89	49.9	23515	129	60.9	26562
10	16.2	14375	50	35.8	19433	90	49.9	23515	130	61.5	26562
11	16.8	14550	51	36.3	19511	91	49.9	23457	131	61.9	26699
12	17.7	14863	52	36.9	19746	92	49.9	23457	132	62.6	26875
13	18.9	14960	53	37.6	19902	93	49.9	23457	133	64.1	27089
14	19.6	15097	54	38.4	20039	94	49.8	23457	134	63.3	27226
15	19.6	15214	55	38.4	20195	95	49.8	23515	135	63.6	27402
16	19.8	15449	56	38.4	20351	96	49.8	23515	136	64.4	27578
17	19.8	15625	57	39	20507	97	49.8	23515	137	65	27675
18	20.2	15703	58	39	20566	98	49.8	23418	138	66.7	27871
19	20.2	15625	59	38.8	20605	99	50	23418	139	67	27988
20	20	15703	60	39	20468	100	50	23496	140	66.6	28300
21	19.9	15625	61	39	20410	101	50.4	23613	141	67.3	28476
22	20.8	15546	62	38.9	20488	102	50.9	23613	142	67.8	28613
23	22.5	15664	63	38.8	20410	103	51.4	23613	143	68.3	28613
24	22.5	15742	64	38.8	20527	104	51.4	23613	144	69.1	28808
25	23	15976	65	38.9	20449	105	51.4	23789	145	70.1	29257
26	24.3	16171	66	39.1	20332	106	52	23964	146	70.5	29355
27	23.9	16347	67	39.7	20468	107	52.5	23964	147	71.2	29570
28	25.5	16425	68	41.4	20820	108	52.5	24101	148	71.8	29726
29	26.1	16543	69	41.7	20976	109	53.1	24101	149	72.8	29843
30	26.1	16699	70	42.9	21093	110	53.1	24101	150	73.1	30273
31	26.6	16875	71	43.6	21269	111	54.1	24335	151	73.3	30351
32	27.3	17070	72	44.1	21523	112	54.8	24335	152	73.8	30351
33	27.5	17265	73	44.6	21757	113	54.8	24472	153	74.6	30488
34	28	17343	74	45.8	21894	114	54.8	24687	154	74.6	30644
35	28.5	17500	75	45.8	22128	115	54.4	24687	155	74.8	30820
36	29.1	17636	76	47.1	22304	116	55	24765	156	75	30859
37	29.6	17812	77	47.1	22480	117	55.9	24765	157	75	30918
38	29.9	17968	78	48.2	22597	118	56.4	25058	158	75	30742
39	29.8	18085	79	48.2	22793	119	57	25234	159	75.2	30898
40	29.5	18027	80	48.7	23007	120	57.5	25585	160	75.1	30898

S. List of Algorithms

Bibliography

[1] Computer Representation of Overcurrent Relay Characteristics. *IEEE Transactions on Power Delivery*, 4(3):1659–1667, jul 1989.

[2] Fundamentals of Speedtronic Mark V Control System. Technical Report A00023 rev.A 8/16/93, GE Power Systems, Schenectady, NY, 1993.

[3] Liquid Wood: Direct Combustion of Wood in a Gas Turbine to Produce Electricity and Heat. *Refocus*, 4(4):52 – 54, 2003.

[4] M.A Abido. Environmental/Economic Power Dispatch Using Multiobjective Evolutionary Algorithms. *Power Systems, IEEE Transactions on*, 18(4):1529–1537, Nov. 2003.

[5] TH. Achammer and A. Denoth. Snow Dielectric Properties: From DC to Microwave X-Band. *Annals of Glaciology*, 19:92–96, 1994.

[6] E. P. Ahern, P. Deane, T. Persson, B. Ó. Gallachóir, , and J. D. Murphy. A Perspective on the Potential Role of Renewable Gas in a Smart Energy Island System. *Renewable Energy*, 78:648–656, 2015.

[7] A. R. Al-Roomi and M. E. El-Hawary. Estimated Economic Load Dispatch Based on Real Operation Logbook. In *2016 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Oct 2016.

[8] A. R. Al-Roomi and M. E. El-Hawary. Optimal Coordination of Directional Overcurrent Relays Using Hybrid BBO/DE Algorithm and Considering Double Primary Relays Strategy. In *2016 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–7, Oct 2016.

[9] A. R. Al-Roomi and M. E. El-Hawary. A New Realistic Optimization-Free Economic Load Dispatch Method Based on Maps Gathered from Sliced Fuel-Cost Curves. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6, 2017.

[10] A. R. Al-Roomi and M. E. El-Hawary. A New Technique to Locate Faults in Distribution Networks Based on Optimal Coordination of Numerical Directional Overcurrent Relays. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6, April 2017.

[11] A. R. Al-Roomi and M. E. El-Hawary. A Novel Multiple Fuels' Cost Function for Realistic Economic Load Dispatch Needs. In *2017 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Oct 2017.

[12] A. R. Al-Roomi and M. E. El-Hawary. Effective Weather/Frequency-Based Transmission Line Models—Part I: Fundamental Equations. In *2017 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Oct 2017.

[13] A. R. Al-Roomi and M. E. El-Hawary. Effective Weather/Frequency-Based Transmission Line Models—Part II: Prospective Applications. In *2017 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–8, Oct 2017.

[14] A. R. Al-Roomi and M. E. El-Hawary. Optimal Coordination of Directional Overcurrent Relays Using Hybrid BBO-LP Algorithm with the Best Extracted Time-Current Characteristic Curve. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6, April 2017.

[15] A. R. Al-Roomi and M. E. El-Hawary. Can Linear Heat Sensors be a Good and Practical Replacement of Traditional Protective Fuses? In *2018 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Oct 2018.

[16] A. R. Al-Roomi and M. E. El-Hawary. Fuel Cost Modeling for Spinning Reserve Thermal Generating Units. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pages 1–6, May 2018.

[17] A. R. Al-Roomi and M. E. El-Hawary. How to Improve Linear Fuel-Cost Function to Compete with Quadratic and Cubic Functions. In *2018 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, 2018.

[18] A. R. Al-Roomi and M. E. El-Hawary. New Heat Energy Trading Concepts for the Next Generation Smart Grids. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pages 1–6, May 2018.

[19] A. R. Al-Roomi and M. E. El-Hawary. Optimizing Load Forecasting Configurations of Computational Neural Networks. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pages 1–6, May 2018.

[20] A. R. Al-Roomi and M. E. El-Hawary. Possible Approaches to Trade Non-Electric Energy Sources in the Next Generation Smart Grids. In *2018 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Oct 2018.

[21] A. R. Al-Roomi and M. E. El-Hawary. Optimal Coordination of Double Primary Directional Overcurrent Relays Using a New Combinational BBO/DE Algorithm. *IEEE Canadian Journal of Electrical and Computer Engineering*, 42(3):135–147, Summer 2019.

[22] A. R. Al-Roomi and M. E. El-Hawary. Universal Functions Originator—Part I: Design. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–6, 2019.

[23] A. R. Al-Roomi and M. E. El-Hawary. Universal Functions Originator—Part II: Evaluation. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–6, 2019.

[24] Ali R. Al-Roomi. Optimal Relays Settings by Using Biogeography Based Optimization "BBO" Technique. Master's thesis, University of Bahrain, Sakhir, Bahrain, 2014.

[25] Ali R. Al-Roomi. Optimal Relay Coordination Test Systems Repository, 2015. [Accessed Aug. 29, 2019].

[26] Ali R. Al-Roomi. Power Flow Test Systems Repository, 2015. [Accessed Aug. 29, 2019].

[27] Ali R. Al-Roomi. Unconstrained Single-Objective Benchmark Functions Repository, 2015. [Accessed Aug 29, 2019].

[28] Ali R. Al-Roomi. Economic Load Dispatch Test Systems Repository, 2016. [Accessed Aug 29, 2019].

[29] Ali R. Al-Roomi. Telegrapher's Equations: The Hidden Backbone of Electric Power Systems Analysis. Technical report, Dalhousie University, Halifax, Nova Scotia, Canada, 2016.

[30] Ali R. Al-Roomi and Mohamed E. El-Hawary. Economic Load Dispatch Using Hybrid MpBBO-SQP Algorithm. In Xin-She Yang, editor, *Nature-Inspired Computation in Engineering*, volume 637 of *Studies in Computational Intelligence*, pages 217 – 250. Springer International Publishing, Switzerland, 1 edition, 2016.

[31] Ali R. Al-Roomi and Mohamed E. El-Hawary. Metropolis Biogeography-Based Optimization. *Information Sciences*, 360:73 – 95, 2016.

[32] Ali R. Al-Roomi and Mohamed E. El-Hawary. Estimating Complex Power Magnitudes Using a Bank of Pre-Defined PFs Embedded in ANNs. In *2019 IEEE Electrical Power and Energy Conference (EPEC) (EPEC19)*, pages 1–7, Montreal, Canada, October 2019.

[33] Ali R. Al-Roomi and Mohamed E. El-Hawary. Estimating Power Flow Directions Using Off-Line PF Analysis and Artificial Neural Networks. In *2019 IEEE Electrical Power and Energy Conference (EPEC) (EPEC19)*, pages 1–6, Montreal, Canada, October 2019.

[34] Ali R. Al-Roomi and Mohamed E. El-Hawary. Is It Enough to Just Rely on Near-End, Middle, and Far-End Points to Get Feasible Relay Coordination? In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE) (CCECE 2019)*, Edmonton, Canada, May 2019.

[35] Ali R. Al-Roomi and Mohamed E. El-Hawary. Novel Highly Precise Power Loss Estimators That Directly Solve Power Balance Equality Constraints. In *2019 IEEE Electrical Power and Energy Conference (EPEC) (EPEC19)*, pages 1–8, Montreal, Canada, October 2019.

[36] Ali R. Al-Roomi and Mohamed E. El-Hawary. Diagnosing Fuel Pumps, Power Transducers, CTs, and PTs via Fuel-Power Function and 2oo3 Voting. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (IEEE CCECE 2020)*, pages 1–6, London, Canada, April 2020.

[37] Ali R. Al-Roomi and Mohamed E. El-Hawary. Hybridizing UFO with Other ML Tools to Locate Faults by Just Knowing Relay Operating Times. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (IEEE CCECE 2020)*, pages 1–6, London, Canada, April 2020.

[38] Ali R. Al-Roomi and Mohamed E. El-Hawary. M-Model: A New Precise Medium-Length Transmission Line Model. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (IEEE CCECE 2020)*, pages 1–6, London, Canada, April 2020.

[39] Ali R. Al-Roomi and Mohamed E. El-Hawary. Mathematical Schemes to Linearize Operating Times of Overcurrent Relays by Sequentially Fixing Plug Settings and Time Multiplier Settings. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (IEEE CCECE 2020)*, pages 1–6, London, Canada, April 2020.

[40] F. A. Albasri, A. R. Alroomi, and J. H. Talaq. Optimal Coordination of Directional Overcurrent Relays Using Biogeography-Based Optimization Algorithms. *IEEE Transactions on Power Delivery*, 30(4):1810–1820, Aug 2015.

[41] A. A. S. Algarni and K. Bhattacharya. Disco Operation Considering DG Units and Their Goodness Factors. *IEEE Transactions on Power Systems*, 24(4):1831–1840, Nov 2009.

[42] M S Almas, R Leelaruji, and L Vanfretti. Over-Current Relay Model Implementation for Real Time Simulation amp; Hardware-in-the-Loop (HIL) validation. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pages 4789–4796, oct 2012.

[43] Ali R. Alroomi, Fadhel A. Albasri, and Jawad H. Talaq. A Comprehensive Comparison of the Original Forms of Biogeography-Based Optimization Algorithms. *International Journal on Soft Computing, Artificial Intelligence and Applications*, 2(5/6):11–30, Dec. 2013.

[44] Ali R. Alroomi, Fadhel A. Albasri, and Jawad H. Talaq. Solving the Associated Weakness of Biogeography-Based Optimization Algorithm. *Inter. Journal on Soft Computing*, 4(4):1–20, Nov. 2013.

[45] Y A Alsariera, H S Alamri, A M Nasser, M A Majid, and K Z Zamli. Comparative Performance Analysis of Bat Algorithm and Bacterial Foraging Optimization Algorithm Using Standard Benchmark Functions. In *Software Engineering Conference (MySEC), 2014 8th Malaysian*, pages 295–300, sep 2014.

[46] ALSTOM. *Network Protection & Automation Guide*. ALSTOM, Levallois-Perret, France, 2002.

[47] J.S. Alsumait, J.K. Sykulski, and A.K. Al-Othman. A Hybrid GA-PS-SQP Method to Solve Power System Valve-Point Economic Dispatch Problems. *Applied Energy*, 87(5):1773 – 1781, 2010.

[48] M. M. Aman, Ghauth Bin Jasmon, Qadeer Ahmed Khan, A. Halim Bin Abu Bakar, and Jasrul Jamani Jamian. Modeling and Simulation of Reverse Power Relay for Generator Protection. In *2012 IEEE Inter. Power Engineering and Optimization Conference Melaka, Malaysia*, pages 317–322, June 2012.

[49] Akihiro Ametani, Naoto Nagaoka, Yoshihiro Baba, and Teruo Ohno. *Power System Transients: Theory and Applications*. CRC Press, Boca Raton, FL, 2013.

[50] M. Amin, M. Molinas, J. Lyu, and X. Cai. Impact of Power Flow Direction on the Stability of VSC-HVDC Seen From the Impedance Nyquist Plot. *IEEE Transactions on Power Electronics*, 32(10):8204–8217, Oct 2017.

[51] T. Amraee. Coordination of Directional Overcurrent Relays Using Seeker Algorithm. *Power Delivery, IEEE Transactions on*, 27(3):1415–1422, July 2012.

[52] Michael A. Anthony. *Electric Power System Protection and Coordination: A Design Handbook for Overcurrent Protection*. McGraw-Hill Inc., New York, 1995.

[53] Michael Anumaka. Influence Corona and Skin Effect on the Nigerian 330kV Interconnected and Power System. *International Journal of Research and Reviews in Applied Sciences (IJRRAS)*, V12:320–327, Aug. 2012.

[54] Arbil. Why is this TensorFlow Implementation Vastly Less Successful than Matlab's NN?, Nov. 2015. [Accessed Dec. 10, 2018].

[55] AREVA. *MiCOM P120/P121/P122/P123 Overcurrent Relays*. Schneider Electric, Headquarter: Palo Alto, California, United States, version 12 edition, 2010.

[56] AREVA. Directional / non Directional Overcurrent Protection, 2011.

[57] J. Arrillaga and C. P. Arnold. *Computer Analysis of Power Systems*. John Wiley & Sons Inc., Chichester, 1990.

[58] J. Arrillaga and N. R. Watson. *Computer Modelling of Electrical Power Systems*. John Wiley & Sons Inc., Chichester, 2 edition, 2001.

[59] Arthur R. Bergen. *Power Systems Analysis*. Electrical and Computer Engineering. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1986.

[60] D. Ashlock and W. Ashlock. A Note on Population Size Inspired by the Extinction of Mammoths. In *2017 IEEE Conf. on Comp. Intel. in Bioinformatics and Comp. Biology (CIBCB)*, pages 1–8, Aug 2017.

[61] M. Bahrami and J. Rashed-Mohassel. An Exact Solution of Coherent Wave Propagation in Rain Medium with Realistic Raindrop Shapes. *Progress In Electromagnetics Research*, 79:107–118, 2008.

[62] Saeed Bahrami, Ali Ghaffari, Magnus Genrup, and Marcus Thern. Performance Comparison Between Steam Injected Gas Turbine and Combined Cycle During Frequency Drops. *Energies*, 8(8):7582–7592, Jul 2015.

[63] A. Bakirtzis, V. Petridis, and S. Kazarlis. Genetic Algorithm Solution to the Economic Dispatch Problem. *Generation, Transmission and Distribution, IEE Proceedings-*, 141(4):377–382, Jul. 1994.

[64] Uday A. Bakshi and Mayuresh V. Bakshi. *Transmission and Distribution of Electrical Power*. Technical Publications, Pune, India, 2014.

[65] M N Bandyopadhyay. *Electrical Power Systems: Theory and Practice*. Prentice-Hall of India Pvt. Ltd., New Delhi, India, 2006.

[66] Patrick Barry and Tony Phillips. Mysteries of Rain and Snow, Mar. 2007. [Accessed Sept. 25, 2019].

[67] Thomas Bartz-Beielstein. *Experimental Research in Evolutionary Computation: The New Experimentalism*. Natural Computing Series. Springer, Berlin, Germany, 2006.

[68] D. R. Baughman and Y. A. Liu. *Neural Networks in Bioprocessing and Chemical Engineering*. Academic Press, San Diego, CA, 1995.

[69] H. Wayne Beaty and Donald G. Fink. *Standard Handbook for Electrical Engineers Sixteenth Edition*. McGraw-Hill Education, New York, 16th ed. edition, 2012.

[70] P.P. Bedekar and S.R. Bhide. Optimum Coordination of Directional Overcurrent Relays Using the Hybrid GA-NLP Approach. *Power Delivery, IEEE Transactions on*, 26(1):109–119, Jan. 2011.

[71] Roberto Benato, Sebastian Dambone Sessa, and Fabio Guglielmi. Determination of Steady-State and Faulty Regimes of Overhead Lines by Means of Multiconductor Cell Analysis (MCA). *Energies*, 5(8):2771–2793, July 2012.

[72] M. M. Bhaskar, S. Muthyala, and S. Maheswarapu. A Music Based Harmony Search (MBHS) Approach to Optimal Power Flow with Reactive Power Loss Optimization. In *2010 IEEE International Conference on Power and Energy*, pages 7–11, Nov 2010.

[73] D. Birla, R.P. Maheshwari, and H.O. Gupta. A New Nonlinear Directional Overcurrent Relay Coordination Technique, and Banes and Boons of Near-End Faults Based Approach. *Power Delivery, IEEE Transactions on*, 21(3):1176–1182, July 2006.

[74] J. Lewis Blackburn and Thomas J. Domin. *Protective Relaying: Principles and Applications*. Power Engineering. CRC Press, Boca Raton, FL, 3rd edition, 2006.

[75] Paul Boggs and J.W. Tolle. Sequential Quadratic Programming for Large-Scale Nonlinear Optimization. *Journal of Computational and Applied Mathematics*, 124, 12 2000.

[76] M. J. Booysen and A. H. Cloete. Sustainability through Intelligent Scheduling of Electric Water Heaters in a Smart Grid. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 848–855, Aug 2016.

[77] Marija Bočkarjova and Göran Andersson. Transmission Line Conductor Temperature Impact on State Estimation Accuracy. In *2007 IEEE Lausanne Power Tech*, pages 701–706, July 2007.

[78] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, June 1998.

[79] B Can, A Beham, and C Heavey. A Comparative Study of Genetic Algorithm Components in Simulation-Based Optimisation. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 1829–1837, dec 2008.

[80] W. Bernard Carlson. *Tesla: Inventor of the Electrical Age*. Princeton University Press, 2013.

[81] Julie Kerr Casper. *Natural Resources - Energy: Powering the Past, Present, and Future*. Chelsea House Publishers, New York, 2007.

[82] V. Cecchi, A. S. Leger, K. Miu, and C. O. Nwankpa. Incorporating Temperature Variations Into Transmission-Line Models. *IEEE Transactions on Power Delivery*, 26(4):2189–2196, Oct 2011.

[83] CERTS. Reliability Performance Monitoring (RPM) Using Phasor and SCADA Measurements: Prototype Functional Specification for MISO. Technical report, CERTS, Advanced Systems Researchers, University of Illinois, USA, July 2012.

[84] S Chan and R Maurer. Modeling Overcurrent Relay Characteristics. *IEEE Computer Applications in Power*, 5(1):41–45, 1992.

[85] K. Chandram, N. Subrahmanyam, and M. Sydulu. Root Finding Techniques for Economic Dispatch Problems. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–8, July 2008.

[86] H. K. Chappa, T. Thakur, and S. C. Srivastava. Reactive Power Loss Based Voltage Instability Detection Using Synchrophasor Technology. In *2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 1–5, Nov 2015.

[87] K.T. Chaturvedi, M. Pandit, and L. Srivastava. Self-Organizing Hierarchical Particle Swarm Optimization for Nonconvex Economic Dispatch. *Power Systems, IEEE Transactions on*, 23(3):1079–1087, Aug. 2008.

[88] P. Che, L. Tang, and J. Wang. Two-Stage Minimax Stochastic Unit Commitment. *IET Generation, Transmission & Distribution*, 12(4):947–956, 2018.

[89] C.H. Chen and S.N. Yeh. Particle Swarm Optimization for Economic Power Dispatch with Valve-Point Effects. In *Transmission Distribution Conference and Exposition: Latin America, 2006. TDC '06. IEEE/PES*, pages 1–5, Aug. 2006.

[90] Debao Chen, Feng Zou, Jiangtao Wang, and Wujie Yuan. A Teaching–Learning-Based Optimization Algorithm with Producer–Scrounger Model for Global Optimization. *Soft Computing*, 19(3):745–762, 2015.

[91] Hao Chen and NicholasS. Flann. Parallel Simulated Annealing and Genetic Algorithms: A Space of Hybrid Methods. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature — PPSN III*, volume 866 of *Lecture Notes in Computer Science*, pages 428–438. Springer Berlin Heidelberg, 1994.

[92] K Chen, C Huang, and J He. Fault Detection, Classification and Location for Transmission Lines and Distribution Systems: A Review on the Methods. *High Voltage*, 1(1):25–33, 2016.

[93] S. X. Chen, Y. S. Foo. Eddy, H. B. Gooi, M. Q. Wang, and S. F. Lu. A Centralized Reactive Power Compensation System for LV Distribution Networks. *IEEE Transactions on Power Systems*, 30(1):274–284, Jan 2015.

[94] Chao-Lung Chiang. Improved Genetic Algorithm for Power Economic Dispatch of Units With Valve-Point Effects and Multiple Fuels. *Power Systems, IEEE Transactions on*, 20(4):1690–1699, Nov. 2005.

[95] Edwin K. P. Chong and Stanislaw H. Żak. *An Introduction to Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2nd edition, 2001.

[96] P. Chopade and M. Bikdash. Minimizing Cost and Power Loss by Optimal Placement of Capacitor Using ETAP. In *2011 IEEE 43rd Southeastern Symposium on System Theory*, pages 24–29, March 2011.

[97] A. Ciani, A. Eroglu, F. Güthe, and B. Paikert. Full-Scale Atmospheric Tests of Sequential Combustion. In *Proceedings of ASME Turbo Expo 2010: Power for Land, Sea and Air (GT2010)*, volume 2, pages 759–768, Glasgow, UK, Jun. 2010.

[98] Martin L. Cody. *Plants on Islands: Diversity and Dynamics on a Continental Archipelago*. University of California Press, Berkeley, Calif, June 2006.

[99] A. J. Conejo. Load Flow, jul 2011.

[100] M. H. Costa, R. R. Saldanha, M. G. Ravetti, and E. G. Carrano. Robust Coordination of Directional Overcurrent Relays Using a Matheuristic Algorithm. *IET Gen., Trans. & Dist.*, 11(2):464–474, 2017.

[101] C. Barry Cox and P. D. Moore. *Biogeography: An Ecological and Evolutionary Approach*. Blackwell Scientific Publications, Oxford England Cambridge, Mass., USA, 1993.

[102] James W. Dally, William F. Riley, and Kenneth G. McConnell. *Instrumentation for Engineering Measurements*. Wiley, New York, 2 edition, 1993.

[103] M.J. Damborg, R. Ramaswami, S.S. Venkata, and J. M. Postforoosh. Computer Aided Transmission Protection System Design Part I: Algorithms. *Power Apparatus and Systems, IEEE Transactions on*, PAS-103(1):51–59, Jan. 1984.

[104] George Bernard Dantzig. Programming in a Linear Structure. Technical Report Report of the September 9, 1948 meeting in Madison, Econometrica, 1948.

[105] J. C. Das. *ARC Flash Hazard Analysis and Mitigation*. IEEE Press Series on Power Engineering. Wiley-IEEE Press, Hoboken, New Jersey, 2012.

[106] Livio de Santoli, Romano Paiolo, and Gianluigi Lo Basso. An Overview on Safety Issues Related to Hydrogen and Methane Blend Applications in Domestic and Industrial Use. *Energy Procedia*, 126:297 – 304, 2017. ATI 2017 - 72nd Conference of the Italian Thermal Machines Engineering Association.

[107] Kalyanmoy Deb. *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall of India, 11th edition, 2010.

[108] Debapriya Das. *Electrical Power Systems*. New Age International (P) Ltd., New Delhi, 2006.

[109] Atif S. Debs. *Modern Power Systems Control and Operation*. Kluwer Academic Publishers, Boston, 1988.

[110] K. Deep and J.C. Bansal. Optimization of Directional Overcurrent Relay Times Using Laplace Crossover Particle Swarm Optimization (LXPSO). In *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 288–293, Dec 2009.

[111] Kusum Deep, Dinesh Birla, R. P. Maheshwari, H. O. Gupta, and Manoj Thakur. A Population Based Heuristic Algorithm for Optimal Relay Operating Times. *World Journal of Modeling and Simulation*, 2(3):167–176, March 2006.

[112] Michael Deighton. *Facility Integrity Management: Effective Principles and Practices for the Oil, Gas and Petrochemical Industries*. Gulf Professional Publishing, an imprint of Elsevier, Cambridge, MA, 2016.

[113] M T N Dinh, M Bahadornejad, A S A Shahri, and N K C Nair. Protection Schemes and Fault Location Methods for Multi-Terminal Lines: A Comprehensive Review. In *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pages 1–6, nov 2013.

[114] J. F. Dopazo, O. A. Klitin, G. W. Stagg, and M. Watson. An Optimization Technique for Real and Reactive Power Allocation. *Proceedings of the IEEE*, 55(11):1877–1885, Nov 1967.

[115] Leandro dos Santos Coelho, Rodrigo Clemente Thom Souza, and Viviana Cocco Mariani. Improved Differential Evolution Approach Based on Cultural Algorithm and Diversity Measure Applied to Solve Economic Load Dispatch Problems. *Mathematics and Computers in Simulation*, 79(10):3136 – 3147, 2009.

[116] Dawei Du, D Simon, and M Ergezer. Biogeography-Based Optimization Combined with Evolutionary Strategy and Immigration Refusal. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 997–1002, 2009.

[117] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. [Accessed Jun. 17, 2019].

[118] J. S. Dudor and L. K. Padden. Protective Relaying on Medium and High Voltage Systems, Some Lessons to be Learned. In *Petroleum and Chemical Industry Conference, 1994. Record of Conference Papers., IEEE Incorporated Industry Applications Society 41st Annual*, pages 53–61, Sep 1994.

[119] L. Dufour, D. Genoud, B. Ladevie, J. J. Bezian, F. Cimmino, and S. Genoud. Economic Interest of Heating and Hot Water Prediction System for Residential District. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 827–832, March 2016.

[120] Bengt Edlén. The Refractive Index of Air. *Metrologia*, 2(2):71–80, Apr 1966.

[121] A. E. Eiben, Elena Marchiori, and V. A. Valkó. Evolutionary Algorithms with On-the-Fly Population Size Adjustment. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, pages 41–50, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[122] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Natural computing series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[123] Mohamed E. El-Hawary. *Electrical Power Systems: Design and Analysis*. Prentice-Hall Inc., Reston, Virginia, 1983.

[124] Mohamed E. El-Hawary. *Electrical Power Systems: Design and Analysis*. Wiley-IEEE Press, New York, 1 edition, 1995.

[125] Mohamed E. El-Hawary. *Introduction to Electrical Power Systems*. John Wiley & Sons Inc., Hoboken, New Jersey, 2008.

[126] Walter A. Elmore. *Protective Relaying: Theory And Applications*. ABB - Marcel Dekker Inc., New York, 1994.

[127] H. B. Elrafie and M. R. Irving. Linear Programming for Directional Overcurrent Relay Coordination in Interconnected Power Systems with Constraint Relaxation. *Electric Power Systems Research*, 27(3):209 – 216, Aug. 1993.

[128] Energy Information Administration (EIA). United States Natural Gas Industrial Price, Nov. 2017. [Accessed Dec. 25, 2017].

[129] Masoud Esmaili, Esmail Chaktan Firozjaee, and Heidar Ali Shayanfar. Optimal Placement of Distributed Generations Considering Voltage Stability and Power Losses with Observing Voltage-Related Constraints. *Applied Energy*, 113:1252 – 1260, 2014.

[130] A Farag, S. Al-Baiyat, and T. C. Cheng. Economic Load Dispatch Multiobjective Optimization Procedures Using Linear Programming Techniques. *Power Systems, IEEE Transactions on*, 10(2):731–738, May 1995.

[131] L. A. F. M. Ferreira. Tellegen's Theorem and Power Systems-New Load Flow Equations, New Solution Methods. *IEEE Transactions on Circuits and Systems*, 37(4):519–526, apr 1990.

[132] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188, September 1936.

[133] R Fletcher. *Practical Methods of Optimization*. Wiley-interscience publication. John Wiley & Sons, Inc., Chichester, New York, 2nd edition, 1987.

[134] Dario Floreano and Claudio Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, Cambridge, Massachusetts, 8 2008.

[135] Gianluigi Folino, Clara Pizzuti, and Giandomenico Spezzano. Genetic Programming and Simulated Annealing: A Hybrid Method to Evolve Decision Trees. In Riccardo Poli, Wolfgang Banzhaf, WilliamB. Langdon, Julian Miller, Peter Nordin, and TerenceC. Fogarty, editors, *Genetic Programming*, volume 1802 of *Lecture Notes in Computer Science*, pages 294–303. Springer Berlin Heidelberg, 2000.

[136] Wilhelm Forst and Dieter Hoffmann. *Optimization– Theory and Practice*. Springer, New York, 2010.

[137] Michelle Michot Foss. Introduction to LNG: An Overview on Liquefied Natural Gas (LNG), Its Properties, Organization of the LNG Industry and Safety Considerations, Jan. 2007. [Accessed Apr. 8, 2016].

[138] S. Frank, J. Sexauer, and S. Mohagheghi. Temperature-Dependent Power Flow. *IEEE Transactions on Power Systems*, 28(4):4007–4018, Nov 2013.

[139] Ernest Freeberg. *The Age of Edison: Electric Light and the Invention of Modern America*. Penguin Publishing Group, New York, 2013.

[140] Y. Fukuyama and Y. Ueki. An Application of Artificial Neural Network to Dynamic Economic Load Dispatching. In *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems*, pages 261–265, Seattle, WA, Jul. 1991. IEEE.

[141] Zwe-Lee Gaing. Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints. *Power Systems, IEEE Transactions on*, 18(3):1187–1195, Aug. 2003.

[142] D. Gan, R. J. Thomas, and R. D. Zimmerman. Stability-Constrained Optimal Power Flow. *IEEE Transactions on Power Systems*, 15(2):535–540, May 2000.

[143] Christie J. Geankoplis. *Transport Processes and Unit Operations*. PTR Prentice Hall, Englewood Cliffs, N.J, 3rd edition, 1993.

[144] L. Gelazanskas and K. A. A. Gamage. Managing Renewable Intermittency in Smart Grid: Use of Residential Hot Water Heaters as a Form of Energy Storage. In *2016 IEEE 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pages 1–4, May 2016.

[145] Stuart Geman and D Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, nov 1984.

[146] Michel Gendreau and Jean-Yves Potvin. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer, 2010.

[147] Juan M. Gers and Edward J. Holmes. *Protection of Electricity Distribution Networks*. IEE Power & Energy Series 47. Institution Of Engineering And Technology, London, UK, 2nd edition, 2004.

[148] Iman Ghalehkhondabi, Ehsan Ardjmand, Gary R. Weckman, and William A. Young. An Overview of Energy Demand Forecasting Methods Published in 2005–2015. *Energy Systems*, 8(2):411–447, Apr 2016.

[149] J. Duncan Glover, Mulutukutla S.Sarma, and Thomas J. Overbye. *Power System Analysis and Design*. Cengage Learning, Stamford, CT, 5 edition, 2012.

[150] J. C. Gómez, D. H. Tourn, S. Nesci, L. Sanchez, and H. Rovere. Why the Operation Failure of High Breaking Capacity Fuses is so Frequent? *CIRED - Open Access Proceedings Journal*, 2017(1):1545–1549, Oct. 2017.

[151] Wenyin Gong, Zhihua Cai, Charles X. Ling, and Hui Li. A Real-Coded Biogeography-Based Optimization with Mutation. *Applied Mathematics and Computation*, 216(9):2749 – 2758, Jul. 2010.

[152] Wenyin Gong, Zhihua Cai, and CharlesX. Ling. DE/BBO: A Hybrid Differential Evolution with Biogeography-Based Optimization for Global Numerical Optimization. *Soft Computing*, 15(4):645–665, 2010.

[153] Ben Goodman. Sequential Quadratic Programming, Jun 2015. [Accessed Aug. 1, 2019].

[154] W. Mack Grady. Transmission Lines, Feb 2012. [Accessed Sept. 21, 2019].

[155] John J. Grainger and William D. Stevenson. *Power System Analysis*. McGraw-Hill Education, New York, 1994.

[156] Leonard Grigsby. *Electric Power Generation, Transmission, and Distribution*. CRC Press, Boca Raton, FL, 2012.

[157] K. Gulzar, S. Sierla, V. Vyatkin, N. Papakonstantinou, P. G. Flikkema, and C. W. Yang. An Auction-Based Smart District Heating Grid. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–8, Sept 2015.

[158] Convenor H. Schmitt, J Amon, D.Braun, F X Camescasse, M Collet, G C Damstra, H Fukagawa, F Gil Garcia, K.-H. Hartung, J Kida, and M Saravolac. Fault Current Limiters. Technical Report CIGRE WG 13.10, IEEE PES Switchgear Committee, 2003.

[159] Atidel Ben Hadj-Alouane and James C. Bean. A Genetic Algorithm for the Multiple-Choice Integer Program. *Operations Research*, 45(1):92–101, 1997.

[160] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural Network Design*. 2nd ed. edition, 2014.

[161] Alen Hatibovic. Derivation of Equations for Conductor and Sag Curves of an Overhead Line Based on a Given Catenary Constant. *Periodica Polytechnica Electrical Engineering*, 58(1):23, 2014.

[162] B. Hayes, J. Gruber, and M. Prodanovic. Short-Term Load Forecasting at the Local Level Using Smart Meter Data. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, June 2015.

[163] William H. Hayt. *Engineering Electromagnetics*. McGraw-Hill Education, Boston, 6th ed. edition, 2001.

[164] Shawn; Tulpule Punit J. Hegde, Bharatkumar; Midlam-Mohler. Thermal Model of Fuse Dynamics for Simulation Under Intermittent DC Faults. In *2015 ASME Dynamic Systems and Control Conference (DSCC)*, volume 2, Columbus, Ohio, USA, Oct. 2015.

[165] S. Hemamalini and Sishaj P. Simon. Artificial Bee Colony Algorithm for Economic Load Dispatch Problem with Non-smooth Cost Functions. *Electric Power Components and Systems*, 38(7):786–803, 2010.

[166] J. Hetzer, D.C. Yu, and K. Bhattarai. An Economic Dispatch Model Incorporating Wind Power. *Energy Conversion, IEEE Transactions on*, 23(2):603–611, Jun. 2008.

[167] Leslie Hewitson, Mark Brown, and Ramesh Balakrishnan. *Practical Power System Protection*. Practical Professional Books. Newnes, Oxford Burlington, MA, 2004.

[168] G. T. Heydt. Identification of Harmonic Sources by a State Estimation Technique. *IEEE Transactions on Power Delivery*, 4(1):569–576, Jan 1989.

[169] William W. Hogan. Markets in Real Electric Networks Require Reactive Prices. *The Energy Journal*, 14(3):171–200, 1993.

[170] Holland H. Farr. *Transmission Line: Design Manual*. A Water Resources Technical Publication, Denver, Colorado, 1980.

[171] T. Hong, A. Raza, and F. de León. Optimal Power Dispatch Under Load Uncertainty Using a Stochastic Approximation Method. *IEEE Trans. on Power Systems*, 31(6):4495–4503, Nov 2016.

[172] Tao Hong. Very Short, Short, Medium and Long Term Load Forecasting [Energy Forecasting], Oct. 2012. [Accessed Aug. 12, 2019].

[173] Wei-Chiang Hong. *Intelligent Energy Demand Forecasting*, volume 10 of *Lecture Notes in Energy*. Springer-Verlag, London New York, 2013.

[174] Bruce J. Hunt. Oliver Heaviside: A First-Rate Oddity. *Physics Today*, 65(11):48, Nov. 2012. [Accessed March 09, 2016].

[175] Ashfaq Husain. *Electrical Power Systems*. CBS Publishers & Distributors, New Delhi, India, 5th edition, 2007.

[176] Nathan Ida. *Sensors, Actuators, and their Interfaces: A Multidisciplinary Introduction*. SciTech Publishing Inc, Edison, NJ, 2014.

[177] IEC. Single Input Energizing Quality Measuring Relays With Dependent or Independent. *IEC Publication 255-3*, 1989.

[178] IEEE. *IEEE Std 399-1997, IEEE Recommended Practice for Industrial and Commercial Power Systems Analysis (The IEEE Brown Book)*. Institute of Electrical & Electronics Engineers, New York, 1998.

[179] IEEE. IEEE Standard Inverse-Time Characteristic Equations for Overcurrent Relays. *IEEE Standard C37.112-1996*, 14(3):868–872, 1999.

[180] IEEE. IEEE Standard Test Code for Dry-Type Distribution and Power Transformers. *C57.12.91-2011*, 2001.

[181] Boris Igelnik and Dan. Simon. The Eigenvalues of a Tridiagonal Matrix in Biogeography. *Applied Mathematics and Computation*, 218(1):195 – 201, Sept. 2011.

[182] W. T. Illingworth. Beginner's Guide to Neural Networks. In *Proceedings of the IEEE National Aerospace and Electronics Conference*, pages 1138–1144 vol.3, May 1989.

[183] L. Imen, L. Djamel, S. Hassiba, D. Abdellah, and F. Selwa. Optimal Power Flow Study Using Conventional and Neural Networks Methods. In *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 1422–1427, Nov 2015.

[184] J. C. Das. *Power System Analysis: Short-Circuit Load Flow and Harmonics*. Power Engineering. Marcel Dekker Inc., New York, 2002.

[185] James A. Momoh. *Electric Power System Applications of Optimization*. Power Engineering. Marcel Dekker Inc., New York, 2001.

[186] James L. Kirtley. *Electric Power Principles: Sources, Conversion, Distribution and Use*. John Wiley & Sons Inc., 2010.

[187] Momin Jamil and Xin She Yang. A Literature Survey of Benchmark Functions for Global Optimisation Problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

[188] V. Jamison Janawitz, James Masso, and Christopher Childs. Heavy-Duty Gas Turbine Operating and Maintenance Considerations. Technical Report GER-3620M (02/15), GE Power & Water, GE Power & Water, Atlanta, GA, Feb. 2015.

[189] Stanko Janković and Bojan Ivanović. Application of Combined Newton-Raphson Method to Large Load Flow Models. *Electric Power Systems Research*, 127:134–140, 2015.

[190] Peter Jansohn. *Modern Gas Turbine Systems: High Efficiency, Low Emission, Fuel Flexible Power Generation*. Energy. Woodhead Publishing, Cambridge, 2013.

[191] X. Jiang, J. Huang, B. Dong, L. Zhou, and C. Yuan. Influence of Fog Water Conductivity on AC Corona Characteristics of Transmission Line. *Gaodianya Jishu/High Voltage Engineering*, 39:636–641, 03 2013.

[192] Jizhong Zhu. *Optimization of Power System Operation*. Wiley-IEEE Press, New Jersey, 2009.

[193] A. T. Johns and S. K. Salman. *Digital Protection for Power Systems*. IEE power series 15. P. Peregrinus on behalf of the Institution of Electrical Engineers, London, UK, 1995.

[194] Richard A. Johnson. *Miller and Freund's Probability and Statistics for Engineers*. Prentice Hall Inc., Upper Saddle River, NJ, 6th edition, 2000.

[195] R.L. Jones. *Biogeography: Structure, Process, Pattern, and Change Within the Biosphere*. Hulton Educational Publications Limited, 1980.

[196] Franz Joos, Philipp Brunner, Marcel Stalder, and Stefan Tschirren. Field Experience of the Sequential Combustion System for the GT24/GT26 Gas Turbine Family. In *ASME 1998 International Gas Turbine and Aeroengine Congress and Exhibition*, volume 4: Heat Transfer; Electric Power; Industrial and Cogeneration, pages 1–10, Stockholm, Sweden, June 1998.

[197] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, and Dusan Fister. A Brief Review of Nature-Inspired Algorithms for Optimization. *Elektrotehniški Vestnik*, 80(3):1–7, 2013.

[198] Francisco Erivaldo Fernandes Junior and Gary G. Yen. Particle Swarm Optimization of Deep Neural Networks Architectures for Image Classification. *Swarm and Evolutionary Computation*, 49:62 – 74, 2019.

[199] Zaakirah Kajee-Bagdadi. Differential Evolution Algorithms for Constrained Global Optimization. Master's thesis, University of the Witwatersrand, Johannesburg, South Africa, 2007. [Accessed Sept. 22, 2019].

[200] Nagaraja Kamsali, B.S.N. Prasad, and Jayati Datt. The Electrical Conductivity as an Index of Air Pollution in the Atmosphere. In *Advanced Air Pollution*. InTech, Aug. 2011.

[201] Hossein Kazemi Karegar, Hossein Askarian Abyaneh, Vivian Ohis, and Matin Meshkin. Pre-Processing of the Optimal Coordination of Overcurrent Relays. *Electric Power Systems Research*, 75(2-3):134–141, Aug. 2005.

[202] V. Karthikeyan, S. Senthilkumar, and V.J. Vijayalakshmi. A New Approach to the Solution of Economic Dispatch Using Particle Swarm Optimization with Simulated Annealing. *Inter. Journal on Comput. Sciences & Applications (IJCSA)*, 3(3):37–49, Jun. 2013.

[203] Y A Katsigiannis, P S Georgilakis, and E S Karapidakis. Hybrid Simulated Annealing-Tabu Search Method for Optimal Sizing of Autonomous Power Systems With Renewables. *Sustainable Energy, IEEE Transactions on*, 3(3):330–338, jul 2012.

[204] Debora M. Katz. *Physics for Scientists and Engineers: Foundations and Connections with Modern*. Cengage Learning, Boston, MA, 2016.

[205] Dogan Kaya. A New Approach to the Telegraph Equation: An Application of the Decomposition Method. *Bulletin of the Institute of Mathematics, Academia Sinica*, 28:51–57, 01 2000.

[206] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[207] D. A. Knoll and D. E. Keyes. Jacobian-Free Newton-Krylov Methods: a Survey of Approaches and Applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[208] K. Y. Kok, P. Rajendran, R. Rainis, and W. M. M. Wan Ibrahim. Investigation on Selection Schemes and Population Sizes for Genetic Algorithm in Unmanned Aerial Vehicle Path Planning. In *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, pages 6–10, Aug 2015.

[209] D P Kothari and J S Dhillon. *Power System Optimization*. PHI Learning Private Limited, New Delhi, India, 2nd edition, 2011.

[210] D.P. Kothari and J.S. Dhillon. *Power System Optimization*. PHI Learning, New Delhi, 2nd revised edition edition, 2010.

[211] D.P. Kothari and I.J. Nagrath. *Modern Power System Analysis*. Tata McGraw-Hill Publishing Company, New Delhi, India, 2003.

[212] N. Kumar, R. Wangneo, P. K. Kalra, and S. C. Srivastava. Application of Artificial Neural Networks to Load Flow Solutions. In *TENCON '91. Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems*, volume 1, pages 199–203, Aug 1991.

[213] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill Education, New York, 1st edition, 1 1994.

[214] Michael H Kutner, Christopher J. Nachtsheim, John Neter, and William Li. *Applied Linear Statistical Models*. McGraw-Hill/Irwin Series Operations and Decision Sciences. McGraw-Hill/Irwin, 5 edition, 8 2004.

[215] D. Larcher and J.-M. Tarascon. Towards Greener and more Sustainable Batteries for Electrical Energy Storage. *Nature Chemistry*, 7(1):19–29, Jan. 2015.

[216] M. A. Lasemi, M. Assili, and M. Baghayipour. Modification of Multi-Area Economic Dispatch with Multiple Fuel Options, Considering the Fuelling Limitations. *IET Generation, Transmission Distribution*, 8(6):1098–1106, June 2014.

[217] F.N. Lee and A.M. Breipohl. Reserve Constrained Economic Dispatch with Prohibited Operating Zones. *IEEE Transactions on Power Systems*, 8(1):246–254, Feb. 1993.

[218] Kwang Y. Lee and Mohamed A. El-Sharkawi, editors. *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems (IEEE Press Series on Power Engineering)*. IEEE Press Series on Power Engineering. IEEE Press Wiley-Interscience, Piscataway, N.J. Hoboken, N.J, 2008.

[219] Aaron St. Leger and Chika Nwankpa. OTA-Based Transmission Line Model with Variable Parameters for Analog Power Flow Computation. *International Journal of Circuit Theory and Applications*, 38(2):199–220, 2010.

[220] Cornelius T. Leondes. *Optimization Techniques, Volume 2 (Neural Network Systems Techniques and Applications) (Pt. 2)*. Academic Press, San Diego, 1998.

[221] S. Li, M. Ding, and S. Du. Transmission Loadability With Field Current Control Under Voltage Depression. *IEEE Transactions on Power Delivery*, 24(4):2142–2149, Oct 2009.

[222] Xie Li, Li Xingcai, and Zheng Xiaojing. Attenuation of an Electromagnetic Wave by Charged Dust Particles in a Sandstorm. *Applied Optics*, 49(35):6756, December 2010.

[223] Chen-Ching Liu, Stephen McArthur, and Seung-Jae Lee, editors. *Smart Grid Handbook, 3 Volume Set*. John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, 2016.

[224] K. L. Lo and Y. A. Al-Turki. A New Method for Real and Reactive Power Loss Allocation in Bilateral Markets. In *2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies. Proceedings*, volume 1, pages 280–284 Vol.1, April 2004.

[225] M.R. Lohokare, S.S. Pattnaik, B.K. Panigrahi, and Sanjoy Das. Accelerated Biogeography-Based Optimization with Neighborhood Search for Optimization. *Applied Soft Computing*, 13(5):2318 – 2342, May 2013.

[226] Mark V. Lomolino, Brett R. Riddle, and James H. Brown. *Biogeography*. Sinauer Associates, Incorporated, Sunderland, Mass, 3 rd. edition, 2009.

[227] Jonathan B. Losos and Robert E. Ricklefs. *The Theory of Island Biogeography Revisited*. Princeton University Press, Princeton, 2010.

[228] Sean Luke. *Essentials of Metaheuristics*. Lulu, Available for free at http://cs.gmu.edu/~sean/book/metaheuristics/, 2009.

[229] M Lundy and A Mees. Convergence of an Annealing Algorithm. *Math. Program.*, 34(1):111–124, jan 1986.

[230] Sergio Lupi, Michele Forzan, and Aleksandr Aliferov. *Induction and Direct Resistance Heating: Theory and Numerical Modeling*. Springer International Publishing, Cham Switzerland, 2015.

[231] Lynn Powell. *Power System Load Flow Analysis*. McGraw-Hill Inc., New York, 2005.

[232] J Singh M. S. Sachdev and R J Fleming. Mathematical Models Representing Time-Current Characteristics of Overcurrent Relays for Computer Applications. In *Proc. of IEEE PES - Winter Meeting*, pages 1–8, 1978.

[233] Haiping Ma. An Analysis of the Equilibrium of Migration Models for Biogeography-Based Optimization. *Information Sciences*, 180(18):3444–3464, 2010.

[234] Haiping Ma, Suhong Ni, and Man Sun. Equilibrium Species Counts and Migration Model Tradeoffs for Biogeography-Based Optimization. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3306–3310, 2009.

[235] Haiping Ma and Dan Simon. Analysis of Migration Models of Biogeography-Based Optimization Using Markov Theory. *Engineering Applications of Artificial Intelligence*, 24(6):1052 – 1060, 2011.

[236] Haiping Ma and Dan. Simon. Blended Biogeography-Based Optimization for Constrained Optimization. *Engineering Applications of Artif. Intel.*, 24(March):517 – 525, 2011.

[237] Robert H. MacArthur. *Geographical Ecology: Patterns in the Distribution of Species*. Harper & Row Publishers Inc., New York, 1972.

[238] Robert H. MacArthur and Joseph H. Connell. *The Biology of Populations*. John Wiley & Sons Inc., New York, 1966.

[239] Robert H. MacArthur and Edward O. Wilson. An Equilibrium Theory of Insular Zoogeography. *Evolution*, 17(4):373–387, Dec. 1963.

[240] Robert H. MacArthur and Edward O. Wilson. *The Theory of Island Biogeography*. Landmarks in Biology Series. Princeton University Press, 1967.

[241] Glen M. MacDonald. *Biogeography: Introduction to Space, Time, and Life*. John Wiley & Sons Inc., New York, Oct. 2003.

[242] S. Madan and K. E. Bollinger. Neural Network Based Power Flow Predictor. In *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, volume 3, pages 1331–1334 vol.3, May 1999.

[243] Amita Mahor, Vishnu Prasad, and Saroj Rangnekar. Economic Dispatch Using Particle Swarm Optimization: A Review. *Renew. and Sust. Energy Reviews*, 13(8):2134 – 2141, 2009.

[244] Franco Maloberti and Anthony C. Davies, editors. *A Short History of Circuits and Systems*. River Publishers, Aalborg, 2016.

[245] Mohamed M. Mansour, S.F. Mekhamer, and N.E.-S. El-Kharbawe. A Modified Particle Swarm Optimizer for the Coordination of Directional Overcurrent Relays. *Power Delivery, IEEE Transactions on*, 22(3):1400–1410, July 2007.

[246] Kimbal Marriott and Peter Stuckey. *Programming with Constraints: An Introduction*. The MIT Press, 1998.

[247] Martin Sewell. History of Support Vector Machines, Jan. 2008. [Accessed Feb. 21, 2019].

[248] C. Russell Mason. *The Art and Science of Protective Relaying*. General Electric series. Wiley, Chapman & Hall, New York, 1956.

[249] Mathworks. Improve Neural Network Generalization and Avoid Overfitting (R2017a). Official Website, 2017. [Accessed Jul. 9, 2017].

[250] Mathworks. Multilayer Neural Network Architecture (R2017a). Brochure, 2017. [Accessed Jun. 17, 2017].

[251] Patrick J. McCleer. *The Theory and Practice of Overcurrent Protection*. Mechanical Products Inc., Jackson, Mich, 1987.

[252] Warren S. McCulloch and Walter Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[253] Constantin Meis. *Light And Vacuum: The Wave-particle Nature Of The Light And The Quantum Vacuum. Electromagnetic Theory And Quantum Electrodynamics Beyond The Standard Model*. World Scientific, New Jersey, 2nd ed. edition, 2017.

[254] A. P. Sakis Meliopoulos. Power System Modeling, Analysis and Control. Technical Report Report of the September 9, 1948 meeting in Madison, Atlanta, GA: Georgia Institute of Technology, 2006.

[255] A. P. Sakis Meliopoulos, Sun Wook Kang, G. J. Cokkinides, and R. Dougal. Animation and Visualization of Spot Prices via Quadratized Power Flow Analysis. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 7 pp.–, jan 2003.

[256] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[257] Zbigniew Michalewicz. A survey of Constraint Handling Techniques in Evolutionary Computation Methods. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 135–155. MIT Press, 1995.

[258] Zakaryia Mohammed and J Talaq. Economic Dispatch by Biogeography Based Optimization Method. In *2011 International Conference on Signal, Image Processing and Applications*, volume 21, pages 161–165, Singapore, 2011. IACSIT Press.

[259] S. T. Mohyud-Din, M. A. Noor, and K. I. Noor. Variational Iteration Method for Solving Telegraph Equations. *Applications and Applied Mathematics: An International Journal (AAM)*, 4(1):114–121, 2009.

[260] James A. Momoh. *Electric Power System Applications of Optimization (Power Engineering (Willis))*. Marcel Dekker, Inc., New York, 2001.

[261] M Monadi, C Koch-Ciobotaru, A Luna, J Ignacio Candela, and P Rodriguez. Multi-Terminal Medium Voltage DC Grids Fault Location and Isolation. *IET Generation, Transmission Distribution*, 10(14):3517–3528, 2016.

[262] Zahra Moravej, Mostafa Jazaeri, and Mehdi Gholamzadeh. Optimal Coordination of Distance and Over-Current Relays in Series Compensated Systems Based on MAPSO. *Energy Conv. and Manag.*, 56:140–151, Apr. 2012.

[263] Jorge J. Moré. The Levenberg-Marquardt Algorithm: Implementation and Theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[264] Jorge J. Moré and D. C. Sorensen. Computing a Trust Region Step. *SIAM J. Sci. Stat. Comput.*, 4(3):553–572, September 1983.

[265] H. H. Muller, M. J. Rider, C. A. Castro, and V. Leonardo Paucar. Power Flow Model Based on Artificial Neural Networks. In *2005 IEEE Russia Power Tech*, pages 1–6, June 2005.

[266] Alan A. Myers and Paul S. Giller, editors. *Analytical Biogeography: An Integrated Approach to the Study of Animal and Plant Distributions*. Chapman and Hall, London, UK, 1 edition, Sept. 1990.

[267] Pratap Mysore. Power System Protection Lectures - Lecture 1 (Fundamentals of Protection), Aug. 2010. [Accessed Apr. 13, 2016].

[268] Paul J. Nahin. *Oliver Heaviside: The Life, Work, and Times of an Electrical Genius of the Victorian Age*. Johns Hopkins University Press, Baltimore, Md, 2002.

[269] Ryohei Nakano, Yuval Davidor, and Takeshi Yamada. Optimal Population Size Under Constant Computation Cost. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature — PPSN III*, pages 130–138, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[270] J. Nanda, L.L. Lai, J.T. Ma, N. Rajkumar, A. Nanda, and M. Prasad. A Novel Approach to Computationally Efficient Algorithms for Transmission Loss and Line Flow Formulations. *International Journal of Electrical Power & Energy Systems*, 21(8):555 – 560, 1999.

[271] National Research Council Canada (NRCC). Gas Turbine Research Facility Virtual Tour, 2016. [Accessed Jan. 05, 2018].

[272] NEOS. Bound Constrained Optimization, Jun 2012. [Accessed Aug. 1, 2019].

[273] NEOS. Sequential Quadratic Programming, Jun 2012. [Accessed Aug. 1, 2019].

[274] T. T. Nguyen. Neural Network Optimal-Power-Flow. In *1997 Fourth International Conference on Advances in Power System Control, Operation and Management, APSCOM-97. (Conf. Publ. No. 450)*, volume 1, pages 266–271 vol.1, Nov 1997.

[275] M. Nick, O. Alizadeh-Mousavi, R. Cherkaoui, and M. Paolone. Security Constrained Unit Commitment With Dynamic Thermal Line Rating. *IEEE Trans. on Power Syst.*, 31(3):2014–2025, May 2016.

[276] William A. Nierenberg, editor. *Encyclopedia of Environmental Biology, Three-Volume Set: 1-3*. Academic Press, 1 edition, June 1995.

[277] Taher Niknam. A New Fuzzy Adaptive Hybrid Particle Swarm Optimization Algorithm for Non-Linear, Non-Smooth and Non-Convex Economic Dispatch Problem. *Applied Energy*, 87(1):327 – 339, 2010.

[278] Taher Niknam, Hasan Doagou Mojarrad, and Hamed Zeinoddini Meymand. A Novel Hybrid Particle Swarm Optimization for Economic Dispatch with Valve-Point Loading Effects. *Energy Conversion and Management*, 52(4):1800 – 1809, 2011.

[279] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Operations Research and Financial Engineering. Springer, New York, 2nd edition, 7 2006.

[280] A.S. Noghabi, J. Sadeh, and H.R. Mashhadi. Considering Different Network Topologies in Optimal Overcurrent Relay Coordination Using a Hybrid GA. *Power Delivery, IEEE Transactions on*, 24(4):1857–1863, Oct. 2009.

[281] Jifu Nong. Radial Basis Function Neural Networks Optimization Algorithm Based on SVM. In Maotai Zhao and Junpin Sha, editors, *Communications and Information Processing*, pages 291–298, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[282] Ferris Waldo Norris and Lloyd Arthur Bingham. *Introductory Study of Electrical Characteristics of Power and Telephone Transmission Lines*. International Textbook Co., Scranton, PA, 1936.

[283] H. Noto. Some Experiments upon the Charge and the Conductivity of Cloud. *Japanese Journal of Astronomy and Geophysics*, 6(3):143–160, Jan. 1929. [Provided by the NASA Astrophysics Data System, copyrighted by the National Research Council of Japan].

[284] Yaghout Nourani and Bjarne Andresen. A Comparison of Simulated Annealing Cooling Strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373–8385, 1998.

[285] Nova Scotia Power. NSDOE Request]. Technical report, Emera Company, Halifax, Nova Scotia, Canada, Apr. 2013. [Accessed Aug. 14, 2019].

[286] Nova Scotia Power. OASIS: Open Access Same-Time Information System, May. 2019. [Accessed Aug. 19, 2019].

[287] D. Dike Obioma and A. M. Izuchukwu. Comparative Analysis of Techniques for Economic Dispatch of Generated Power with Modified Lambda-Iteration Method. In *2013 IEEE International Conference on Emerging Sustainable Technologies for Power ICT in a Developing Society (NIGERCON)*, pages 231–237, Nov 2013.

[288] M. O. Odetayo. Optimal Population Size for Genetic Algorithms: An Investigation. In *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pages 2/1–2/4, May 1993.

[289] Robert G. Olsen. *High Voltage Overhead Transmission Line Electromagnetics: Volume I.* CreateSpace Independent Publishing Platform, San Bernardino, California Lexington, Kentucky, 2015.

[290] Tarik Ozkul. *Data Acquisition and Process Control Using Personal Computers.* CRC Press, New York, 1996.

[291] K. R. Padiyar. *Power System Dynamics: Stability and Control.* BSP, BS Publications, Hyderabad, India, 2nd edition, 2008.

[292] Y.G. Paithankar and S.R. Bhide. *Fundamentals of Power System Protection.* Prentice-Hall of India Pvt. Ltd., New Delhi, 2003.

[293] P. Pals, R. M. Dunnett, C. J. Aldridge, and A. Prangley. Calculation of Transmission Loss Factors in the Presence of Constraints. In *DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No.00EX382)*, pages 326–331, 2000.

[294] S. N. Pandey, S. Verma, and L. Srivastava. ANN Based Approach for Nodal Congestion Pricing. In *2006 IEEE International Conference on Industrial Technology*, pages 13–18, Dec 2006.

[295] S. Panta, S. Premrudeepreechachani, S. Nuchprayoon, C. Dechthummarong, S. Janjornmanit, and S. Yachiangkam. Optimal Economic Dispatch for Power Generation Using Artificial Neural Network. In *International Power Engineering Conference (IPEC 2007)*, pages 1343–1348. IEEE, Dec. 2007.

[296] V. A. Papaspiliotopoulos, G. N. Korres, and N. G. Maratos. A Novel Quadratically Constrained Quadratic Programming Method for Optimal Coordination of Directional Overcurrent Relays. *IEEE Transactions on Power Delivery*, 32(1):3–10, Feb 2017.

[297] V. A. Papaspiliotopoulos, T. A. Kurashvili, and G. N. Korres. Optimal Coordination of Directional Overcurrent Relays in Distribution Systems with Distributed Generation Based on a Hybrid PSO-LP Algorithm. In *MedPower 2014*, pages 1–6, Nov 2014.

[298] J.H. Park, Y.S. Kim, IK. Eom, and K.Y. Lee. Economic Load Dispatch for Piecewise Quadratic Cost Function Using Hopfield Neural Network. *Power Systems, IEEE Transactions on*, 8(3):1030–1038, Aug. 1993.

[299] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and K.Y. Lee. A Particle Swarm Optimization for Economic Dispatch With Nonsmooth Cost Functions. *Power Systems, IEEE Transactions on*, 20(1):34–42, Feb. 2005.

[300] S.S. Pattnaik, M.R. Lohokare, and S. Devi. Enhanced Biogeography-Based Optimization Using Modified Clear Duplicate Operator. In *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pages 715–720, Dec. 2010.

[301] Paul M. Anderson. *Power System Protection*. IEEE Press Series on Power Engineering. Wiley-IEEE Press, New York, 1998.

[302] Roointon Pavri and Gerald D. Moore. Gas Turbine Emissions and Control. Technical Report GER-4211, GE Energy Services, Atlanta, GA, Mar. 2001.

[303] PayScale. The Average Wage: United States, Dec. 2017. [Accessed Dec. 25, 2017].

[304] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[305] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. sklearn.svm.SVR, Nov. 2011. [Accessed Aug. 18, 2019].

[306] PEX Universe. PEX Tubing Technical Specifications and General Installation Recommendations, Jul. 2017. [Accessed Jan. 09, 2018].

[307] A. G. Phadke, S. H. Horowitz, and J. S. Thorp. Integrated Hierarchical Computer Systems for Adaptive Protective Relaying and Control of Electric Transmission Power Systems, Nov. 1989. [Accessed Apr. 13, 2016].

[308] C R S Pierre and T E Wolny. Standardization of Benchmarks for Protective Device Time-Current Curves. *IEEE Transactions on Industry Applications*, IA-22(4):623–633, 1986.

[309] M. Pilz and L. Al-Fagih. Recent Advances in Local Energy Trading in the Smart Grid Based on Game — Theoretic Approaches. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2017.

[310] P.S.R. Murthy. *Power System Analysis*. BS Pub., Hyderabad, 2014.

[311] F. L. Quilumba, W. Lee, H. Huang, D. Y. Wang, and R. L. Szabados. Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities. *IEEE Transactions on Smart Grid*, 6(2):911–918, March 2015.

[312] I. Jacob Raglend and Narayana Prasad Padhy. Comparison of Practical Unit Commitment Problem Solutions. *Electric Power Components and Systems*, 36(8):844–863, 2008.

[313] N.A. Rahmat and I. Musirin. Differential Evolution Ant Colony Optimization (DEACO) technique in solving Economic Load Dispatch problem. In *2012 IEEE Inter. on Power Engineering and Optimization Conference (PEDCO)*, pages 263–268, Melaka, Malaysia, Jun. 2012.

[314] V. N. Rajput, F. Adelnia, and K. S. Pandya. Optimal Coordination of Directional Overcurrent Relays Using Improved Mathematical Formulation. *IET Generation, Transmission Distribution*, 12(9):2086–2094, 2018.

[315] Badri Ram and D. N. Vishwakarma. *Power System Protection and Switchgear*. Tata McGraw-Hill, New Delhi, India, 1995.

[316] Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Hoboken, New Jersey, 4 edition, 2009.

[317] G. I. Rashed, Y. Sun, K. A. Rashed, and H. I. Shaheen. Optimal Location of Unified Power Flow Controller by Differential Evolution Algorithm Considering Transmission Loss Reduction. In *2012 IEEE International Conference on Power System Technology (POWERCON)*, pages 1–6, Oct 2012.

[318] Malcolm Raven. Experimental Measurements of the Skin Effect and Internal Inductance at Low Frequencies. *Acta Technica CSAV (Ceskoslovensk Akademie Ved)*, 60:51–69, 01 2015.

[319] W. Rebizant, J. Szafran, and A. Wiszniewski. *Digital Signal Processing in Power System Protection and Control*. Signals and Communication Technology. Springer, London, UK, 2011.

[320] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez. Control of Power Converters in AC Microgrids. *IEEE Transactions on Power Electronics*, 27(11):4734–4749, Nov 2012.

[321] Raquel Rodríguez-Pérez, Martin Vogt, and Jürgen Bajorath. Support Vector Machine Classification and Regression Prioritize Different Structural Features for Binary Compound Activity and Potency Value Prediction. *ACS Omega*, 2(10):6371–6379, October 2017.

[322] RSD Fire Engineering Pte Ltd. Fire Alarm System, 2018. [Accessed Apr. 03, 2018].

[323] Tomas Tinoco De Rubira. Alternative Methods for Solving Power Flow Problems, 2012.

[324] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall - Pearson, Upper Saddle River, NJ, 3rd ed. edition, 2010.

[325] R. M. Rylatt, J. R. Snape, P. Allen, B. M. Ardestani, P. Boait, E. Boggasch, D. Fan, G. Fletcher, R. Gammon, M. Lemon, et al. Exploring Smart Grid Possibilities: A Complex Systems Modelling Approach. *Smart Grid*, 1(1):1–15, 2015.

[326] Hadi Saadat. *Power System Analysis*. McGraw-Hill series in electrical and computer engineering. WCB/McGraw-Hill, Singapore, 1999.

[327] N Sadati, M Zamani, and H R F Mahdavian. Hybrid Particle Swarm-Based-Simulated Annealing Optimization Techniques. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 644–648, nov 2006.

[328] Safe Fire Detection, Inc. Standard Linear Heat Detection, 2018. [Accessed Apr. 03, 2018].

[329] Murari Mohan Saha, Jan Jozef Izykowski, and Eugeniusz Rosolowski. *Fault Location on Power Networks (Power Systems)*. Springer, London, 2010.

[330] Balwinder Singh Samra. An Analysis of Single Wire Earth Return (SWER) System for Rural Electrification. Master's thesis, Missouri University of Science and Technology, Rolla, Missouri, 1972. [Accessed Sept. 22, 2019].

[331] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, July 1959.

[332] J. R. Santos, A. G. Expósito, and F. P. Sáanchez. Assessment of Conductor Thermal Models for Grid Studies. *IET Generation, Transmission Distribution*, 1(1):155–161, January 2007.

[333] Wolfgang Schellong. Energy Demand Analysis and Forecast. In *Energy Management Systems*. InTech, Aug 2011.

[334] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester New-York, 1998.

[335] Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. Wiley, Hoboken, NJ, 2003.

[336] A. Immanuel Selvakumar and K. Thanushkodi. Optimization Using Civilized Swarm: Solution to Economic Dispatch with Multiple Minima. *Electric Power Systems Research*, 79(1):8 – 16, 2009.

[337] A.I. Selvakumar and K. Thanushkodi. A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems. *Power Systems, IEEE Transactions on*, 22(1):42–51, Feb. 2007.

[338] Horng-Lin Shieh, Cheng-Chien Kuo, and Chin-Ming Chiang. Modified Particle Swarm Optimization Algorithm with Simulated Annealing Behavior and Its Numerical Verification. *Applied Mathematics and Computation*, 218(8):4365–4383, 2011.

[339] Tom Short. Fault Location on Distribution Systems: An Update on EPRI and DOE Research. Technical Report P128.003, Electric Power Research Institute (EPRI), Headquarter: Palo Alto, California, United States, 2007.

[340] R. R. Shoults and M. M. Mead. Optimal Estimation of Piece-Wise Linear Incremental Cost Curves for EDC. *IEEE Transactions on Power Apparatus and Systems*, PAS-103(6):1432–1438, June 1984.

[341] Gerard Sierksma. *Linear and Integer Programming: Theory and Practice*. CRC Press, New York, 2 edition, 2002.

[342] Dan. Simon. Biogeography-Based Optimization. *Evolutionary Computation, IEEE Transactions on*, 12(6):702–713, Dec. 2008.

[343] Dan. Simon. The Matlab Code of Biogeography-Based Optimization, Aug. 2008. [Accessed Feb. 1, 2013].

[344] Dan. Simon. A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm. *Evolutionary Computation*, 19(2):167–188, June 2011.

[345] Dan. Simon. *Evolutionary Optimization Algorithms : Biologically-Inspired and Population-Based Approaches to Computer Intelligence*. John Wiley & Sons Inc, Hoboken, New Jersey, 2013.

[346] Dan. Simon, M. Ergezer, and Dawei Du. Population Distributions in Biogeography-Based Optimization Algorithms with Elitism. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 991–996, Oct. 2009.

[347] Manohar Singh, B.K. Panigrahi, and A.R. Abhyankar. Optimal Coordination of Directional Over-Current Relays Using Teaching Learning-Based Optimization (TLBO) Algorithm. *International Journal of Electrical Power & Energy Systems*, 50:33 – 41, 2013.

[348] N. Sinha, R. Chakrabarti, and P.K. Chattopadhyay. Evolutionary Programming Techniques for Economic Load Dispatch. *Evolutionary Computation, IEEE Transactions on*, 7(1):83–94, Feb. 2003.

[349] S. Sivanagaraju and G. Sreenivasan. *Power System Operation and Control*. Pearson Education, Chennai, 2010.

[350] S. Sivananaithaperumal, S. Miruna Joe Amali, S. Baskar, and P.N. Suganthan. Constrained Self-Adaptive Differential Evolution Based Design of Robust Optimal Fixed Structure Controller. *Engineering Applications of Artificial Intelligence*, 24(6):1084 – 1093, Sept. 2011.

[351] Alex J. Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3):199–222, Aug 2004.

[352] Glenn W. Stagg and Ahmed H. El-Abiad. *Computer Methods in Power System Analysis*. McGraw-Hill Inc., Tokyo, 1968.

[353] T. Steihaug. The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[354] R. H. Stevens. Power Flow Direction Definitions for Metering of Bidirectional Power. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(9):3018–3022, Sept 1983.

[355] R. Storn. On the Usage of Differential Evolution for Function Optimization. In *Fuzzy Info. Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, pages 519–523, Jun. 1996.

[356] Rainer Storn and Kenneth Price. Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704-1198, Suite 600, Fax: 510-643-7684, March 1995.

[357] Syed A. Nasar. *Schaum's Outline of Electrical Power Systems*. McGraw-Hill Inc., New York, 1990.

[358] H Szu and R Hartley. Fast Simulated Annealing. *Physics Letters A*, 122:157–162, jun 1987.

[359] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

[360] Radha Thangaraj, Millie Pant, and Kusum Deep. Optimal Coordination of Over-Current Relays Using Modified Differential Evolution Algorithms. *Engineering Applications of Artificial Intelligence*, 23(5):820–829, Aug. 2010. Advances in metaheuristics for hard optimization: new trends and case studies.

[361] William A Thue. *Electrical Power Cable Engineering*. Power Engineering (Willis). Taylor & Francis, 3rd edition, 2011.

[362] Hamid R. Tizhoosh. Opposition-Based Learning: A New Scheme for Machine Intelligence. In *Proceedings of the International Conference on Computational*

Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'06) - Volume 01, CIMCA '05, pages 695–701, Washington, DC, USA, 2005. IEEE Computer Society.

[363] A Trias. The Holomorphic Embedding Load Flow Method. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8, jul 2012.

[364] Pınar T́ufekci. Prediction of Full Load Electrical Power Output of a Base Load Operated Combined Cycle Power Plant Using Machine Learning Methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140, 2014.

[365] John Twidell. *Renewable energy resources*. Routledge, Taylor & Francis Group, London - New York, 2015.

[366] H. Ungrad. *Protection Techniques in Electrical Energy Systems*. CRC Press, New York, 1 edition, 8 1995.

[367] A.J. Urdaneta, R. Nadira, and L.G. Perez Jimenez. Optimal Coordination of Directional Overcurrent Relays in Interconnected Power Systems. *Power Delivery, IEEE Transactions on*, 3(3):903–911, Jul. 1988.

[368] A.J. Urdaneta, L.G. Perez, and H. Restrepo. Optimal Coordination of Directional Overcurrent Relays Considering Dynamic Changes in the Network Topology. *Power Delivery, IEEE Transactions on*, 12(4):1458–1464, Oct. 1997.

[369] Nguyen Quang Uy, Nguyen Xuan Hoai, Michael O'Neill, R. I. McKay, and Edgar Galván-López. Semantically-Based Crossover in Genetic Programming: Application to Real-Valued Symbolic Regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, Jun 2011.

[370] Ebrahim Vaahedi. *Practical Power System Operation*. IEEE Press Series on Power Engineering. Wiley-IEEE Press, Hoboken, New Jersey, 2014.

[371] K. Vanthournout, R. D'hulst, D. Geysen, and G. Jacobs. A Smart Domestic Hot Water Buffer. *IEEE Transactions on Smart Grid*, 3(4):2121–2127, Dec 2012.

[372] V. N. Vapnik and A. Ya. Lerner. Pattern Recognition Using Generalized Portraits. *Automation and Remote Control*, 24(6):774–780, June 1963.

[373] V. Černý. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

[374] P Venkataraman. *Applied Optimization: with MATLAB Programming*. John Wiley & Sons, Hoboken, N.J, 2 edition, 2009.

[375] Ajit Kumar Verma, Srividya Ajit, and Manoj Kumar. *Dependability of Networked Computer-based Systems (Springer Series in Reliability Engineering)*. Springer, London, 2011.

[376] T.Aruldoss Albert Victoire and A.Ebenezer Jeyakumar. Hybrid PSO-SQP for Economic Dispatch with Valve-Point Effect. *Electric Power Systems Research*, 71(1):51 – 59, 2004.

[377] Bill Waggener. *Pulse Code Modulation Techniques: with Applications in Communications and Data Recording*. Van Nostrand Reinhold, New York, 1995.

[378] Xi-Fan Wang, Yonghua Song, and Malcolm Irving. *Modern Power Systems Analysis*. Springer US, New York, 2008.

[379] A. G. Ward, N. R. Watson, C. P. Arnold, A. J. Turner, and B. J. Ring. Inversion of Real Time Spot Prices in the Direction of Real Power Flow in a Transmission Line. *IEEE Transactions on Power Systems*, 15(4):1197–1203, Nov 2000.

[380] Mathias Wärja, Pontus Slottner, and Markus Bohlin. Customer Adapted Maintenance Plan (CAMP): A Process for Optimization of Gas Turbine Maintenance. In *Proceedings of ASME Turbo Expo 2008: Power for Land, Sea and Air (GT2008): Controls, Diagnostics and Instrumentation; Cycle Innovations; Electric Power*, volume 2, pages 37–45. ASME, June 2008.

[381] weather.gc.ca. Past Weather and Climate - Historical Data. Technical Report Weather, Climate, and Hazard, Government of Canada - Environment and Natural Resources, Nova Scotia: Halifax Dockyard Station, Jan. 2012. [Accessed Dec. 28, 2017].

[382] E. F. Wehlage. Geothermal or Hydrothermal Heat Source Refrigeration Versus Electric Power and Fossil Fuel Sources for Process and Air Conditioning. *IEEE Transactions on Industry Applications*, IA-10(4):449–455, July 1974.

[383] S. Weitemeyer, D. Kleinhans, T. Vogt, and C. Agert. Integration of Renewable Energy Sources in Future Power Systems: The Role of Storage. *Renewable Energy*, 75:14–20, 2015.

[384] K.S.K Weranga, Sisil Kumarawadu, and D. P. Chandima. *Smart Metering Design and Applications*. SpringerBriefs in Applied Sciences and Technology. Springer, Singapore, 2013.

[385] Colin Wilkes and Anthony J. Dean. Gas Fuel Conditioning System Design Considerations for Utility Gas Turbines. In *ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition*, pages 3–10, Orlando, Florida, USA, Jun 1997.

[386] Allen J. Wood and Bruce F. Wollenberg. *Power Generation, Operation, and Control*. Wiley-Interscience, New York, 3 edition, 2013.

[387] H. C. Wood, T. S. Sidhu, M. S. Sachdev, and M. Nagpal. A General Purpose Hardware for Microprocessor Based Relays. In *International Conference on Power System Protection '89*, pages 43–59, 1989, Sept. 1989.

[388] Hong-Tzer Yang, Pai-Chum Yang, and Ching-Lien Huang. Evolutionary Programming Based Economic Dispatch for Units with Non-Smooth Fuel Cost Functions. *Power Systems, IEEE Transactions on*, 11(1):112–118, Feb. 1996.

[389] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary Programming Made Faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82–102, 1999.

[390] Özgür Yeniay. Penalty Function Methods for Constrained Optimization with Genetic Algorithms. *Mathematical and Computational Applications*, 10:45–56, 2005.

[391] L. Yu, X. Jin, Z. Li, Y. Xiong, and W. Huang. An Intelligent Scheduling Approach for Electric Power Generation. *Chinese Journal of Electronics*, 27(6):1170–1175, Nov. 2018.

[392] X. Yu and M. Gen. *Introduction to Evolutionary Algorithms*. Decision Engineering. Springer Science+Business Media, London, UK, 2010.

[393] P. Zarco and A. G. Exposito. Power System Parameter Estimation: A Survey. *IEEE Transactions on Power Systems*, 15(1):216–222, Feb 2000.

[394] H.H. Zeineldin, E.F. El-Saadany, and M.M.A. Salama. Optimal Coordination of Overcurrent Relays Using a Modified Particle Swarm Optimization. *Electric Power Systems Research*, 76(11):988 – 995, 2006.

[395] Mohamed Zellagui and Almoataz Youssef Abdelaziz. Optimal Coordination of Directional Overcurrent Relays using Hybrid PSO-DE Algorithm. *Inter. Electrical Engineering Journal (IEEJ)*, 6(4):1841 – 1849, 2015.

[396] H. Zhang, G. T. Heydt, V. Vittal, and J. Quintero. An Improved Network Model for Transmission Expansion Planning Considering Reactive Power and Network Losses. *IEEE Transactions on Power Systems*, 28(3):3471–3479, Aug 2013.

[397] Weiguo Zhao and Liying Wang. An Effective Bacterial Foraging Optimizer for Global Optimization. *Information Sciences*, 329:719–735, 2016.

[398] You-He Zhou, Qin Shu He, and Xiao Jing Zheng. Attenuation of Electromagnetic Wave Propagation in Sandstorms Incorporating Charged Sand Particles. *The European Physical Journal E*, 17(2):181–187, Jun 2005.

[399] Jizhong Zhu. *Optimization of Power System Operation*. Wiley-IEEE Press, Piscataway, N.J. Chichester, 2009.