AUGMENTED AND EXACT LAGRANGIAN APPROACHES TO CONTINUOUS CONSTRAINED OPTIMIZATION WITH EVOLUTION STRATEGIES

by

Jeremy Porter

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

at

Dalhousie University
Halifax, Nova Scotia
August 2022

For Miles and Owen.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis considers variations on Lagrangian approaches to constraint-handling in the context of stochastic black-box optimization. The augmented Lagrangian is one such approach from the well-known method of multipliers that transforms a constrained problem into a sequence of unconstrained problems. Iterative updates to the Lagrangian parameters are designed to use each solution in the sequence of problems to drive the next solution closer to the desired optimum of the constrained problem. We review a novel adaptation of this method for evolution strategies that simultaneously updates Lagrangian parameters alongside internal parameters in order to avoid the cost of converging to intermediate values that become obsolete later in the sequence. Existing implementations of this adaptation are compared analytically and experimentally, and a new weakness highlighted.

This investigation leads to proposing a new algorithm for constrained optimization that for the first time adapts an exact Lagrangian approach for use with evolution strategies. This is related to augmented Lagrangian evolution strategies in that it forms iterative updates for Lagrangian parameters such that convergence to an optimum in the search space corresponds with convergence to optimal Lagrange multipliers. The approach is distinguished however by framing the multipliers as dependent on position in the search space rather than as separate parameters and by approaching a solution through solving implicit quadratic subproblems with identical optimal multipliers. Along with comparisons on selected benchmark results from the literature, the exact Lagrangian method is compared experimentally on a range of archetypal test problems against previous implementations using the augmented Lagrangian approach, and found to compare favourably. These results are further justified by single-step analyses of an evolution strategy on the exact Lagrangian function.

Acknowledgements

First and foremost, this document and the work it represents would not have been possible without my wife, who has supported and encouraged me in innumerable ways over the years. So, while a single line hardly seems to do it justice: thank-you, Carolyn.

Thanks to my parents, Jerry and Kathy, and siblings, Nick and Laura, who have been rocks for me since forever, and are always willing to engage with me, despite maybe a certain glazing-over of the eyes, while I talk at length.

Finally, many thanks are owed to my supervisor Dr. Arnold for his patience and insight throughout the entire process of my degree, and in particular while putting this thesis together. I'm also very appreciative of the many helpful and supportive comments both before and after my defence from Dr. Heywood, Dr. Trappenberg, and Dr. Akimoto.

Chapter 1

Introduction

1.1 Optimization

The study of optimization is a rigorous approach to solving problems that require determining an input for a given process in order to achieve a desired outcome: finding a path that minimizes distance traveled, or a manufacturing plan that balances safety and efficiency, are both examples that can be framed as optimization problems. This thesis specifically considers optimization problems with a continuous domain or *search space*. Mathematically, we seek a value x that maximizes or minimizes an *objective function* $f(x)$. The solution is called a *global optimum*, and is a point x^* for which $f(x^*) \leq f(x)\ \forall x \in \mathbb{R}$. Algorithmically, it is often sufficient to find a *local optimum* x^* which satisfies $f(x^*) \leq f(x)$ within a local neighbourhood $\mathcal{N}(x^*)$ which is an open set containing x^*. Since maximizing and minimizing are equivalent operations up to a change of sign, I use the term minimization to refer to them both. The same concepts extend directly to when $f : \mathbb{R}^n \to \mathbb{R}$ is a function mapping n-dimensional vectors $\boldsymbol{x}$ to a scalar objective value $f(\boldsymbol{x})$, in which case we seek an optimum $\boldsymbol{x}^* \in \mathbb{R}^n$.

Depending on the particulars of the objective function, various methods exist for solving such problems. If derivative information is not available, a solution may be undertaken using *derivative free optimization*. If additionally no analytic definition of $f(\boldsymbol{x})$ is available, *black-box* optimization methods are employed that operate using only the computed value $f(\boldsymbol{x})$. Black-box methods assume no other knowledge of the internals of the function or how the value is calculated, and in particular no derivative information is assumed. Evolution strategies (ES) are a prominent example of stochastic black-box methods that are inspired heuristically by evolution in nature: a candidate solution is iteratively modified by combining information sampled locally and randomly that is selected with a bias for improvement.

Evolution strategies have been used effectively for solving a wide variety of problems [1, 33, 91, 47, 37], and in the context of continuous optimization problems under certain assumptions are known to converge log-linearly[1] on convex-quadratic problems [53]. It remains an open question which constraint-handling methods are most appropriate for use with evolution strategies, and a central goal of this thesis is to propose an approach that will demonstrably improve on existing methods. Efficacy of the algorithm will be justified both through a type of single-step analysis previously used to investigate constraint-handling with evolution strategies [92] as well as experimental validation.

1.2 Constrained optimization

Constrained optimization considers approaches to problems where the domain of solutions is somehow restricted: we seek a point $\boldsymbol{x}^*$ which minimizes an objective function $f(\boldsymbol{x})$ while also satisfying a set of constraints. The restrictions imposed by constraints partition the search space into two complementary regions: the *feasible* region of points that satisfy all constraints, and the *infeasible* region where at least one constraint is violated. At a general level, constraints can be classified using a taxonomy [75] that distinguishes between important characteristics using four named and largely orthogonal partitions. This in turn distinguishes between optimization algorithms that are applicable on different types of constraints.

Under this taxonomy, the majority of this thesis will focus on Quantifiable (as opposed to non-quantifiable), Relaxable (as opposed to unrelaxable), and Known (as opposed to hidden) constraints, given as QR*K in the taxonomic notation. Respectively, these taxonomic classifications imply that every constraint can be evaluated as a numerical value, infeasible points still have meaningful objective function values, and the number of constraints is known before execution. This matches with the assumptions of much of the literature from numerical optimization, where constraints are implicitly assumed to be of type QRAK. The additional letter indicates that the constraints

[1] Stochastic log-linear convergence can be thought of as the expected difference between the logarithms of the distance from the optimum in successive iterations tending to a constant value. If $\Delta^{(k)}$ is the distance from the optimum in iteration k, then log-linear convergence (in expectation) implies $\lim_{k\to\infty} \mathbb{E}\left[\log\left(\Delta^{(k)}\right) - \log\left(\Delta^{(k+1)}\right)\right] = c$, for some positive constant c. See [103, 53].

must also be given A priori (as opposed to by simulation). The notion of constraints being determined a priori overlaps significantly with considering constraints that are of negligible cost to calculate relative to evaluating the objective function.

In the case of quantifiable and known constraints (Q**K), it is usually convenient to further separate them into two categories: equality constraints where feasible points satisfy $g_i(\boldsymbol{x}) = 0,\ i \in \mathcal{E}$, and inequality constraints where feasible points satisfy $g_i(\boldsymbol{x}) \leq 0,\ i \in \mathcal{I}$. As with objective functions, constraints in the black-box setting are treated by using only the values of $g_i(\boldsymbol{x})$, and without relying on the specific definitions of the functions g_i or assuming any derivative information. This justifies the use of the wildcard * in the QR*K taxonomic classification for this thesis, implying that a priori constraint information is permitted but not required.

1.3 Numerical optimization

Numerical optimization here broadly refers to the overlapping collection of classical approaches from numerical analysis including, among others, linear programming, quadratic programming, nonlinear programming, convex programming, and non-differentiable optimization. These stem historically from attempting to devise rigorous solutions for decision and planning problems involving multiple variables, and frame their approaches in terms of the underlying mathematical structure of the problems. Awareness of this structure is a key distinction between the assumptions underpinning the methods of numerical optimization and those of black-box methods like evolution strategies. In spite of this, the insights carry over in a natural way that justifies their study in the context of black-box optimizers. These are important to understanding and justifying the approach I take to constraint-handling for evolution strategies, so while a brief outline is given here of several important concepts from numerical optimization, additional details are given in Appendix B at the cost of additional exposition.

In unconstrained optimization, the minimum for a sufficiently smooth (differentiable) objective function $f(\boldsymbol{x})$ can be characterized in terms of its first and second derivatives, analogously to the one-dimensional case from introductory calculus. It is necessary for the minimum point $\boldsymbol{x}^*$ to satisfy $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*) = 0$ for instance, which states

that the gradient at the optimum must be zero. Relatedly, any point where the gradient is zero is a *stationary point*. Throughout the search space, the value of the gradient gives useful first-order information about the objective function that can be exploited. One of the simplest examples of this is in the method of *gradient descent* (equivalently *ascent*) which attempts to iteratively converge to the minimizer of f from an arbitrary starting point within a neighbourhood of the optimum. In the k-th iteration, the gradient at the current point is calculated as $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)})$, and the next point is determined by making a step in the negative direction of that gradient as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - a^{(k)} \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)})$$

with the scalar $a^{(k)}$ controlling the length of the step, and superscripts throughout indicating the associated iteration number. Since the negative gradient points in the direction of greatest decrease for f, we expect that moving in this direction will lead to a point with decreased objective value. With sufficient iterations and appropriately chosen step sizes, the sequence of points $\{\boldsymbol{x}^{(k)}\}$ generated by these steps can be shown to converge to the minimum $\boldsymbol{x}^*$ on continuously differentiable convex problems.

In constrained optimization, similar statements can be made about the combination of objective f and constraint functions g_i by using their respective first- and second-order derivative information. The most significant result of these characterizations is the idea of *Lagrange multipliers* that are expressed as the coefficients α_i in the *Lagrangian* function

$$L(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \alpha_i g_i(\boldsymbol{x}).$$

This function expresses a relationship between the m constraints and the objective function in such a way that under certain conditions, the point that minimizes $L(\boldsymbol{x})$ corresponds to the point $\boldsymbol{x}^*$ that minimizes $f(\boldsymbol{x})$ while satisfying the constraints g_i. In other words, solving the constrained problem may be possible simply by applying standard unconstrained minimization routines to $L(\boldsymbol{x})$. Framed this way, the Lagrangian resembles a type of penalty function that transforms the constrained optimization problem into one of unconstrained optimization, similar in spirit to the

penalty function

$$Q(\boldsymbol{x}) = f(\boldsymbol{x}) + \omega \cdot g(\boldsymbol{x})^2$$

that quadratically penalizes points away from the constraint boundary. Like the Lagrangian $L(\boldsymbol{x})$, it can also be shown under certain conditions that the point $\boldsymbol{x}^*$ that minimizes the constrained problem defined by objective function f and constraint functions g_i is a point that minimizes $Q(\boldsymbol{x})$. The key difference between the two unconstrained formulations lies in their parameters. In order to achieve convergence for $Q(\boldsymbol{x})$ using an iterative approach, the penalty coefficient ω will often need to be increased to become very large, and convergence proofs will even assume that $\omega \to \infty$. This creates problems with numerical stability, and significantly increases the *ill-conditioning* of the problem, meaning small changes in $\boldsymbol{x}$ are able to have disproportionately large impacts on the value of $Q(\boldsymbol{x})$. Both of these are serious issues for unconstrained optimizers. On the other hand, the parameters α_i for $L(\boldsymbol{x})$ have finitely-bounded values that are provably optimal under certain conditions. This ties in closely with the first- and second-order characterization of the problem, and the result is that in fact there is a *Karush-Kuhn-Tucker pair* of optimal vector values $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*)$ that gives the minimizer of the constrained problem and that corresponds to the point minimizing $L(\boldsymbol{x})$. Rather than having to increase the parameter ω to become arbitrarily large in order to achieve convergence on $Q(\boldsymbol{x})$, this result means we only have to accurately set the values of α_i in the expression of $L(\boldsymbol{x})$ in order to be able to find the constrained minimizer.

A visual example is given in Figure 1.1 for the objective function $f(x) = x^2$ (blue lines) and single constraint function $g(x) = 2 - x = 0$. In the left-most plot, the curves resulting from

$$L(x, \alpha) = f(x) + \alpha g(x)$$

using various choices of Lagrange multiplier α are shown along with their associated minima. Since the optimal choice of Lagrange multiplier is $\alpha^* = 4$ for this problem, the curve for $L(x, 4)$ (red lines) shares its unconstrained minimum with the solution of the constrained problem at $x = 2$. In the right-most plot, lines

$$\ell(x) = \alpha(2 - x)$$

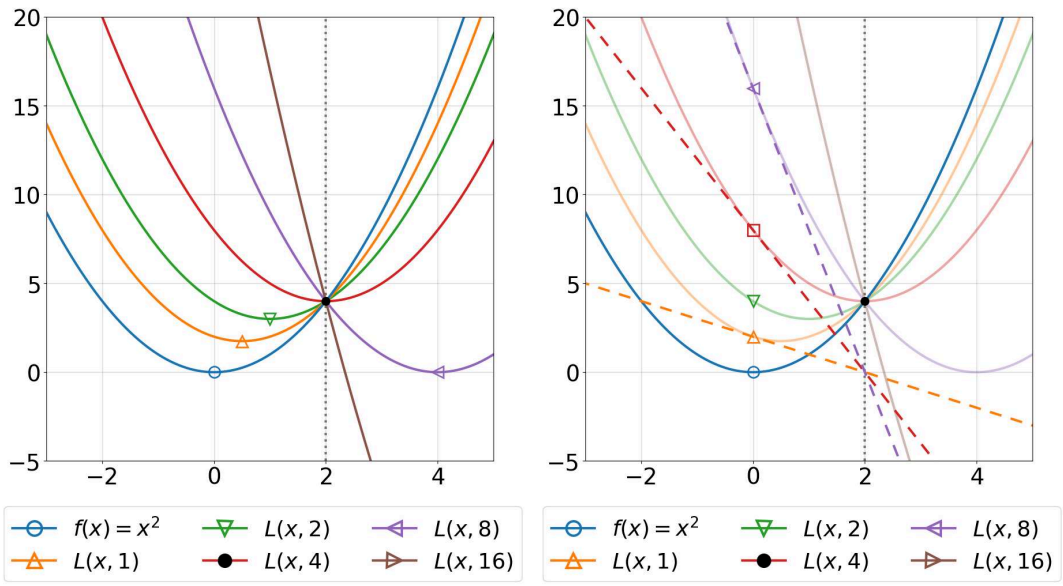

Figure 1.1: Visualizations in $n = 1$ of objective function $f(x) = x^2$ with equality constraint $x = 2$ and the Lagrangians $L(x, \alpha)$ resulting from $\alpha = 2^k$ for $k = 0, \ldots, 4$. The optimal multiplier is $\alpha^* = 4$. At left, the minimal points are marked for each curve $L_0(x, \alpha)$. At right, the intersection is marked between each curve $L(x, \alpha)$ and the line $\alpha(2 - x)$. Figure B.1 gives the analogous case for an inequality constraint.

are additionally shown for selected increasing values of α. These equations $\ell(x)$ represent the second half of the respective Lagrangian functions, and geometrically the lines are seen to lead to a "shift" of the objective function that results in a curve which shares its minimum with the solution of the constrained problem. Choosing a value of α that is distant from the optimum value α^* results in a curve with a solution distant from the solution of the constrained problem.

At first, it seems that we may have only shifted the difficulty from finding $\boldsymbol{x}^*$ to now additionally finding optimal α_i^*. However, it is possible to derive relatively precise update rules for the Lagrange multipliers α_i, that even operate independently of the minimization of $\boldsymbol{x}$. The *method of multipliers* gives one such approach by combining the quadratic penalty and Lagrangian functions given above into the single expression

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \left(\alpha_i g_i(\boldsymbol{x}) + \omega_i g_i^2(\boldsymbol{x}) \right)$$

referred to as the *augmented Lagrangian*, having parameter vectors $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_m]$

and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]$ for the penalty coefficients and Lagrange multipliers, respectively.

A visual example of the augmented Lagrangian is given in Figure 1.2 for the TR2 sphere problem showing objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ (blue contour lines) and both the infeasible region (shaded grey) and constraint boundary (dashed lines) for the linear inequality constraint $2 - x_1 - x_2 \leq 0$. Shaded contour regions are given for three augmented Lagrangian functions resulting from using the optimal $\alpha^* = 2$ and unit penalty coefficient $\omega = 1$ (top right), from increasing the penalty coefficient by a factor of 20 (bottom left), and from increasing the Lagrange multiplier by a factor of 20 (bottom right). The colours used to indicate the contour regions are inconsistent between each plot, and are instead scaled to highlight relevant details. When using the optimal multiplier, the unconstrained minimum of $L_\omega(\boldsymbol{x}, \alpha)$ is seen to correspond with the constrained optimum $\boldsymbol{x}^* = [1, 1]$. The contour lines also indicate that the augmented Lagrangian is still reasonably well-conditioned; they are slightly stretched ellipsoids, rather than the circles seen for $f(\boldsymbol{x})$. Increasing the penalty coefficient ω while holding the multiplier α steady does not change the location of the unconstrained optimum of the augmented Lagrangian, but it significantly increases the ill-conditioning near that optimum as the ellipsoids are seen to become much more elongated. Increasing the multiplier α while holding the penalty coefficient ω steady maintains the milder conditioning while shifting the unconstrained minimum of $L_\omega(\boldsymbol{x}, \alpha)$ far from the constrained optimum.

It can be proven [43, 27] that so long as the penalty coefficients satisfy a lower bound $\omega_i > \bar{\omega}$ for a finite value of $\bar{\omega}$, the augmented Lagrangian function is appropriate for use with unconstrained minimization when the Lagrange multipliers are updated in the k-th iteration using a variation of

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \omega_i^{(k)} g_i(\boldsymbol{x}^{(k)}).$$

Even more, this can be proven to give a sequence of Lagrange multiplier vectors $\{\boldsymbol{\alpha}^{(k)}\}$ that converges to the optimum KKT point $\boldsymbol{\alpha}^*$ under mild conditions. Because of this prescribed update rule, a general unconstrained optimization routine is suitable for minimizing $L_{\boldsymbol{\omega}}$ in order to thereby find the constrained minimizer $\boldsymbol{x}^*$. This makes

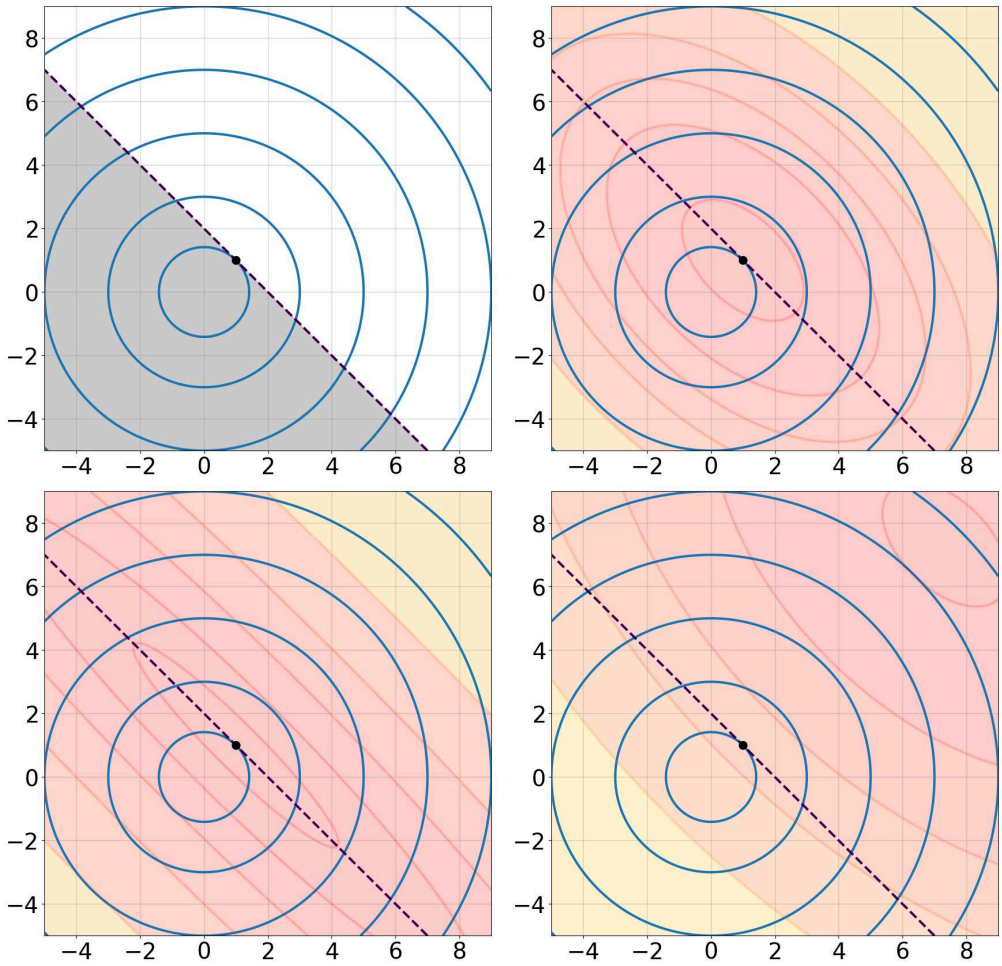

Figure 1.2: Visualizations in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with inequality constraint $g(\boldsymbol{x}) = 2 - x_1 - x_2 \leq 0$. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $L_\omega(\boldsymbol{x}, \alpha)$ with $\alpha = 2,\ \omega = 1$. Bottom left: contour regions for $L_\omega(\boldsymbol{x}, \alpha)$ with $\alpha = 2,\ \omega = 20$. Bottom right: contour regions for $L_\omega(\boldsymbol{x}, \alpha)$ with $\alpha = 40,\ \omega = 1$. The constrained optimum is marked throughout at $\boldsymbol{x}^* = [1, 1]$.

use of the method of multipliers and augmented Lagrangian a broadly attractive approach.

1.4 Constrained optimization with evolution strategies

The method of multipliers and associated augmented Lagrangian function have been used extensively in continuous numerical optimization, and stochastic and evolutionary optimization methods have also implemented variations of the approach many

times [121, 97, 35, 80, 81, 78, 79, 23, 24], much of the work occurring in the last decade. A consistent feature between these varied implementations is their adherence to the basic inner/outer loop model implied by the original method of multipliers: an outer loop sets the values of the Lagrangian's parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ in order to determine an unconstrained problem that is operated on by an optimizer in the inner loop. Once a solution is found, this is fed back to the outer loop in order to update the Lagrangian parameters and repeat the cycle until convergence is observed to the optimum. Although the original method of multipliers assumes that each inner loop will find an *exact* solution for the given Lagrangian parameters, and some convergence results even rely on this assumption, in practice it seems that only an approximation to the solution is needed within some reasonable bounds.

A novel augmented Lagrangian algorithm for evolution strategies (the AL-ES) was proposed as part of early work for this thesis [14] in the context of problems with a single linear constraint. A significant contribution of the AL-ES was to propose updates for the Lagrangian parameters as being integrated alongside other internal parameter updates of the evolution strategy. The motivating idea was that perhaps the limits of reasonable bounds for finding a solution of the inner loop could be pushed sufficiently far that the evolution strategy could keep pace with changes to the Lagrangian parameters within a single iteration. If so, then considerable computational expense could be saved. With integrated updates, there would be no need to converge to intermediate results within each inner loop that would ultimately become obsolete in subsequent iterations. This was demonstrated to be the case through single-step analysis and experimental results, in large part due to careful updates of the penalty coefficient $\boldsymbol{\omega}$ that maintains balance between the evolution strategy's progress on the constraint and objective functions.

The AL-ES was analyzed by Atamna et al. [16] who used Markov chain analysis to describe linear convergence results for a single constraint. Similar results were given by the same authors for a version of the AL-ES modified for multiple constraints [18, 19], and for the AL-ES integrated with covariance matrix adaptaion (CMA) on a single constraint [17]. Encouraging empirical results on archetypal problems were shown for both modified algorithms. The thread of AL-ES convergence results from

Markov chain analysis also formed part of the PhD thesis of Atamna [15]. Dufossé and Hansen considered an adaptation of the AL-ES for use with surrogate functions [38], and additionally performed a parameter study to suggest improved parameter values for the AL-ES when using covariance matrix adaptation (the AL-CMA-ES). Empirical results on a selection of benchmark problems showed improved performance for the AL-CMA-ES when using those parameter settings.

The inclusion of CMA is an important modification. It is the "de facto standard" [53] for continuous optimization with evolution strategies, to the extent that it is reasonable to partition ES algorithms into those that use CMA and those that do not. Although its implementation invokes some technical detail, a key concept is that it uses an approximation of the inverse of the local Hessian [52, 110] to make more informed choices about where the evolution strategy should sample next. This will significantly reduce the impact of ill-conditioning, as the evolution strategy is often observed to adaptively determine an appropriate scaling of the local search space. Put another way, the effect of including covariance matrix adaptation is that the evolution strategy is ideally able to operate on a search space resembling a sphere or reasonably well-conditioned ellipsoid, with the CMA encoding the transformation between that almost-spherical space and the true search space. For any method that uses evolution strategies then, it remains important to understand its behaviour on those sphere and ellipsoid search spaces when no CMA is employed.

1.5 Contributions

Work from this thesis led to proposing the $(1 + 1)$-AL-ES [14], a novel constraint-handling approach for evolution strategies that was considered on convex quadratic problems with a single linear constraint. Single-step analysis revealed that updates to Lagrangian parameters should be done with the goal of balancing the progress of an evolution strategy in the constrained subspace with that in the unconstrained subspace. Experimental results showed log-linear convergence on spheres and moderately conditioned ellipsoids.

This thesis will demonstrate that existing extensions of the AL-ES, in spite of promising published results, can exhibit poor performance on well-conditioned, spherical

problems after the addition of even small numbers of linear constraints. The most notable of these is when the constraint boundaries form a narrow feasible region. An example in two dimensions with two constraints is when the constraint boundaries form nearly antiparallel lines with the optimum lying at their intersection. The resulting augmented Lagrangian becomes ill-conditioned, introducing a significant impediment to convergence that is partly covered by application of CMA (which greatly reduces the negative impacts of any ill-conditioning) but persists without. The ability of CMA to correct for the ill-conditioning is also directly affected by how effectively the penalty parameter $\boldsymbol{\omega}$ is adapted, yet the penalty update rule used for AL-ES is designed to give good update steps for the Lagrange multipliers, not necessarily to give good information about the relative constraints. Finally, by considering the AL-ES update rule for the Lagrange multipliers as a type of gradient ascent for optimizing a dual function, I will argue that effective progress of the AL-ES toward the constrained optimum should depend both on progress of the evolution strategy in the primal search space and on good progress of the gradient ascent method in the dual space. As a result, problems that are ill-conditioned in the dual space will progress more slowly toward the optimal Lagrange multipliers, regardless of progress in the primal search space. I will show experimentally that the performance on certain problems of the existing CMA adaptations of the AL-ES can be improved upon, even without the use of CMA.

This improvement is achieved by the proposed exact Lagrangian approach for evolution strategies (the EL-ES) forming the core of my thesis. This is a result of my investigation into alternative ways of including constraint information as part of the AL-ES and accounting for ill-conditioning outside the use of CMA. The EL-ES relies on unconstrained optimization of a function similar in form to the augmented Lagrangian, but justifies its Lagrange multiplier updates in a different way that is more aligned with the iterative and stochastic nature of an evolution strategy. By adapting an early approach from Fletcher [41] that defines an *exact Lagrangian* function, the optimizing step in any given iteration of the EL-ES can be understood as solving certain quadratic subproblems that are defined for any candidate solution reached by the ES. The state of the exact Lagrangian function operated on by the evolution strategy is therefore defined completely in terms of the current position in the search

space. In order to calculate the needed values related to each quadratic subproblem, it becomes necessary to approximate which constraints are active at the optimum. This is accomplished through heuristics proposed as part of the EL-ES for managing the working set of constraints likely to be active.

The efficacy of this novel approach for handling constrained optimization problems with evolution strategies is supported both by theoretical and experimental results. The rule used for updating the Lagrange multiplier is justified through performing a single-step analysis, similar to the theoretical analyses performed on the original AL-ES [92] and on a class of constrained problems consisting of linear objective functions and conically constrained feasible regions [14]. Experimental results on a range of archetypal and benchmark problems from the literature additionally demonstrate that the EL-ES is competitive on the selected problems when compared on number of function evaluations used to converge.

1.6 Summary and outline

The remainder of this thesis is organized as follows.

Chapter 2 summarizes evolution strategies as a stochastic approach to continuous optimization, and provides an overview of the literature on constrained optimization that highlights approaches using Lagrangian methods and approaches for evolution strategies. Specific criteria are outlined for allowing comparisons between the variety of distinct approaches to constrained optimization.

Chapter 3 outlines the method of multipliers (augmented Lagrangian) in Section 3.1 and exact Lagrangian approach in Section 3.2, both in the context of constrained numerical optimization where the objective and constraint functions (and their related derivatives) can be written analytically.

Chapter 4 presents black-box evolution strategy implementations of the augmented Lagrangian approach (AL-ES) in Section 4.1 and the proposed exact Lagrangian approach (EL-ES) in Section 4.3. For the AL-ES, both the original proposal and subsequent variants are considered and compared in Section 4.2. Key connections between both approaches are given in Section 4.4, including a single-step analysis of

the novel EL-ES algorithm.

Chapter 5 provides experimental results validating the efficacy of the proposed EL-ES. Archetypal sphere and ellipsoid problems are considered in Section 5.2, including with randomly generated linear constraints that sample from the full range of possible orientations. Section 5.3 compares on selected benchmark problems that are widely used in the literature. Section 5.4 compares on a recently proposed scalable problem based on the Klee-Minty hypercube.

Chapter 6 summarizes and discusses the results of the thesis holistically, and presents several viable avenues for future research.

The additional contents of the appendices are included as valuable information that would nonetheless interrupt the flow of presentation elsewhere in the thesis. Appendix A provides a detailed description of the constrained optimization problems used from the literature. Appendix B contains a general overview of optimization with particular emphasis on understanding the Lagrangian approach to constraint handling. Appendix C consists of additional plots and figures supporting the experimental results of Chapter 5.

Chapter 2

Background and literature review

Constraint handling for evolutionary algorithms is an active and diverse area of research. For black-box optimizers like evolution strategies, which have no explicit knowledge of the underlying constraint or objective functions, a fundamental issue is establishing a balance between managing feasibility of individuals and improving their objective function values. Achieving this balance is very specific to both the underlying problem's definition and the optimizer being used, and the result is an abundance of constraint handling methods [84, 32]. The evaluation and subsequent comparison of these methods is not always straightforward.

In this chapter, I focus on providing relevant background information and surveying the current state of the literature for constrained optimization relevant to evolution strategies. In Section 2.1, I give a simplified overview of evolution strategies as applied to unconstrained optimization, along with pseudo-code and explanations of several representative implementations. Section 2.2 discusses three separate criteria I use for making comparisons between alternative approaches and with my own work: benchmark performance, archetypal problem performance, and constraint handling classification. In Sections 2.3 - 2.6, I survey comparable evolutionary and stochastic approaches to constrained optimization, in particular those using penalty or augmented Lagrangian methods. Section 2.7 highlights approaches specifically applied to evolution strategies, excepting those related to AL-ES which are covered in detail in Chapter 4.

2.1 Evolution strategies

Evolution strategies (ES) [29, 53] are a well-established class of iterative, stochastic algorithms for solving black box unconstrained optimization problems. They are

comparison based, and so operate by using only objective function evaluations. Evolution strategies are considered an example of the broader category of *evolutionary algorithms*, although often the only shared feature with other evolutionary algorithms is the heuristic for their original development being nature-inspired. Roughly speaking, evolution strategies move through a search space by evaluating local samples about the current iteration's candidate solution, then combining this information to determine a candidate solution for the next iteration. If the transitions between iterations are appropriately biased towards improved solutions, then with enough iterations an evolution strategy may converge towards an optimum. The subsequent performance of an algorithm is measured both in its ability to converge and in the number of iterations (or function evaluations) required to do so.

One key feature of success for an ES algorithm lies in adapting the *step size*, an internal parameter that determines the variance of each local sample. Intuitively, the step size should grow when successive iterations show a sufficient improvement in $f(x)$, and shrink when the values of $f(x)$ for candidate solutions stagnate or degrade. In the former case, this leads to sampling with larger variances, potentially saving iterations by replacing many smaller successful steps with fewer large ones. The latter case encourages smaller variances for sampling, thereby decreasing the chance of making an unsuccessful step and permitting convergence when in the neighbourhood of a local optimum.

In a general way, the function of an evolution strategy can be broken down into four steps: first, given an existing set of *parent* candidate solutions, a set of offspring is generated through stochastic sampling scaled by a step-size σ and centered on the parent(s). Second, these offspring are evaluated based on an *objective function* that takes as input one individual and returns a score that the ES optimizer is seeking to minimize (this is without loss of generality, as if maximization is desired then we may consider the negative of the objective function). Third, the offspring are combined in order to form the parent solution(s) for the next generation in such a way that bias is introduced toward improving objective function scores. And fourth, the step-size σ is updated in a manner that allows it to decrease in the vicinity of a local minimum in order to achieve convergence.

Defining exactly how each of these four steps are performed determines which of the many flavours of evolution strategy will be used, although some steps are more distinguishing than others. For instance, it is almost universal that a normal distribution is used when performing the stochastic sampling needed in the first step. This is justified by the normal distribution being both a distribution of maximum entropy and inherently isotropic [53, 54]. Given a parent candidate solution $\boldsymbol{x}^{(k)}$ in iteration k, the next candidate solution could therefore be created by computing

$$\boldsymbol{y} = \boldsymbol{x}^{(k)} + \sigma\mathcal{N}(\mathbf{0}, \mathbf{I})$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ represents the n-dimensional normal distribution, centered at $\mathbf{0}$ and with *covariance matrix* $\mathbf{I}$ (the identity matrix in dimension n). This applies a *mutation* to the original candidate solution, resulting in a new candidate which then must be evaluated as part of a selection process.

We will see that modifying this covariance matrix and updating the step-size σ between iterations are two key areas where designs of evolution strategies may differ. The step size controls the expected sample distance from the distribution's centre point, and directly affects the convergence of the algorithm; setting σ too large will prevent the ES from sampling candidate offspring that improve upon their parent, while setting σ too small may cause the ES to converge prematurely to a non-stationary point. Meanwhile, the covariance matrix gives control over the relative directions emphasized by the sample; using the identity matrix leads to an isotropic sampling distribution in all dimensions, while other positive-definite matrices dictate the relative preferences between dimensions. In practice, this models various levels of ill-conditioning of the Hessian, as recent directions with relatively large improvement can be sampled with greater frequency (in accordance with the covariance matrix). This is given more explicitly by the description of CMA-ES in Section 2.1.3.

There are several combinations of selection and recombination operations that give viable evolution strategy variants. A shorthand notation is used to encapsulate these differences, given as $(\mu/\rho\dagger\lambda)$. Taken from right to left: the value of λ indicates the population size or number of offspring generated in each iteration, the value of ρ indicates the number of parent individuals used to generate each offspring, and the

value of μ indicates the total number of parent individuals in each iteration. The value of ρ is sometimes omitted; this has been used to refer to the case of $\rho = \mu$ when it is clear from context, but the case of $\rho = 1$ is otherwise to be assumed. Additionally, the separator indicates whether elitism is used in the selection process: a plus indicates that the best individuals are maintained between generations (so μ parents are chosen from $\mu + \lambda$ individuals), while a comma indicates that each generation is evaluated separately (so μ parents are chosen from λ individuals). As an example of interpreting this notation, the $(1 + 1)$-ES is an evolution strategy that generates a single offspring using a single parent, and in each iteration allows the best of those two individuals to continue to the next iteration. This is a single-membered evolution strategy with elitism. Another example is the $(\mu/\mu, \lambda)$-ES, which is an evolution strategy that combines the best μ offspring from a population of λ to create the next generation's parental centroid. This is a multimembered evolution strategy without elitism. Covariance matrix adaptation (CMA) may be used alongside either of these approaches, and results in one of the most competitive black-box optimizers based on benchmarking results [56, 125].

2.1.1 The $(1 + 1)$-ES

One of the simplest examples of an evolution strategy, the $(1 + 1)$-ES creates one offspring candidate $\boldsymbol{y}$ from the single parent candidate $\boldsymbol{x}$ in each generation. Both the offspring and parent are evaluated using the same objective function f, and their results compared. If $f(\boldsymbol{y}^{(k)}) < f(\boldsymbol{x}^{(k)})$ in the k-th iteration then the offspring is considered successful and it becomes the parent for the $(k+1)$-th generation as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)}.$$

Otherwise, the offspring is considered unsuccessful, and the parent $\boldsymbol{x}^{(k)}$ remains the parent for the next generation as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}.$$

In this way, unsuccessful candidates are discarded, and only those with a better objective function value are retained to become the parent for the next generation. At any

point, the current parent candidate solution is the best solution discovered thus far; this implements the concept of *elitism* in the context of evolutionary algorithms.

Control of the step-size is a critical component for any evolution strategy, including the $(1+1)$-ES. In this case, it is the only internal parameter of the algorithm that changes between iterations, and the sample distribution is always kept isotropic. One of the earliest step size control schemes is Rechenberg's 1/5-th rule [94], which updates σ in order to maintain in expectation a ratio of $1:4$ between the successful and unsuccessful offspring. A multiplicative version of this rule [70] is to compute $\sigma^{(k+1)}$ from the k-th iteration as

$$\sigma^{(k+1)} = \sigma^{(k)} \cdot \exp\left(\frac{\mathbb{S} - 0.2}{n}\right), \tag{2.1}$$

where n is the problem dimension and $\mathbb{S}$ is a binary value that indicates whether the latest offspring was successful or not. Under this rule, the step size will be increased when the proportion of successful candidates is more than 1/5, and decreased when this proportion is less than 1/5.

Algorithm 2.1 Single iteration of $(1+1)$-ES with 1/5-th rule

Require: $f : \mathbb{R}^n \to \mathbb{R}$
1: $\boldsymbol{z} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ sample from normal distribution
2: $\boldsymbol{y} \leftarrow \boldsymbol{x} + \sigma\boldsymbol{z}$
3: $\mathbb{S} \leftarrow (f(\boldsymbol{y}) < f(\boldsymbol{x}))$ ▷ boolean check
4: **if** $\mathbb{S}$ **then**
5: $\quad \boldsymbol{x} \leftarrow \boldsymbol{y}$
6: **end if**
7: $\sigma \leftarrow \sigma \cdot \exp\left(\frac{\mathbb{S}-0.2}{n}\right)$ ▷ control step size

A single iteration of the $(1+1)$-ES is given in Algorithm 2.1, using the multiplicative 1/5-th rule of Eq. (2.1).

2.1.2 The $(\mu/\mu, \lambda)$-ES

The $(1+1)$-ES has very little overhead and is a straightforward optimization method. At each iteration where an improvement is found, there is no consideration given for how much of an improvement is made; the $(1+1)$ method will accept an offspring that offers *any* improvement. This is problematic on functions exhibiting multimodality,

that are highly non-convex, or with very narrow basins surrounding the optimum point. In a general sense, the issue is that the single-membered ES does not convey sufficient information about the local neighbourhood of the parent before updating its position in the search space along with other internal parameters including the step size.

The issue of insufficient information is partly addressed through performing multiple sampling operations within a single iteration in order to generate the offspring population. The $(\mu/\mu, \lambda)$-ES approach does this by creating a set of λ offspring for each parent candidate solution $\boldsymbol{x}^{(k)}$ as

$$\boldsymbol{y}_i^{(k)} = \boldsymbol{x}^{(k)} + \sigma^{(k)} \cdot \boldsymbol{z}_i^{(k)}$$

for $i \leq \lambda$, where the $\boldsymbol{z}_i \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ are vectors with elements drawn independently and identically from the standard normal distribution. The offspring $\boldsymbol{y}_i$ are all evaluated using the objective function and re-ordered so that their indices i indicate their relative ranking according to $f(\boldsymbol{y}_i)$. The best μ offspring are then combined using the average

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \frac{1}{(\sum_{i=1}^{\mu} w_i)} \sum_{i=1}^{\mu} w_i \boldsymbol{y}_i^{(k)} \\ &= \boldsymbol{x}^{(k)} + \sigma^{(k)} \cdot \frac{1}{(\sum_{i=1}^{\mu} w_i)} \sum_{i=1}^{\mu} w_i \boldsymbol{z}_i^{(k)} \end{aligned} \tag{2.2}$$

where the coefficients w_i are weights. The new point $\boldsymbol{x}^{(k+1)}$ is referred to as a *parental centroid* as it is a combination of the μ selected previous offspring, rather than any single individual. The simplest case of chosen weights is setting $w_i = 1$ for each of the μ offspring, in which case the update becomes the usual average

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \frac{1}{\mu} \sum_{i=1}^{\mu} \boldsymbol{y}_i^{(k)} \\ &= \boldsymbol{x}^{(k)} + \sigma^{(k)} \cdot \frac{1}{\mu} \sum_{i=1}^{\mu} \boldsymbol{z}_i^{(k)} \end{aligned} \tag{2.3}$$

and this is assumed to be the case if weights are not specified. In the case that the

coefficients w_i are specified and not all equal to values of unity, the update becomes a weighted average. The shorthand notation is modified in this case to be $(\mu/\mu_W, \lambda)$ to indicate that a set W of weights is used when combining the offspring. Such weights can always be normalized so that

$$\sum_{i=1}^{\mu} w_i = 1$$

and this allows for writing the update succinctly as

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \sum_{i=1}^{\mu} w_i \boldsymbol{y}_i^{(k)} \\ &= \boldsymbol{x}^{(k)} + \sigma^{(k)} \cdot \sum_{i=1}^{\mu} w_i \boldsymbol{z}_i^{(k)}. \end{aligned} \tag{2.4}$$

Like the weights in W, the values of μ and λ are parameters to be chosen by the user, and are maintained for the entire run of an evolution strategy. Choosing values that result in a *truncation ratio* given by μ/λ in the range of 0.2 to 0.5 is typical. There is an inherent trade-off here, as larger parameter values for a given ratio will increase the number of offspring and make it more likely that a single generation will improve its candidate solution, yet may slow down the algorithm overall by using more objective function evaluations than are actually needed. There is no elitism in $(\mu/\mu, \lambda)$, as the centroid in each generation is always calculated from the offspring.

To accompany the increasing complexity in moving from a $(1 + 1)$ to a $(\mu/\mu, \lambda)$ evolution strategy, a more sophisticated step size update rule is appropriate. While the 1/5-th rule is generally successful in the context of $(1 + 1)$ strategies on well-conditioned convex problems, it enforces a fixed ratio between successful and unsuccessful iterations that may not be optimal through the entire run of the algorithm on more general problems. One alternative approach is that of *cumulative step-size adaptation* (CSA) [89, 58] which adaptively updates the step size by incorporating non-local information through the use of an exponentially fading record of successful steps.

After each iteration, the mutation vector is computed for the average of the μ best

offspring as

$$\hat{\boldsymbol{z}} = \frac{1}{(\sum_{i=1}^{\mu} w_i)} \sum_{i=1}^{\mu} w_i \boldsymbol{z}_i \tag{2.5}$$

using the same weights as in Eq. (2.2). Taking this sum involves modifying the normally distributed samples in two ways: through introducing a bias by considering only the μ best $\boldsymbol{z}_i$ according to their associated objective function values, and through combining those $\boldsymbol{z}_i$ using a linear (affine) combination. As will be seen shortly, we are interested in the information conveyed by the former process and not by the latter, so it will be necessary to normalize the resulting distribution of $\hat{\boldsymbol{z}}$ to account for its derivation from a linear combination of normally distributed variables. To do so, we recall first from elementary probability theory that:

1. normal distributions are equal if they have equal mean and variance;

2. a scalar multiple w applied to a standard normal variable $\boldsymbol{z}$ results in a variable with equal mean and modified variance w^2; and

3. the distribution of a sum of normal variables $\boldsymbol{z}_i$ drawn independently from $\mathcal{N}(m_i, \sigma_i^2)$ is equal to
$$\mathcal{N}\left(\sum_i m_i, \sum_i \sigma_i^2\right).$$

Using the above, we derive a normalizing constant μ_{eff} for the variance of $\hat{\boldsymbol{z}}$ that will account for the variances of the terms in the weighted sum. Referring to Eq. (2.5), each term is seen to be an independent normal variable with distribution

$$\mathcal{N}\left(\mathbf{0}, \left(\frac{w_i}{\sum_j w_j}\right)^2 \cdot \boldsymbol{I}\right)$$

so calculating a weighted sum *without* re-ordering for the best μ offspring would give a random variable of mean $\mathbf{0}$ and variance given by $\boldsymbol{I}$ scaled by the coefficient

$$\sum_i \left(\frac{w_i}{\sum_j w_j}\right)^2 = \frac{1}{\left(\sum_j w_j\right)^2} \cdot \sum_i w_i^2.$$

The desired normalizing constant μ_{eff} resulting in unit variance is given by the inverse of the above, so is therefore represented by

$$\begin{aligned}\mu_{\text{eff}} &= \left(\sum_i w_i\right)^2 \cdot \left(\sum_i w_i^2\right)^{-1} \\ &= \frac{\left(\sum_{i=1}^{\mu} w_i\right)^2}{\left(\sum_{i=1}^{\mu} w_i^2\right)}.\end{aligned} \tag{2.6}$$

This results in $\sqrt{\mu_{\text{eff}}} \cdot \hat{\boldsymbol{z}}$ being normalized such that it would be distributed as a standard normal variable if the samples $\boldsymbol{z}_i$ were not re-ordered in Eq. (2.5) to only select the μ best individuals. Note that in the case of using a weighted average with weight values already normalized as in Eq (2.4), the expression for the constant becomes

$$\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1}.$$

Similarly, in the case of using weights equal to unity as in Eq. (2.3), the expression for the normalizing constant becomes simply

$$\mu_{\text{eff}} = \mu.$$

An exponential record is maintained of the average mutations $\hat{\boldsymbol{z}}$, termed the *search path* or *evolution path* and calculated as

$$\boldsymbol{s}^{(k+1)} = (1-c)\boldsymbol{s}^{(k)} + \sqrt{\mu_{\text{eff}} c(2-c)}\hat{\boldsymbol{z}} \tag{2.7}$$

where the $c \in (0,1)$ is a user chosen parameter controlling the rate of exponential fading. The coefficient $\sqrt{\mu_{\text{eff}}(2-c)/c}$ involves the normalizing constant μ_{eff} already derived. The extra terms result from the geometric series [89]

$$\lim_{k\to\infty} \sqrt{(c\cdot(1-c)^0)^2 + (c\cdot(1-c)^1)^2 + \cdots + (c\cdot(1-c)^k)^2} = \sqrt{\frac{c}{2-c}}$$

and normalize the distribution of $\hat{\boldsymbol{z}}$ with respect to the other half of the sum in Eq. (2.7). This chosen normalization ensures that if successive steps are uncorrelated, then the expected squared length of the search path $\boldsymbol{s}$ is equal to the problem dimension n.

Algorithm 2.2 Single iteration of $(\mu/\mu, \lambda)$-ES with CSA

Require: $f : \mathbb{R}^n \to \mathbb{R}, \quad c \in (0, 1), \quad D > 0$

1: **for** $i = 1 \to \lambda$ **do**
2: $\quad \boldsymbol{z}_i \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ sample from normal distribution
3: $\quad \boldsymbol{y}_i \leftarrow \boldsymbol{x} + \sigma \boldsymbol{z}_i$
4: **end for**
5: $\text{sort}([\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\lambda], [f(\boldsymbol{y}_1), \ldots, f(\boldsymbol{y}_\lambda)])$ ▷ sort $\boldsymbol{z}_i$ by values of $f(\boldsymbol{y}_i)$
6: $\hat{\boldsymbol{z}} = \dfrac{1}{\mu} \displaystyle\sum_{i=1}^{\mu} \boldsymbol{z}_k$
7: $\boldsymbol{x} \leftarrow \boldsymbol{x} + \sigma \hat{\boldsymbol{z}}$
8: $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s} + \sqrt{\mu c(2 - c)}\hat{\boldsymbol{z}}$ ▷ update $\boldsymbol{s}$
9: $\sigma \leftarrow \sigma \cdot \exp^{\frac{c}{D}} \left(\dfrac{\|\boldsymbol{s}\|}{\mathbb{E}\left[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|\right]} - 1 \right)$ ▷ update σ

An overview of $(\mu/\mu, \lambda)$-ES using CSA is given in Algorithm 2.2. On Line 9, the step size σ is updated using the search path $\boldsymbol{s}$, and a damping constant D controls how rapidly the step size can be adapted. The denominator $\mathbb{E}\left[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|\right]$ is the expected length of an n-dimensional vector with elements drawn independently and identically from a standard normal distribution, and can be calculated numerically from

$$\mathbb{E}\left[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|\right] = \sqrt{2} \cdot \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}.$$

2.1.3 The CMA-ES

A common theme between the two previous ES approaches is the isotropy of the sampling distribution; since the $\boldsymbol{z}_i$ are drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, their relative direction with respect to $\boldsymbol{x}$ is distributed uniformly. Yet the underlying search space of an optimization problem is rarely so symmetric, and ill-conditioned problems in particular can cause these approaches to use far more iterations than needed, or even converge completely to a non-stationary point.

Ideally, we would like to control the step sizes along each dimension independently

in a way that does not rely on the coordinate representation. To this end, CMA-ES [58, 51] uses a covariance matrix $\boldsymbol{C}$ to define a linear map that approximately models the inverse of the underlying Hessian of the optimization problem [110], ill-conditioned or otherwise. In each iteration, offspring $\boldsymbol{y}_i$ are generated using

$$\boldsymbol{y}_i^{(k)} = \boldsymbol{x}^{(k)} + \sigma^{(k)} \boldsymbol{C}^{(k)\,\frac{1}{2}} \boldsymbol{z}_i^{(k)} \tag{2.8}$$

where the elements of $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ are drawn independently and identically from a standard normal distribution as before, and the matrix $\boldsymbol{C}^{(k)\,\frac{-1}{2}}$ transforms these isotropic samples into samples within the space defined by the linear map of the covariance matrix. The square root notation of this matrix refers to using its eigen-decomposition as

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{D}^2\boldsymbol{B}^{\mathrm{T}}$$

with $\boldsymbol{D}$ a diagonal matrix in order to arrive at

$$\boldsymbol{C}^{\frac{1}{2}} = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^{\mathrm{T}}.$$

In each iteration, the covariance matrix is updated as

$$\boldsymbol{C}^{(k+1)} = (1-c_1-c_\mu)\boldsymbol{C}^{(k)} + c_1 \boldsymbol{p}_c^{(k+1)} \left(\boldsymbol{p}_c^{(k+1)}\right)^{\mathrm{T}} + c_\mu \sum_{i=1}^{\mu} w_i \left(\boldsymbol{C}^{(k)\,\frac{1}{2}} \boldsymbol{z}_i\right) \left(\boldsymbol{C}^{(k)\,\frac{1}{2}} \boldsymbol{z}_i\right)^{\mathrm{T}} \tag{2.9}$$

with non-negative learning constants $c_1 \leq 1$ and $c_\mu \leq 1$ and normalized weights w_i. This is an accumulated value consisting respectively of a multiple of the estimated matrix $\boldsymbol{C}^{(k)}$ taken from the current iteration, a rank one update term, and a rank μ update term. The second term is the rank one update, and is defined using the exponentially faded evolution path

$$\boldsymbol{p}_c^{(k+1)} = (1-c_c)\boldsymbol{p}_c^{(k)} + \sqrt{\mu_{\mathrm{eff}} c_c (2-c_c)} \left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right) \cdot \frac{1}{\sigma^{(k)}}$$

similar to the update rule from Eq. (2.7), with non-negative learning rate $c_c \leq 1$. This evolution path accumulates information about steps taken in the search space in order to encourage sampling in directions that have been recently successful. The

third term of Eq. (2.9) is a rank μ update that is based on the weighted average of candidates' mutations from the most recent generation.

The global step size σ is updated separately with a generalization of the CSA rule outlined in Algorithm 2.2. This generalization updates the evolution path

$$\boldsymbol{p}_\sigma^{(k+1)} = (1-c_\sigma)\boldsymbol{p}_\sigma^{(k)} + \sqrt{\mu_{\text{eff}} \cdot c_\sigma(2-c_\sigma)}\boldsymbol{C}^{(k)\,\frac{-1}{2}}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right) \cdot \frac{1}{\sigma^{(k)}} \tag{2.10}$$

in each generation, and compares this against the expected path length in n dimensions. The intent of $\boldsymbol{p}_\sigma$ is to accumulate information about successful samples $\boldsymbol{z}_i$ drawn from the isotropic distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. If the offspring $\boldsymbol{y}_i$ are generated with $\boldsymbol{C}^{\frac{1}{2}} = \boldsymbol{I}$, then the usual CSA rule in Eq. (2.7) is appropriate; however, when transforming the samples as in Eq. (2.8), the effects of the covariance matrix must be accounted for when calculating $\boldsymbol{p}_\sigma$ in Eq. (2.10).

Algorithm 2.3 Single iteration of $(\mu/\mu_W, \lambda)$-CMA-ES

Require: $f : \mathbb{R}^n \to \mathbb{R}$, $c_c, c_\sigma, c_\mu \in (0,1)$, $D > 0$, $\sum_i^\mu w_i = 1$

for $i = 1 \to \lambda$ **do**
  $\boldsymbol{z}_i \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ sample from normal distribution
  $\boldsymbol{y}_i \leftarrow \boldsymbol{x} + \sigma\mathbf{C}^{\frac{1}{2}}\boldsymbol{z}_i$ ▷ $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \sigma^2\mathbf{C})$
end for
$\text{sort}([\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\lambda], [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_\lambda)])$ ▷ sort $\boldsymbol{z}_k$ by values of $f(\boldsymbol{x}_k)$
$\hat{\boldsymbol{z}} \leftarrow \sum_{i=1}^\mu w_i \mathbf{C}^{\frac{1}{2}}\boldsymbol{z}_i$
$\boldsymbol{x} \leftarrow \boldsymbol{x} + \sigma\hat{\boldsymbol{z}}$ ▷ Update centroid

$\boldsymbol{p}_\sigma \leftarrow (1-c_\sigma)\boldsymbol{p}_\sigma + \sqrt{\mu_{\text{eff}} \cdot c_\sigma(2-c_\sigma)}\mathbf{C}^{\frac{-1}{2}}\hat{\boldsymbol{z}}$
$\sigma \leftarrow \sigma \cdot \exp^{\frac{c_\sigma}{D}}\left(\dfrac{\|\boldsymbol{p}_\sigma\|}{\mathbb{E}\left[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|\right]} - 1\right)$ ▷ update σ
$\boldsymbol{p}_c \leftarrow (1-c_c)\boldsymbol{p}_c + \sqrt{\mu_{\text{eff}} c_c(2-c_c)} \cdot \hat{\boldsymbol{z}}$
$\boldsymbol{C} \leftarrow (1-c_1-c_\mu)\boldsymbol{C} + c_1\boldsymbol{p}_c\boldsymbol{p}_c^{\mathrm{T}} + c_\mu \sum_{i=1}^\mu w_i \left(\boldsymbol{C}^{\frac{1}{2}}\boldsymbol{z}_i\right)\left(\boldsymbol{C}^{\frac{1}{2}}\boldsymbol{z}_i\right)^{\mathrm{T}}$ ▷ Eq. (2.9)

An overview of $(\mu/\mu_W, \lambda)$-CMA-ES is given in Algorithm 2.3, adapted directly from Hansen [51]. For the multimembered CMA-ES, suggested defaults for the population

parameters are given by Hansen as

$$\lambda = 4 + \lfloor 3 \ln n \rfloor, \qquad \mu = \lfloor \frac{\lambda}{2} \rfloor \tag{2.11}$$

for problems of dimension n. Similar recommended values are given for each of the other parameters, including learning rates, and are omitted here for brevity.

The $(1 + 1)$-CMA-ES

Covariance matrix adaptation can also be employed with the much simpler $(1+1)$-ES [67]. The $(1 + 1)$-CMA-ES may also make use of an active step size update scheme [68] that in the context of a $(\mu/\mu, \lambda)$-CMA-ES uses negative weightings for the worst individuals in each generation in order to shift the covariance matrix away from those directions where individuals are observed with poor objective function values. In the context of a $(1 + 1)$-CMA-ES there is only one individual in each generation, so instead a comparison is made between the individuals from the most recent iterations and the current candidate solution, in order to determine how the covariance matrix should be updated.

An overview of an implementation of the $(1 + 1)$-CMA-ES that additionally uses active updates is given in Algorithm 2.4, adapted directly from the approach proposed by Arnold and Hansen [5]. Certain implementation details are omitted in the presentation in the interest of clarity.

The matrix $\boldsymbol{A}$ is the Cholesky decomposition $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{C}$ of the covariance matrix $\boldsymbol{C}$, and in practice is used as a more cost-effective implementation for both of the covariance matrix updates [120]. This is based on the fact that if $\boldsymbol{z}$ is drawn from a standard normal distribution, then $\boldsymbol{A}\boldsymbol{z}$ gives a vector sampled from a normal distribution with covariance matrix $\boldsymbol{C}$. An exponentially fading record P_{succ} is maintained of the proportion of successful iterations. This is used in the global step size adaptation of σ using a multiplicative version of Rechenberg's 1/5-th rule, with user parameter D acting as a damping constant. An exponentially fading record $\boldsymbol{s}$ is used for updating the covariance matrix $\boldsymbol{C}$, in a manner similar to the rank 1 component of

Algorithm 2.4 Single iteration of $(1+1)$-CMA-ES

Require: $f : \mathbb{R}^n \to \mathbb{R}, \quad c, c_P \in (0,1), \quad c^+_{\text{cov}}, c^-_{\text{cov}}, D > 0, \quad \mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{C}$

$\boldsymbol{z} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample from normal distribution
$\boldsymbol{y} \leftarrow \boldsymbol{x} + \sigma\mathbf{A}\boldsymbol{z}$
if $(f(\boldsymbol{y}) < f(\boldsymbol{x}))$ **then**
    $P_{\text{succ}} \leftarrow (1 - c_P)P_{\text{succ}} + c_P$
    $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s} + \sqrt{c(2-c)}\mathbf{A}\boldsymbol{z}$
    $\mathbf{C} \leftarrow (1 - c^+_{\text{cov}})\mathbf{C} + c^+_{\text{cov}}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}$ ▷ Implicit by updating $\mathbf{A}$
    $\boldsymbol{x} \leftarrow \boldsymbol{y}$
else
    if $(f(\boldsymbol{y}) > f(\boldsymbol{x}^{(-5)}))$ **then** ▷ Compare with 5^{th} order ancestor
        $\mathbf{C} \leftarrow (1 + c^-_{\text{cov}})\mathbf{C} - c^-_{\text{cov}}(\boldsymbol{A}\boldsymbol{z})(\boldsymbol{A}\boldsymbol{z})^{\mathrm{T}}$ ▷ Implicit by updating $\mathbf{A}$
    end if
    $P_{\text{succ}} \leftarrow (1 - c_P)P_{\text{succ}}$
end if
$\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{D}\frac{P_{\text{succ}} - \frac{1}{5}}{1 - \frac{1}{5}}\right)$ ▷ Update global step size

Eq. (2.9). The active update is performed whenever the current offspring $\boldsymbol{y}$ is inferior to its fifth-order ancestor. In this case, the covariance matrix is updated to discourage sampling future offspring similar to $\boldsymbol{y}$. The principle behind active covariance matrix updates derives from Jastrebski and Arnold [68] who proposed it in the context of multimembered evolution strategies.

2.2 Criteria for comparison

Benchmarks

Problem benchmarks such as those used in competitions from the IEEE Congress on Evolutionary Computation (CEC) [77, 82, 129] aim to rank algorithms by establishing a metric of comparison (in this case, comparing solution quality with fixed budgets of function evaluations) across multiple distinct problems. The underlying metric for comparison between algorithms on a single problem is based on either distance from a known optimum or best feasible point found. These results can provide useful guidelines, although it can be difficult to extrapolate from individual benchmark results to performance on broader problem classes or even real-world examples. Hellwig and Beyer [62] observe that the CEC 2006 benchmark problems have a bias in favour of

algorithms using axis-aligned searches. This bias is less severe, though still present, in both of the CEC 2010 and 2017 problem sets, yet those problems have comparatively few constraints (an average of slightly over 2 per problem). The established metric is also at issue. Hansen et al. [55] argue that evaluating on fixed budgets gives data that is not usefully interpretable, as comparing the quantitative quality of solutions found by different algorithms does not give insight into the relative quality of the algorithms themselves. Instead, they advocate comparing the number of function evaluations needed by each algorithm to reach fixed targets.

The COCO (Comparing Continuous Optimizers) benchmark [55, 57] is a proposed framework for benchmarking and comparing algorithms, and that aims in part to address the aforementioned shortcomings. At present, the constrained portion of the benchmark is still quite new, and few published results are available using the framework.

The top results in the CEC 2006 and 2010 benchmarks [77, 82] are both variants of the ϵ-constrained differential evolution (ϵDE) approach of Takahama and Sakai [122, 123], while a 2019 update [111] to the CEC 2017 benchmark [129] ranks as first the HECO-DE approach [130, 131] of Xu et al. Results are reported according to the fixed budget CEC benchmark specifications, and so involve hundreds of thousands of function evaluations.

Problem archetypes

Work on evolution strategies has provided an alternative to pure benchmark performance by establishing an analytical framework for understanding algorithm performance through simple, archetypal test problems with known difficulties for optimization. This began with very early work from Rechenberg [94] and Schwefel [107] who established methods for parameter control on unconstrained optimization problems in part by analyzing behaviour on archetypal problems, such as the sphere and corridor fitness functions. Ellipsoidal problems are also naturally considered [20], especially in the context of non-isotropic offspring mutation mechanisms like CMA by Hansen and Ostermeier [58]. It then seems natural to consider similar archetypal problems for

constrained optimization. In order to evaluate performance and suggest new directions of research, the behaviour of an evolution strategy can be analyzed [83] under a given constraint-handling method when applied to an elementary problem. So long as these problems are chosen to have representative properties, they can be generalized to predict the behaviour of the constraint-handling mechanisms in more complicated situations.

Arnold and Brauer [13] analyze a simple $(1 + 1)$-ES on a linear objective function with single linear constraint, building on earlier work and observations from Schwefel [108] and Rechenberg [94]. Arnold continues these analyses in differents contexts, considering the impacts of repair of infeasible offspring for the $(1, \lambda)$-ES [3], as well as resampling of infeasible offspring for the $(1, \lambda)$-ES with cumulative step-size adaptation [2] and mutative self-adaptation [9]. Analysis of multiple linear constraints is facilitated by modeling a conical feasible region with the optimum point at the cone's apex for a $(1, \lambda)$-ES by Arnold [4, 10] and by Porter and Arnold [92] for the multirecombinative $(\mu/\mu, \lambda)$-ES. Related steady-state analyses are performed for conically constrained problems with repair by projection by Spettel and Beyer using the $(\mu/\mu_I, \lambda)$-ES with σ-self-adaptation [113], the $(1, \lambda)$-ES with σ-self-adaptation [114], and the $(\mu/\mu_I, \lambda)$-ES with CSA [116]. Spettel et al. [118] and Hellwig and Beyer [61] also analyze the use of a meta-ES for parameter control with conical constraints.

Linear problems like these are simple to describe, but serve as limited models for general optimization problems. The sphere model is perhaps the next simplest case, which is the class of quadratic problems with positive definite Hessian equal to a scalar multiple of the identity matrix. Knowing that even single-membered evolution strategies converge log-linearly on the unconstrained sphere, and considering that every constraint adds an extra dimension to the problem, a naive expectation would be for performance on the constrained sphere to scale approximately with the number of constraints. In that case, an ES algorithm operating on the constrained sphere with a single linear constraint should have very similar performance to the unconstrained sphere. A limited version of this problem with dimension 2 is evaluated by Kramer and Schwefel [72] in the general context of constraint handling for evolution strategies and found to be surprisingly difficult for techniques available at the time.

Constraint classification

Le Digabel and Wild [75] define a taxonomy for classifying constraints at their most general level. Algorithms for constrained optimization can then be identified using the taxonomy in order to contextualize their comparison. The *QRAK taxonomy* uses four letters that partition the constraint types, distinguishing between constraints that are Quantifiable or Non-quantifiable, Relaxable or Unrelaxable, Simulation based or A priori, and Known or Hidden. Hidden constraints (H) are not given explicitly or are not known to the solving algorithm until they are encountered, whereas known constraints (K) are given explicitly in the problem definition. Relaxable constraints (R) permit violations by candidate solutions, while for unrelaxable constraints (U) an infeasible point is not meaningfully interpretable by the objective function or other constraint functions. Quantifiable constraints (Q) confer a magnitude of violation on points in the search space and allow an ordering of more or less infeasible values, whereas non-quantifiable constraints (N) simply indicate a binary value for whether a point is feasible or infeasible.

The augmented Lagrangian and exact Lagrangian evolution strategies (AL-ES and EL-ES) considered in this thesis deal exclusively with quantifiable, relaxable, and known constraints, and so are identified as QR*K in the taxonomic notation. Quantifiable constraints potentially include both equality constraints $g_i(x) = 0,\ i \in \mathcal{E}$ and inequality constraints $g_i(x) \leq 0,\ i \in \mathcal{I}$. An important distinction can be further made between those QR*K constraints that are a priori (A), where the constraint is calculable from the input parameters to the constrained optimization algorithm, versus simulation based (S), where part or all of the optimization algorithm must be executed in order for the constraints to be evaluated. This latter case includes black box constraints where no gradient information is available. Optimization methods from classical numerical optimization, including those using augmented and exact Lagrangians, typically deal with constraints that are fully defined a priori, and so are identified as QRAK.

2.3 Penalty methods

Penalty methods transform constrained problems into unconstrained optimization problems by using a penalty term alongside the objective function. The resulting function measures fitness in the constrained search space. The quadratic penalty function $Q(\boldsymbol{x})$ for equality constraints is defined as

$$Q(\boldsymbol{x}) = f(\boldsymbol{x}) + \omega \cdot g(\boldsymbol{x})^2, \tag{2.12}$$

using both the values of the objective function f and a non-negative measure of the constraint violation, and this is used to evaluate an individual $\boldsymbol{x}$. A similar formulation can be given for inequality constraints by instead using the term $\omega \cdot \min(0, g(\boldsymbol{x}))^2$. By penalizing infeasible individuals, the quadratic penalty approach results in an unconstrained problem with an optimum point that ideally corresponds with the constrained optimum. This correspondence relies on the choice of penalty coefficient ω in the penalty term of Q, and Nocedal and Wright [128] give convergence results for the quadratic penalty approach under the assumption that $\omega \to \infty$. They note that under-penalizing with ω set too small can result in convergence to an infeasible or non-stationary point, yet over-penalizing with large values of ω induces several other problems: the Hessian of the fitness function becomes increasingly ill-conditioned, issues of numerical accuracy are encountered, and Taylor series quadratic approximations are accurate only in increasingly small neighbourhoods of the optimum point.

The simplest penalty method is the *death penalty* which discards all infeasible candidates by assigning a static "infinite" penalty to constraint violation. In the context of an evolution strategy that relies on fitness comparisons between successive generations, this may be handled by *resampling* infeasible individuals until feasible individuals are found. The behaviour of evolution strategies with resampling has been investigated by Arnold [4, 10] and Porter and Arnold [92] where it was found to lead to premature convergence to non-stationary points on certain problem types.

Finite static penalties other than the death penalty are not typically employed for stochastic approaches to constrained optimization as there may be no single value that will lead to convergence for every possible state of the algorithm. Methods using

a *dynamic penalty* attempt to modify the penalty parameter on a fixed schedule, yet this still leaves the problem of determining a reasonable schedule. Early examples include work from Joines and Houck [69], who describe a dynamic penalty method for a real-valued genetic algorithm that was later adapted for an evolution strategy by Kramer [73]. In both cases, the penalty P is steadily increased as

$$P = (C \cdot t)^A \cdot \left(\sum_i g_i(\boldsymbol{x})^\beta \right)$$

with fixed parameters $C < 1$, $\beta > 0$, $A > 1$, and iteration number given by t. Michalewicz and Attia [86] also follow a dynamic schedule as part of GENOCOP II, which employs an inner/outer loop model that increases the penalty coefficient in the outer loop after termination of a genetic algorithm optimizing a quadratic penalty function in the inner loop.

More modern penalty-based methods employ *adaptive* updates [84] of the penalty parameter by using the state information of the algorithm. Beyond some benchmark comparisons, the general performance of these methods is difficult to evaluate [25]. Given the reliance of convergence on arbitrarily large penalties, and the noted problems this creates, it seems unlikely that a simple penalty approach will be generally competitive.

2.4 Lagrangian method of multipliers

The method of multipliers refers to an approach for constrained optimization proposed independently by Powell [93] and Hestenes [64] in the context of solving equality constrained problems (ECP). As observed by Fletcher [43], the idea is that convergence properties of the quadratic penalty method of Eq. (2.12) may be preserved while avoiding the difficulties incurred by increasing the penalty to infinity. By including the Lagrangian function, the *augmented Lagrangian*

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \Omega g(\boldsymbol{x}).$$

is defined, with vector $\boldsymbol{\alpha}$ of Lagrange multipliers, and diagonal penalty matrix Ω with nonzero entries consisting of the entries of $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_m]$. The original method is presented as suitable for an iterative approach, where the Lagrange multipliers are continually approximated in a sequence $\{\boldsymbol{\alpha}^{(k)}\} \to \boldsymbol{\alpha}^*$ and solutions $\boldsymbol{x}^{(k)}$ are found for each vector of Lagrange multipliers in this sequence. The multipliers are updated as

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \Omega^{(k)} g(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)}))$$

where $\boldsymbol{x}(\boldsymbol{\alpha}^{(k)})$ is the solution $\boldsymbol{x}$ in iteration k that minimizes $L_{\boldsymbol{\omega}}$ with respect to fixed $\boldsymbol{\alpha}^{(k)}$. Under relatively mild assumptions, it can be proven [27] that this sequence of multipliers converges to $\boldsymbol{\alpha}^*$ as $\boldsymbol{x} \to \boldsymbol{x}^*$. The method of multipliers is extended to the case of inequality constrained problems (ICP) by Rockafellar [99, 98, 101], resulting in the slightly modified augmented Lagrangian given by

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_i \begin{cases} \alpha_i g_i(\boldsymbol{x}) + \frac{1}{2}\omega_i g_i(\boldsymbol{x})^2 & \text{if } \alpha_i + \omega_i g_i(\boldsymbol{x}) \geq 0 \\ \frac{-\alpha_i^2}{2\omega_i} & \text{if } \alpha_i + \omega_i g_i(\boldsymbol{x}) < 0. \end{cases}$$

Using either formulation of the function $L_{\boldsymbol{\omega}}$, an important feature of the method of multipliers is that it allows placing finite limits on the values of the Lagrange multipliers $\boldsymbol{\alpha}$ and penalty terms $\boldsymbol{\omega}$. This is in contrast with penalty methods that may require increasing the penalty parameter arbitrarily in order to achieve convergence. By alternating between finding a minimum $\boldsymbol{x}(\boldsymbol{\alpha})$ and updating multipliers $\boldsymbol{\alpha}$, both the optimal point $\boldsymbol{x}^*$ of the ECP (or ICP) will be approached as well as the optimal Lagrange multipliers $\boldsymbol{\alpha}^*$, and these are the optimal Karush-Kuhn-Tucker (KKT) pair $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*)$. A more detailed description of the method of multipliers in the context of other Lagrangian optimization is given in Section 3.1.

Genetic or evolutionary algorithms based on the method of multipliers or that rely on the augmented Lagrangian function typically implement the alternating behaviour by employing an inner and outer loop. The outer loop takes as input a point $\boldsymbol{x}^{(k)}$ representing the solution $\boldsymbol{x}(\boldsymbol{\alpha}^{(k)})$ and uses this to calculate updates to the parameters resulting in $\boldsymbol{\alpha}^{(k+1)}$ (and possibly the penalty term $\boldsymbol{\omega}^{(k+1)}$) before executing the inner loop. The inner loop takes as input these Lagrangian parameters as fixed and uses them to find the local optimum $\boldsymbol{x}(\boldsymbol{\alpha}^{(k+1)})$, which is returned again to the outer loop

to be used as a starting point in the next iteration.

This inner/outer loop format is used by Costa et al. [34] along with a hybrid genetic algorithm and pattern search method (HGPSAL) to optimize the augmented Lagrangian. Their inner loop consists of first running a genetic algorithm to generate a population of candidate solutions, then applying a Hooke and Jeeves pattern search to refine the candidates locally. The scalar penalty parameter ω is either held steady or increased by a constant factor. Safeguards are used in the outer loop to maintain boundedness on the Lagrange multipliers as well as updating stopping criteria for the inner loop.

In [35], Srivastava and Deb expand on previous work [119] to implement a genetic algorithm (GAAL) using tournament binary selection, simulated binary crossover, and adaptive polynomial mutation for solving the inner optimization loop for an augmented Lagrangian. Each candidate solution from the genetic algorithm is further improved with a "classical algorithm" (the authors use `fmincon` from Matlab) for local optimization. The outer loop iterations then rely on user-supplied parameters to evaluate the inner loop solution and subseqneutly determine whether to update either the Lagrange multipliers or the penalty coefficient.

In [80], Long et al. use a modified differential evolution approach (MAL-DE) for optimizing the inner loop that combines multiple trial vector generation strategies and adaptively selects the best within each iteration. The penalty term ω is either held steady or increased by a constant factor, subject to several criteria on the decreasing constraint violation. Termination criteria are also evaluated based on the decrease in constraint violation.

Multiple authors have also implemented the augmented Lagrangian approach for several variations of particle swarm optimization: Rocha et al. [97] use a fish swarm method, Mahdavi and Shiri [81] use a continuous ant colony method, Wen et al. use both grey wolf optimization [78] as well as an artificial bee colony algorithm [79], Bahreininejad [23] applies a water cycle algorithm, and Balande and Shrimankar [24] use a firefly algorithm.

By relying on the inner/outer loop model, each of these augmented Lagrangian methods will necessarily spend a significant portion of their function evaluations in progressing to intermediate, non-optimal solutions. Each run of an inner loop converges to a solution for the given parameters, but the parameters themselves need to also converge in the outer loop before this will coincide with the optimum $\boldsymbol{x}^*$ of the fitness function. Additionally, both of the genetic algorithm approaches rely heavily on separate optimizers like `fmincon` to ultimately achieve convergence on each subproblem formed by the augmented Lagrangian in the inner loop.

2.5 Surrogate methods

Surrogate model algorithms attempt to reduce expensive function evaluations by maintaining an internal model that can be queried instead. A notable example is the COBRA algorithm described by Regis [95] that matches radial basis functions with the search space in order to apply numerical constrained optimizers like `fmincon`. In order to reduce the need for parameter tuning, Bagheri et al. [22, 21] propose SACOBRA as a refinement, and experimentally demonstrate convergence on most of the CEC 2006 benchmark problems [77] while using fewer than 500 function evaluations. Given their strong performance in the presence of expensive objective or constraint function evaluations, surrogate models have also been used in various contexts by evolutionary algorithms.

Regis [96] proposes CONOPUS as a particle swarm method, and uses an implementation with radial basis function surrogate models of the objective and constraint functions to compare performance on several real-world as well as selected CEC benchmark problems.

Wang et al. [127] introduce GLoSADE which relies on surrogate-assisted differential evolution to globally locate regions of interest, then applies a gradient descent interior point method to a local surrogate model in order to refine solutions.

The MPMLS method of Li and Zhang [76] applies multiobjective optimization principles to solve multiple penalty problems simultaneously using differential evolution, and performance is evaluated on selected problems from CEC benchmarks as well as

an airfoil design problem.

Surrogate methods are combined with an augmented Lagrangian approach by Dufossé and Hansen [38]. The MM-AL-CMA-ES uses a linear model to internally represent the constraints, and applies CMA-ES to the unconstrained function formed by an augmented Lagrangian using the modeled constraints. Lagrangian parameters are updated in every iteration, analogous to the AL-ES [14] and as described in Section 4.2 of this thesis. Good performance is observed across the same selection of problems as used to evaluate the $(1 + 1)$-aCMA-ES [6].

2.6 Other notable approaches to constraint handling

Several other notable approaches exist for constrained optimization with evolutionary algorithms that either offer significant variations of penalty, Lagrangian, and surrogate methods, or else avoid them entirely. Given the difficulty in setting user-defined parameters for penalty-style methods [84], a common theme is to find alternate ways of balancing progress on feasible individuals against progress on infeasible individuals.

The stochastic ranking method of Runarsson and Yao [104] is based on the idea that penalty parameters are intrinsically difficult to set correctly. They argue that the ultimate goal of any penalty-based method is a ranking of individuals that does not give undue preference to either the objective function value or the constraint violation. Instead of trying to achieve this through a derived penalty, their ranking of individuals is directly manipulated by setting a fixed probability for preferring the objective function value over constraint violation. The given probability must be set by the user, something that is done implicitly in a penalty method. Stochastic ranking has been applied to several optimization strategies, including differential evolution (DE) by Zhang et al. [132] and a multimembered evolution strategy [85] by Mezura-Montes and Coello Coello.

Tahk and Sun [121] employ a modified augmented Lagrangian approach by maintaining two separate populations and co-evolving a candidate solution in the search

space alongside the Lagrangian parameters. Simulated annealing is used to discourage premature convergence near the constraint boundaries. Evolution strategies are used to evolve both populations, which are framed as solving a zero-sum game where the worst individual in the opposing population is used to determine the fitness of each offspring.

The ASCHEA method of Hamida and Schoenauer [50] uses an adaptive penalty that is specifically updated to maintain a user-defined proportion of feasible individuals in the population of each generation. The penalty is modified by multiplying by a user-defined constant, with larger penalties resulting from too few feasible individuals. A special selection operator is used to try to maintain a certain user-defined proportion of feasible individuals, and a special combination operator explicitly encourages exploration near the border when the proportion of infeasible individuals is within a specified range.

Tessema and Yen [124] use an adaptive two-penalty approach that normalizes both the objective and constraint function values for an individual, then attempts to explicitly balance favouring feasible over infeasible offspring. The proportion of feasible individuals is used to adapt the amount of additional penalties imposed on infeasible candidates.

The ϵ-constrained differential evolution (ϵDE) aproach of Takahama and Sakai [122, 123] maintains a bounded region within distance ϵ of each constraint and compares candidate solutions within this band exclusively on their objective function values. The value of ϵ is decreased according to a fixed schedule so that the solutions are eventually driven to satisfy the constraints exactly. Gradient approximations are additionally used to generate offspring, and feasible elitism ensures that the best-so-far feasible individuals are preserved. Xu et al. [130, 131] propose the HECO-DE approach which applies a similar ϵ-constrained approach to multi-objective optimization with differential evolution, where the constraints and objective functions are treated as separate functions to be jointly optimized.

The CORCO method is proposed by Wang et al. [126] which attempts to measure the correlation between objective and constraint functions. A pre-processing step is used

to generate separate populations based on improved objective or improved constraint violation scores, and these in turn generate a scalar measure of correlation that is used to guide the subsequent evolution of a population using differential evolution.

2.7 Constrained optimization with evolution strategies

Constraint handling specifically for evolution strategies has thus far followed several of its own lines of inquiry. One approach is to directly modify the population rankings in each iteration of an ES to balance the preference of feasible over infeasible candidate solutions. Runarsson and Yao [104] use an evolution strategy for defining their stochastic ranking process, which is adapted for a multimembered evolution strategy by Mezura-Montes and Coello Coello [85]. The authors of both works report experimental results on a subset of the 2006 CEC benchmark using hundreds of thousands of function evaluations.

The Adaptive Ranking Constraint Handling (ARCH) method is introduced by Sakamoto and Akimoto [105, 106] which uses CMA and adaptively updates an additional ranking coefficient based on the Mahalanobis distance between infeasible offspring and their projection on the constraint boundary, within the context of the underlying covariance matrix. This ranking coefficient in turn determines the total ranking of candidate solutions. The ARCH method explicitly assumes that constraints are a priori according to the taxonomy of Le Digabel and Wild [75] and inexpensive to calculate, and aims to preserve certain invariance properties that allow taking full advantage of covariance matrix adaptation.

Constraint-handling methods are adapted from differential evolution for use with evolution strategies as with ϵMAg-ES by Hellwig and Beyer [60]. Their MA-ES variant implements a reduced variant of CMA-ES alongside ϵ-level comparisons and gradient-based repairs from Takahama and Sakai [123]. The authors note that within each iteration that uses the repair operation, extra function evaluations are consumed making the action more expensive. The ϵMAg-ES method was retroactively ranked third in 2019 [111] among all submissions to the CEC 2017 problem set, using the prescribed $2n \cdot 10^4$ budget of function evaluations for problems of dimension n.

Active CMA methods re-purpose active updates from unconstrained optimization [68, 5], where a separate evolution path is maintained to track the worst offspring and the covariance matrix is then adapted to avoid long steps in these directions. This is used by Arnold and Hansen with the (1+1)-aCMA-ES [6] by shifting the variances of the covariance matrix away from recently violated boundaries, specifically to avoid sampling problems caused by small constraint angles. This is extended to the multimembered case by Chocat et al. [31] for application to a rocket design problem in the presence of noise, and by Krause and Glasmachers [74] who integrate features of natural evolution strategies to propose xCMA-ES and evaluate its performance on several sphere problems with varying bound constraints. Similar active updates are combined with both a multimembered MA-ES and CMA-ES by Spettel and Beyer [115], and experimental results for each compared across a variety of problems. The resulting CA-MA-ES algorithm's performance seems promising, but focuses on a benchmark that is still under development and not yet widely used in the literature.

Under the assumption of purely linear constraints, Spettel et al. [117] propose the lcCMSA-ES that uses a pre-processing step to project onto the manifold of intersecting feasible regions. Feasibility is then maintained by a combination of biased mutations for encouraging new offspring within the feasible manifold together with a repair mechanism. The algorithm is shown to perform well across a selection of linearized problems from the COCO bbob-constrained framework, as well as on the Rotated Klee-Minty problem in various dimensions.

Active-set evolution strategies project candidate solutions onto the feasible subspace where constraints are satisfied as equality constraints, then update the step size according to the reduced subspace dimension. Lagrange multipliers are used implicitly to determine whether a constraint is active. So long as there is a robust method for adding and dropping constraints from the active set, this allows the step size to adapt appropriately and avoid issues of stagnation. Building on initial work by Arnold [11, 12], Spettel et al. [112] implement a $(1+1)$-ES with an updated approach for suspending constraints from the active set. Experimental results show good performance on certain linearly constrained sphere problems, as well as problems from

the CEC 2006 benchmark. Throughout, the active set approaches assume that constraints are given a priori according to the taxonomy of Le Digabel and Wild [75] and inexpensive to calculate.

2.8 Summary

A wide variety of constraint handling methods exist for stochastic optimization. In order to make reasonable comparisons between algorithms, the QRAK taxonomy allows classifying approaches based on assumptions made about the constraints. For instance, approaches like the active set ES [112] and ARCH [106] methods perform very well for QUAK problems where constraints can be evaluated with negligible cost.

Other approaches for evolution strategies including the aCMA-ES [6] and MM-AL-ES [38] have established a set of QR*K problems, many of them taken from the CEC 2006 benchmark problem set, that serve as a useful starting point for comparing the performance of any novel ES approach. Convergence on these problems is observed to occur within several thousand function evaluations. By comparison, algorithms evaluated on the CEC 2006 problem set [77] typically report solutions with tens or even hundreds of thousands of function evaluations.

Augmented Lagrangian approaches are a popular mode for handling QR*K optimization problems, but the usual design relies on an inner/outer loop model that consumes function evaluations while converging to values that quickly become obsolete. Implementing an approach without the inner/outer loop model may provide the benefits of the augmented Lagrangian model without its largest drawback. Integrating a Lagrangian approach with evolution strategies may additionally take advantage of their beneficial convergence and invariance properties [19].

Chapter 3

Augmented and exact Lagrangian methods

This chapter presents the augmented and exact Lagrangian methods for solving constrained optimization problems, framed in terms of approaches from numerical optimization. Each of Sections 3.1 and 3.2 respectively provide justifications for the evolution strategy algorithms presented later in Sections 4.1 and 4.3. A more comprehensive overview of some of the theory behind optimization with Lagrangian functions is given in Appendix B.

Section 3.1 describes the method of multipliers, also referred to as the augmented Lagrangian method, which aims to both define an unconstrained function with an optimum shared by the constrained problem and give update rules for the Lagrange multipliers that will lead to an optimal KKT pair. Section 3.2 outlines Fletcher's exact penalty method for both equality and inequality constraints, which also defines an unconstrained function with a desired optimum but replaces multiplier updates with a continuous approximation that can be understood as implicitly solving local quadratic approximations of the constrained problem. Connections between the two approaches are given in Section 4.4 in the context of their implementations for evolution strategies.

Note that to avoid confusion with the (μ, λ) notation commonly used for evolution strategies, Lagrange multipliers here are generally referred to as $\boldsymbol{\alpha}$ rather than the traditional $\boldsymbol{\lambda}$ used both in the appendix and in the literature.

3.1 Method of multipliers

The method of multipliers is a method proposed independently by Hestenes [64] and Powell [93] for solving ECPs by combining penalty and Lagrangian functions

with a sequence of Lagrange multiplier approximations $\{\boldsymbol{\alpha}^{(k)}\}$ that converges to $\boldsymbol{\alpha}^*$. Central to the method is the definition of an *augmented* Lagrangian function as well as recommended updates for the parameters of that function.

Powell forms an augmented Lagrangian by beginning with the usual quadratic penalty method that attempts to solve

$$Q_{\boldsymbol{\omega}}(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}) \tag{3.1}$$

by gradually increasing $\omega_i \to \infty$ which are the elements of $\boldsymbol{\omega}$ forming the nonzero entries of the diagonal matrix $\boldsymbol{\Omega}$. However, in order to avoid ill-conditioning and other problems with arbitrarily large ω_i, the penalty term's origin is shifted and a new variable $\boldsymbol{\theta}$ introduced to instead solve

$$L^{\mathrm{Po}}_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\theta}) = f(\boldsymbol{x}) + \frac{1}{2} (g(\boldsymbol{x}) - \boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\Omega} (g(\boldsymbol{x}) - \boldsymbol{\theta}). \tag{3.2}$$

These new variables are updated as

$$\theta_i^{(k+1)} = \theta_i^{(k)} + g_i(\boldsymbol{x}^{(k)}(\boldsymbol{\theta})) \tag{3.3}$$

where $\boldsymbol{x}^{(k)}(\boldsymbol{\theta})$ represents the local solution for parameter $\boldsymbol{\theta}$, and the values ω_i are intended to be kept relatively constant. Formulated this way, Powell's augmented Lagrangian has the large advantage that $\omega_i \to \infty$ is no longer a requirement for finding a solution, so long as both $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ are updated correctly in order to shift the constraints so that the minimum for $L^{\mathrm{Po}}_{\boldsymbol{\omega}}(\boldsymbol{x})$ is also $\boldsymbol{x}^*$. By expanding the expression in Eq. (3.2) for Powell's augmented Lagrangian and dropping a constant term, we find

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\theta}) = f(\boldsymbol{x}) + (\boldsymbol{\Omega} \cdot \boldsymbol{\theta})^{\mathrm{T}} g(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}) \tag{3.4}$$

and after defining $\boldsymbol{\Omega} \cdot \boldsymbol{\theta} = \boldsymbol{\alpha}$ we have an equivalent formulation of the augmented Lagrangian as proposed by Hestenes

$$L^{\mathrm{He}}_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}). \tag{3.5}$$

This is the form of the augmented Lagrangian we will most often use, and so refer to

it simply as $L_{\boldsymbol{\omega}}$. The values of $\boldsymbol{\alpha}$ are here updated as

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \omega_i \cdot g_i(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)}))$$

or equivalently

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \boldsymbol{\Omega} \cdot g(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)})). \tag{3.6}$$

This encodes a shift of the quadratic penalty term (compare to Eq. (3.3)) and can be proven under mild assumptions to give a sequence of vectors that converges to the optimal Lagrange multipliers $\boldsymbol{\alpha}^*$ as $\boldsymbol{x} \to \boldsymbol{x}^*$. An implementation of this approach to the method of multipliers in given in pseudo-code by Algorithm 3.1, using the usual inner/outer loop model. Line 2 encapsulates the inner loop portion, where an iterative algorithm is used to minimize the unconstrained function given by $L_{\boldsymbol{\omega}}$. Exactly how the penalty coefficients $\boldsymbol{\omega}$ should be updated in Line 4, and whether they should be increased in every iteration, is a decision that can vary by implementation. We omit these details here in the interest of simplicity, but recommended bounds are offered analytically by both Fletcher [43] and Bertsekas [27].

Algorithm 3.1 Method of multipliers with inner/outer loop

Require: Initialize $\boldsymbol{x}^{(0)},\ \boldsymbol{\alpha}^{(0)},\ \boldsymbol{\omega}^{(0)},\ k = 0,\ \chi \geq 1$

1: **while** $\boldsymbol{x}^{(k)} \neq \boldsymbol{x}^*$ **do** ▷ Outer loop: updates $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$
2: $\quad \boldsymbol{x}^{(k+1)} \leftarrow \min_{\boldsymbol{x}} \left[L_{\boldsymbol{\omega}^{(k)}}(\boldsymbol{x}, \boldsymbol{\alpha}^{(k)}) \right]$ ▷ Inner loop: minimizes $L_{\boldsymbol{\omega}}$
3: $\quad \boldsymbol{\alpha}^{(k+1)} \leftarrow \boldsymbol{\alpha}^{(k)} + {\boldsymbol{\omega}^{(k)}}^{\mathrm{T}} g(\boldsymbol{x}^{(k+1)})$ ▷ Eq. (3.6)
4: $\quad \boldsymbol{\omega}^{(k+1)} \leftarrow \boldsymbol{\omega}^{(k)} \cdot \chi$ ▷ Optional: increase, if needed
5: $\quad k \leftarrow k + 1$
6: **end while**

3.1.1 Justification of the update rule

One justification behind the specific update term for the Lagrange multipliers in Eq. (3.6) comes from observing that a solution to the augmented Lagrangian in Eq. (3.5) may not always lead to a solution of the underlying constrained problem, particularly if we have not used the correct Lagrange multipliers. This is often

the case in practice, such as when the Lagrange multipliers $\boldsymbol{\alpha}^{(k)}$ are iteratively approximated. This is obviously a relevant concern for an adaptation of augmented Lagrangian approaches for stochastic methods, and so the justification is summarized below. The core idea of the justification is that inaccurate solutions to $L_{\boldsymbol{\omega}}$ allow expressing the Lagrange multiplier approximations in terms of gradients with respect to local changes in the constraint violations, and setting these gradients to 0 to find the stationary point where the constraints are satisfied leads to Eq. (3.6). The discussion here is given in terms of equality constraints (ECP) for simplicity of presentation, but extends similarly to inequalities.

To begin, note that if our current approximation $\boldsymbol{\alpha}^{(k)} \neq \boldsymbol{\alpha}^*$, then the local minimum of $L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}^{(k)})$ need not correspond to the constrained solution $\boldsymbol{x}^*$. For brevity, let $\boldsymbol{x}^{(k)} = \boldsymbol{x}(\boldsymbol{\alpha}^{(k)})$ represent the solution minimizing $L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}^{(k)})$. Then recalling the necessary conditions of Theorem B.6, and assuming $\boldsymbol{\omega}$ is set appropriately in order to guarantee locally positive curvature for $L_{\boldsymbol{\omega}}$, we can identify the situation where $\boldsymbol{x}^{(k)} \neq \boldsymbol{x}^*$ by the existence of constraint values $g(\boldsymbol{x}^{(k)}) \neq 0$. With this in mind, we return to the augmented Lagrangian defined using $\boldsymbol{\alpha}^{(k)}$, and since $\boldsymbol{x}^{(k)}$ is a local minimizer, this implies we have also found a stationary point satisfying

$$\begin{aligned}
\nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)}) &= \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}) + (\boldsymbol{\alpha}^{(k)})^{\mathrm{T}} \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}) + g(\boldsymbol{x}^{(k)})^{\mathrm{T}} \boldsymbol{\Omega} \cdot \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}) \\
&= \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}) + \left(\boldsymbol{\alpha}^{(k)} + \boldsymbol{\Omega} \cdot g(\boldsymbol{x}^{(k)})\right)^{\mathrm{T}} \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}) \\
&= 0.
\end{aligned}$$

Again for brevity, let $\boldsymbol{u} = g(\boldsymbol{x}^{(k)}) \in \mathbb{R}^m$ be the vector of constraint violations for the found point $\boldsymbol{x}^{(k)}$. Then the gradient above can also be re-written as

$$\nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)}) = \nabla_{\boldsymbol{x}} L_0(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)} + \boldsymbol{\Omega} \cdot \boldsymbol{u}). \tag{3.7}$$

This is the gradient of the ordinary Lagrangian L_0 evaluated at the points $\boldsymbol{x}^{(k)}$ and $\boldsymbol{\alpha}^{(k)} + \boldsymbol{\Omega} \cdot g(\boldsymbol{x}^{(k)})$, and it is also equal to zero. Note that if $\boldsymbol{u} = 0$ then as mentioned, we are already at the solution $\boldsymbol{x}^*$ of the constrained optimization problem and $\boldsymbol{x}^{(k)} = \boldsymbol{x}^*$. If however $u_i \neq 0$, then the current point $\boldsymbol{x}^{(k)}$ minimizes the ordinary Lagrangian with gradient given in Eq. (3.7), and so is also a local minimum of a related but *distinct*

ECP, one given by

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) - u_i = 0. \end{aligned}$$

This is the equality constrained problem asking for the point $\boldsymbol{x}$ that minimizes $f(\boldsymbol{x})$ subject to constraints g_i being violated by exactly $u_i = g_i(\boldsymbol{x}^{(k)})$, the amount of violation at our current solution. If we treat $\boldsymbol{u}$ as a variable, then we find an entire family of related ECPs, each asking for the point $\boldsymbol{x}$ minimizing $f(\boldsymbol{x})$ subject to constraint violations being equal to $\boldsymbol{u}$. Clearly, we are interested in the behaviour as $\boldsymbol{u} \to 0$. Let $\boldsymbol{x}(\boldsymbol{u})$ be the minimizing point $\boldsymbol{x}$ for the variable $\boldsymbol{u}$, then the solutions to these ECPs can be expressed as solutions to the ordinary Lagrangian

$$L_0(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\alpha}) = f(\boldsymbol{x}(\boldsymbol{u})) + \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{u}$$

with corresponding gradient

$$\nabla_{\boldsymbol{u}} L_0(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\alpha}) = \nabla_{\boldsymbol{u}} f(\boldsymbol{x}(\boldsymbol{u})) + \boldsymbol{\alpha}.$$

Setting this equal to zero we arrive at

$$-\nabla_{\boldsymbol{u}} f(\boldsymbol{x}(\boldsymbol{u})) = \boldsymbol{\alpha}. \tag{3.8}$$

Thus, the Lagrange multipliers $\boldsymbol{\alpha}$ correspond with the negative gradient of f taken at the local minimizer $\boldsymbol{x}(\boldsymbol{u})$ with respect to the variable $\boldsymbol{u}$ of constraint violation.

Returning to consider this in the context of our augmented Lagrangian, we can re-write its expression as

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\alpha}^{(k)}) = f(\boldsymbol{x}^{(k)}) + (\boldsymbol{\alpha}^{(k)})^{\mathrm{T}}\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{\Omega}\boldsymbol{u}$$

where $\boldsymbol{x}^{(k)} = \boldsymbol{x}(\boldsymbol{u})$. Minimizing this expression with respect to $\boldsymbol{u}$ requires a stationary

point, which can be found using

$$\nabla_{\boldsymbol{u}} L_{\boldsymbol{\omega}} = \nabla_{\boldsymbol{u}} \Big(f(\boldsymbol{x}^{(k)}) + (\boldsymbol{\alpha}^{(k)})^{\mathrm{T}} \boldsymbol{u} + \frac{1}{2} \boldsymbol{u}^{\mathrm{T}} \boldsymbol{\Omega} \boldsymbol{u} \Big)$$

which after equating to zero and re-arranging gives

$$-\nabla_{\boldsymbol{u}} f(\boldsymbol{x}^{(k)}) = \boldsymbol{\alpha}^{(k)} + \boldsymbol{\Omega} \boldsymbol{u}.$$

This gives the negative gradient of f taken at the local minimizer $\boldsymbol{x}(\boldsymbol{u})$ with respect to $\boldsymbol{u}$. Since this should correspond to the Lagrange multipliers by Eq. (3.8), we therefore have an appropriate value for setting $\boldsymbol{\alpha}^{(k+1)}$. This is the same update rule as in Eq. (3.6) and serves to justify its use.

This idea behind justifying the multiplier step is treated more rigorously by Bertsekas [27] who defines the *primal functional* p in terms of constraint violation $\boldsymbol{u}$ as

$$p(\boldsymbol{u}) = \min_{g(\boldsymbol{x}) = \boldsymbol{u}} f(\boldsymbol{x})$$

and observes in part that

$$\nabla_{\boldsymbol{u}} p(\boldsymbol{u}) = -\boldsymbol{\alpha}(\boldsymbol{u}),$$

and in particular

$$\nabla_{\boldsymbol{u}} p(\boldsymbol{0}) = -\boldsymbol{\alpha}^{*}.$$

Since the primal functional returns the minimal value of $f(\boldsymbol{x})$ across all points where $g(\boldsymbol{x}) = \boldsymbol{u}$, this means the Lagrange multipliers near the optimum can be interpreted as rates of change of the minimum of f with respect to changes in constraint violations $\boldsymbol{u}$.

3.1.2 Extensions and inequalities

Along with having a convenient multiplier update rule, it can be shown [43, 27] that the augmented Lagrangian given in Eq. (3.5) is also positive definite in a region of the

optimum $\boldsymbol{x}^*$ and thus satisfies Eq. (B.18) for sufficiently large choices of ω_i. Together, Eqs. (3.5) and (3.6) form what is generally understood as the method of multipliers for equality constraints, but there are alternative ways of viewing this formulation that give beneficial insight. Nocedal and Wright [128] describe the augmented Lagrangian as a suitable function for correcting consistent errors or perturbations in the quadratic penalty approach. They show that approximate solutions for minimizing $Q_{\boldsymbol{\omega}}(\boldsymbol{x})$ will tend to give constraint violations $g_i(\boldsymbol{x}) \approx \frac{\alpha_i^*}{\omega_i}$, and therefore include this term as an estimator for the optimal Lagrange multiplier within each iteration. Bertsekas [27, 28] meanwhile treats the quadratic penalty function in Eq. (3.1) fully as an objective function, and constructs the related Lagrangian function as

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = \left(f(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}) \right) + \boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}).$$

The augmented Lagrangian is then no longer bound only to a quadratic penalty function; other penalty functions would lead just as easily to a variation thereof.

The method of multipliers was originally extended to solving ICP problems by Rockafellar [99, 98, 101] and results in expressing the Lagrangian as

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \Psi(\boldsymbol{x}) \tag{3.9}$$

where

$$\Psi(\boldsymbol{x}) = \sum_i \begin{cases} \alpha_i g_i(\boldsymbol{x}) + \frac{1}{2}\omega_i g_i(\boldsymbol{x})^2 & \text{if } \alpha_i + \omega_i g_i(\boldsymbol{x}) \geq 0 \\ \frac{-\alpha_i^2}{2\omega_i} & \text{if } \alpha_i + \omega_i g_i(\boldsymbol{x}) < 0. \end{cases}$$

Roughly speaking, this defines a function continuous at the constraint boundaries that nonetheless distinguishes between constraints that are consequential for the Lagrangian and those that are not.

Rockafellar arrives at this expression by converting the inequalities using introduced slack variables $z_i \geq 0$ so that the augmented Lagrangian can be written as

$$\begin{aligned} L_{\boldsymbol{\omega}}^{\mathrm{Ro}}(\boldsymbol{x}, \boldsymbol{\alpha}) &= f(\boldsymbol{x}) + \sum_{i \in \mathcal{I}} \left(g_i(\boldsymbol{x}) + z_i \right) \alpha_i + \frac{1}{2} \sum_{i \in \mathcal{I}} \omega_i \left(g_i(\boldsymbol{x}) + z_i \right)^2 \\ &= f(\boldsymbol{x}) + \boldsymbol{\alpha}^{\mathrm{T}} (g(\boldsymbol{x}) + \boldsymbol{z}) + \frac{1}{2} (g(\boldsymbol{x}) + \boldsymbol{z})^{\mathrm{T}} \boldsymbol{\Omega} (g(\boldsymbol{x}) + \boldsymbol{z}) \end{aligned} \tag{3.10}$$

for inequality constraints indexed by $\mathcal{I}$. The expression in the second line follows from using vector notation and defining the vector of slack variables

$$\boldsymbol{z} = \left(-\boldsymbol{\Omega}^{-1}\boldsymbol{\alpha} - g(\boldsymbol{x})\right)_+$$

where the plus operator $\cdot_+ = \max(0, \cdot)$ restricts element-wise to non-negative values. The value of $L_{\boldsymbol{\omega}}^{\mathrm{Ro}}$ can be explicitly minimized with respect to the z_i by taking partial derivatives and writing

$$\frac{\partial L}{\partial z_i} = \omega_i g_i(\boldsymbol{x}) + \omega_i z_i + \alpha_i = 0$$

and then re-arranging gives

$$z_i = \frac{-\alpha_i}{\omega_i} - g_i(\boldsymbol{x}). \tag{3.11}$$

Now since $z_i \in \mathbb{R}_+$, each value must be either positive or 0. If we allow $\mathcal{P}$ and $\mathcal{Z}$ to respectively indicate the complementary index sets for positive and zero values of z_i, we can write the augmented Lagrangian L_ω with respect to these index sets. Consider first the situation where only one index set is non-empty at a time. Taking $\mathcal{Z}$ to be non-empty resolves to the Lagrangian

$$L_{\boldsymbol{\omega}}^{\mathrm{Ro}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_{i \in \mathcal{Z}} \left(\alpha_i g_i(\boldsymbol{x}) + \frac{\omega_i}{2} g_i(\boldsymbol{x})^2\right)$$

which is equivalent to the expression in Eq. (3.10), while non-empty $\mathcal{P}$ gives

$$\begin{aligned} L_{\boldsymbol{\omega}}^{\mathrm{Ro}}(\boldsymbol{x}, \boldsymbol{\alpha}) &= f(\boldsymbol{x}) + \frac{1}{2} \sum_{i \in \mathcal{P}} \omega_i \left(\frac{\alpha_i^2}{\omega_i^2} - \frac{2\alpha_i^2}{\omega_i^2}\right) \\ &= f(\boldsymbol{x}) - \sum_{i \in \mathcal{P}} \frac{\alpha_i^2}{2\omega_i} \end{aligned}$$

after expansion and substitution of the value given by Eq. (3.11). Noting that the indices $i \in \mathcal{Z}$ imply $\omega_i g_i(\boldsymbol{x}) \geq -\alpha_i$ and include any constraints satisfied as equalities, the two alternative expressions for the augmented Lagrangian can be joined together to encompass both of $\mathcal{P}$ and $\mathcal{Z}$ and simply expressed as in Eqs. (3.9) and (3.10). The condition of the top row of Eq. (3.10) is indexed by $\mathcal{Z}$ and includes both *weakly active* constraints where $g_i(\boldsymbol{x}) = 0$ and *active* constraints where additionally $\alpha_i > 0$. The

condition of the bottom row is indexed by $\mathcal{P}$ and accounts for inactive constraints. The statement of Eq. (3.9) is also equivalent to

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_i \frac{1}{2\omega_i}\Big([\alpha_i + \omega_i g_i(\boldsymbol{x})]_+^2 - \alpha_i^2\Big) \tag{3.12}$$

where the plus operator $\cdot_+$ again maps to non-negative values, or using matrix notation evaluated only on specified constraint sets as

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \left(\boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}) + \frac{1}{2} g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x})\right|_{\mathcal{Z}} - \left(\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{\alpha}\right|_{\mathcal{P}}.$$

The same extension to ICPs is derived by Nocedal and Wright [128] by solving

$$\max_{\alpha_i \geq 0} L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_i \alpha_i g_i(\boldsymbol{x}) + \frac{1}{2\omega_i} \sum_i \left(\alpha_i - \alpha_i^{(k)}\right)^2 \tag{3.13}$$

explicitly in terms of variables α_i. The result is the same as in Eq. (3.9), but frames the augmented Lagrangian as quadratically penalizing new multipliers α_i that are more distant from the current multiplier $\alpha_i^{(k)}$ in a given iteration k. This is similar in principle to solving a dual problem in order to determine good updates for the multipliers, as outlined in Section B.5.

3.2 Fletcher's exact penalty method

Penalty methods typically require coefficients that must be made arbitrarily large in order to guarantee convergence. The method of multipliers as described in Section 3.1 provides an alternative formulation where an optimum can be reached for finite penalty coefficients, and Fletcher [39, 40] together with Lill [45] additionally propose several so-called penalty functions with the desirable feature that any penalty coefficients above a fixed limit would be sufficient for convergence. These are examples of *exact* penalty methods, so named in order to contrast them with existing sequential methods; while the latter attempt to approach the constrained optimum by solving a sequence of optimization problems with an increasing penalty coefficient and the sequence of solutions converging to $\boldsymbol{x}^*$, the former aim to define a single

problem with its unconstrained solution corresponding exactly to $\boldsymbol{x}^*$. In sequential methods such as the method of multipliers, both an inner and an outer loop control the algorithm, with the outer loop updating parameters like the penalty coefficient only after the inner loop has found an appropriate intermediate solution under the existing parameters. In the exact method, only a single optimization problem is iterated on, defined completely in terms of location $\boldsymbol{x}$ in the search space. As will be seen in Chapter 4, existing implementations based on the AL-ES [14] straddle the difference between the exact and sequential approaches, while other augmented Lagrangian methods such as from Deb and Srivastava [35] rely entirely on a sequential approach.

The exact penalty functions will be seen to be directly connected with Lagrangian functions, as both of the functions' values and their gradients are equal at the optimum and their formulations are similar. The continuous approximation of Lagrange multipliers provided by the exact Lagrangian functions is also used to extend the method to handle inequality constraints [41].

3.2.1 Exact Lagrangian for equality constraints

Recall that the ECP asks for a solution $\boldsymbol{x}$ to the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) = 0. \end{aligned}$$

With objective function $f(\boldsymbol{x})$ and constraint function $g(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m$ both continuously differentiable, Fletcher's exact penalty function is defined as

$$\phi(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{J}^{+} \nabla f + \omega \cdot g(\boldsymbol{x})^{\mathrm{T}} (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J})^{-1} g(\boldsymbol{x}) \tag{3.14}$$

with scalar $\omega > 0$, full rank $n \times m$ Jacobian matrix $\boldsymbol{J}$ of g, and other relevant terms including derivatives evaluated with respect to $\boldsymbol{x}$. So long as ω is chosen sufficiently large, $\phi(\boldsymbol{x})$ will be positive definite in a neighbourhood $\mathcal{N}_r(\boldsymbol{x}^*)$ of the optimum because of the associated "augmenting" penalty term, ensuring it is a local

minimum.

Relation to Lagrangian functions

In order to see how the exact penalty function is related to Lagrangian functions, first define the ordinary Lagrangian ψ in the usual way as

$$\begin{aligned}\psi(\boldsymbol{x}) &= f(\boldsymbol{x}) + g(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\alpha} \\ &= f(\boldsymbol{x}) - g(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{J}^{+} \cdot \nabla f. \end{aligned} \tag{3.15}$$

where the derivatives are taken with respect to $\boldsymbol{x}$. The multipliers $\boldsymbol{\alpha}$ here are derived from the KKT first-order necessary condition (see Theorem B.6) that $\nabla_{\boldsymbol{x}}\psi = 0$, which allows evaluating and re-arranging the condition to give

$$\begin{aligned}\nabla_{\boldsymbol{x}} f + \nabla_{\boldsymbol{x}} g^{\mathrm{T}} \cdot \boldsymbol{\alpha} &= 0 \\ \boldsymbol{J} \cdot \boldsymbol{\alpha} &= -\nabla_{\boldsymbol{x}} f \\ \boldsymbol{\alpha} &= -\boldsymbol{J}^{+} \cdot \nabla_{\boldsymbol{x}} f \end{aligned} \tag{3.16}$$

as a least-squares solution for the Lagrange multipliers. This is also equivalent to determining the multipliers by solving

$$\begin{aligned}\boldsymbol{J}\left(-\boldsymbol{J}^{+}\nabla_{\boldsymbol{x}} f\right) &= -\nabla_{\boldsymbol{x}} f \\ (\boldsymbol{I} - \boldsymbol{P})\nabla_{\boldsymbol{x}} f &= 0\end{aligned}$$

with respect to the projection matrix $\boldsymbol{P} = \boldsymbol{J}\boldsymbol{J}^{+}$ described in Eq. (B.4). This says that the KKT optimum must be among the points $\boldsymbol{x}$ where the gradient is zero after projection into the unconstrained subspace by $(\boldsymbol{I} - \boldsymbol{P})$.

Although the Lagrangian function $\psi(\boldsymbol{x})$ has a stationary point at the constrained optimum $\boldsymbol{x}^{*}$, this is not guaranteed to be a minimum as the curvature in the directions of the constraint normals may not be positive. However, positive curvature may be induced in these directions within $\mathcal{N}_r(\boldsymbol{x}^{*})$ by adding an extra term involving a sufficiently large positive definite matrix. To this end, the exact penalty function $\phi(\boldsymbol{x})$ is defined as $\psi(\boldsymbol{x})$ together with the term $g(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Omega}g(\boldsymbol{x})$ for some positive definite

matrix $\boldsymbol{\Omega}$. Fletcher suggests appropriate choices as including $\boldsymbol{\Omega} = \omega \boldsymbol{I}$, $\boldsymbol{\Omega} = \omega \nabla^2 f(\boldsymbol{x})$, and $\boldsymbol{\Omega} = \omega (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J})^{-1}$, all with scalar $\omega > 0$, and it is the final option used in the definition of the exact Lagrangian in Eq. (3.14).

Rather than modifying the Lagrangian, we could equivalently consider this as a modification of the Lagrange multipliers. Returning to the definition of the ordinary Lagrangian $\psi(\boldsymbol{x})$ in Eq. (3.15), let the multipliers be given in terms of $\boldsymbol{x}$ by

$$\boldsymbol{\alpha}(\boldsymbol{x}) = -\boldsymbol{J}^{+} \nabla f(\boldsymbol{x}) + \omega \cdot (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J})^{-1} g(\boldsymbol{x}) \tag{3.17}$$

then after substitution we again have the definition of the exact Lagrangian in Eq. (3.14).

A key difference arises here between the exact Lagrangian and other Lagrangian approaches: rather than taking the vector $\boldsymbol{\alpha}$ of multipliers to be a parameter alongside $\boldsymbol{x}$ that will be solved by minimizing $L(\boldsymbol{x}, \boldsymbol{\alpha})$, the exact approach is to instead define the multipliers completely in terms of $\boldsymbol{x}$ using the above approximation. It is in this sense that the Lagrangian is *exact*, as its optimum will correspond under mild assumptions to the constrained optimum without any sequential updates to external parameters.

Using Eq. (3.16) and defining $\boldsymbol{\beta} = (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J})^{-1}$, the exact Lagrangian defined in Eq. (3.14) is seen to still very closely resemble an augmented Lagrangian with a slight change to the definition of the Lagrange multipliers:

$$L_{\beta}(\boldsymbol{x}, \boldsymbol{\alpha}(\boldsymbol{x})) = f(\boldsymbol{x}) + \boldsymbol{\alpha}(\boldsymbol{x})^{\mathrm{T}} g(\boldsymbol{x}) + \omega \cdot g(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\beta} g(\boldsymbol{x}).$$

The important distinction is that $\boldsymbol{\alpha}(\boldsymbol{x})$ is not a separate parameter, as it would be for the method of multipliers, but is defined completely in terms of the position $\boldsymbol{x}$ in the search space.

Multiplier equivalence with subproblems

To help understand the way these multipliers are continuously estimated, it will first be shown that each set of approximated multipliers is in fact shared with an equality constrained subproblem involving quadratic/linear approximations for the objective/constraint functions of the ECP, respectively.

Theorem 3.1 (Fletcher [41])**.** *The Lagrange multipliers defined in Eq.* (3.17) *are identical to those solving the equality constrained subproblem*

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & Q(\boldsymbol{\delta}) = \frac{\omega}{2}\boldsymbol{\delta}^T\boldsymbol{\delta} + \boldsymbol{\delta}^T\nabla_{\boldsymbol{x}} f \\ s.t. \quad & \ell(\boldsymbol{\delta}) = \boldsymbol{J}^T\boldsymbol{\delta} + g = 0 \end{aligned} \tag{3.18}$$

consisting of a local quadratic approximation of $f(\boldsymbol{x})$ and linear approximations of the constraints g_i evaluated at $\boldsymbol{x}$.

Proof. From Eq. (3.18) we can write the associated Lagrangian function as

$$L(\boldsymbol{\delta}, \boldsymbol{\alpha}) = Q(\boldsymbol{\delta}) + \boldsymbol{\alpha}^{\mathrm{T}}\ell(\boldsymbol{\delta}).$$

The usual first-order necessary condition then requires that both $\nabla_{\boldsymbol{\delta}} L$ and $\nabla_{\boldsymbol{\alpha}} L$ equal 0 at the subproblem's optimum, and solving analytically gives the two expressions

$$\begin{aligned} \nabla_{\boldsymbol{\delta}} L &= \nabla_{\boldsymbol{\delta}} Q(\boldsymbol{\delta}) + \boldsymbol{\alpha}^{\mathrm{T}}\nabla_{\boldsymbol{\delta}}\ell(\boldsymbol{\delta}) \\ &= \omega\boldsymbol{\delta} + \nabla_{\boldsymbol{x}} f + \boldsymbol{J}\boldsymbol{\alpha}, \\ \nabla_{\boldsymbol{\alpha}} L &= \ell(\boldsymbol{\delta}) \\ &= \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\delta} + g \end{aligned}$$

which each re-arrange to

$$\begin{aligned} -\nabla_{\boldsymbol{x}} f &= \omega\boldsymbol{\delta} + \boldsymbol{J}\boldsymbol{\alpha}, \\ -g &= \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\delta}. \end{aligned}$$

Thus, the KKT pair $\boldsymbol{\delta}, \boldsymbol{\alpha}$ for the subproblem can be recovered from the linear system of block matrices

$$\begin{pmatrix} \omega \boldsymbol{I} & \boldsymbol{J} \\ \boldsymbol{J}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} -\nabla_{\boldsymbol{x}} f \\ -g \end{pmatrix}$$

$$\begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \frac{1}{\omega}(\boldsymbol{I} - \boldsymbol{P}) & \boldsymbol{J}^{\mathrm{T}+} \\ \boldsymbol{J}^{+} & -\omega(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1} \end{pmatrix} \begin{pmatrix} -\nabla_{\boldsymbol{x}} f \\ -g \end{pmatrix}. \tag{3.19}$$

The validity of the inverse matrix given in the second line above can be verified explicitly. To do so, first recall that the projection matrix defined by Eq. (B.3) gives both

$$\begin{aligned} (\boldsymbol{I} - \boldsymbol{P}) &= (\boldsymbol{I} - \boldsymbol{J}\boldsymbol{J}^{+}) \\ &= \boldsymbol{I} - \boldsymbol{J}(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}} \\ &= \boldsymbol{I} - \boldsymbol{J}^{\mathrm{T}+}\boldsymbol{J}^{\mathrm{T}} \end{aligned}$$

and

$$\boldsymbol{J}^{+} = (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}}.$$

Therefore,

$$\begin{aligned} \omega \boldsymbol{I} \cdot \frac{1}{\omega}(\boldsymbol{I} - \boldsymbol{P}) + (\boldsymbol{J}^{\mathrm{T}+})(\boldsymbol{J}^{\mathrm{T}}) &= (\boldsymbol{I} - \boldsymbol{P}) + \boldsymbol{P} \\ &= \boldsymbol{I} \end{aligned}$$

and

$$\begin{aligned} \frac{1}{\omega}(\boldsymbol{I} - \boldsymbol{P}) \cdot (\boldsymbol{J}) &= \frac{1}{\omega}\Big(\boldsymbol{J} - \boldsymbol{J}(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}} \cdot \boldsymbol{J}\Big) \\ &= 0 \end{aligned}$$

satisfy the first line of the inverse block matrix on the right-hand side of Eq. (3.19), while

$$\begin{aligned} \boldsymbol{J}^{+} \cdot \omega \boldsymbol{I} + (-\omega(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1})(\boldsymbol{J}^{\mathrm{T}}) &= \omega \boldsymbol{J}^{+} - \omega \boldsymbol{J}^{+} \\ &= 0 \end{aligned}$$

and

$$\boldsymbol{J}^{+} \cdot (\boldsymbol{J}) = \boldsymbol{I}$$

satisfy the second line.

With the matrix validated, we can expand the full expression of Eq. (3.19) and collect terms in order to find

$$\begin{aligned} \boldsymbol{\delta} &= \frac{-1}{\omega}(\boldsymbol{I} - \boldsymbol{P})\nabla_{\boldsymbol{x}} f - \boldsymbol{J}^{\mathrm{T}^{+}} g, \\ \boldsymbol{\alpha} &= -\boldsymbol{J}^{+}\nabla_{\boldsymbol{x}} f + \omega(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1} g, \end{aligned}$$

as the KKT pair for the equality subproblem defined in the theorem statement. The expression above for $\boldsymbol{\alpha}$ corresponds directly with Eq. (3.17), as required. ■

This theorem justifies an understanding of the sequence of multiplier approximations given by the function in Eq. (3.17) by means of understanding the sequence of underlying subproblems. Each subproblem also allows examining its solution analytically. Consider a shift of origin for $\boldsymbol{\delta}$-space from the subproblem so that we let $\boldsymbol{\delta} = \mathbf{0}$ correspond to the current point $\boldsymbol{x}$ in the search space for the ECP, and recall that the subproblem of Eq. (3.18) is quadratic with Hessian $\omega\boldsymbol{I}$ and the constraints ℓ_i are linear. The function Q must have a global unconstrained minimum where its derivative is zero. Let $\boldsymbol{\delta}^{*}$ indicate the constrained optimum of the subproblem and $\boldsymbol{\delta}^{(*)}$ be the unconstrained optimum of Q, so that by solving the first-order equation

$$\begin{aligned} \nabla_{\boldsymbol{\delta}} Q &= \omega\boldsymbol{\delta} + \nabla_{\boldsymbol{x}} f \\ &= 0 \end{aligned} \tag{3.20}$$

we get

$$\boldsymbol{\delta}^{(*)} = \frac{-\nabla_{\boldsymbol{x}} f}{\omega}. \tag{3.21}$$

Geometrically, this is the location $\boldsymbol{\delta}^{(*)}$ reached by moving from $\boldsymbol{\delta} = 0$ in the direction of the negative gradient of the objective function f. The solution $\boldsymbol{\delta}^*$ to the *constrained* subproblem must therefore lie on the intersection of the constraints at minimal distance from $\boldsymbol{\delta}^{(*)}$. This statement makes intuitive sense, and is also supported by referring to Eq. (3.19) which gives the subproblem's solution analytically as

$$\begin{aligned} \boldsymbol{\delta}^* &= (\boldsymbol{I} - \boldsymbol{P}) \left(\frac{-\nabla_{\boldsymbol{x}} f}{\omega} \right) - \boldsymbol{J}^{+\mathrm{T}} g(\boldsymbol{x}) \\ &= (\boldsymbol{I} - \boldsymbol{P}) \left(\frac{-\nabla_{\boldsymbol{x}} f}{\omega} \right) - \boldsymbol{J} (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J})^{-1} g(\boldsymbol{x}). \end{aligned} \tag{3.22}$$

This vector is written as the sum of two complementary components in the unconstrained and constrained subspaces, respectively: the first term is the negative gradient vector of $Q(\boldsymbol{\delta})$ evaluated at $\boldsymbol{\delta} = \mathbf{0}$ (recall that this is the current position of $\boldsymbol{x}$ in the search space) and projected into the unconstrained subspace, while the second term is the vector connecting $\boldsymbol{\delta} = \mathbf{0}$ orthogonally with the intersection of the linear constraints.

Moving towards the solution of the subproblem given by Eq. (3.18) therefore means moving in both the unconstrained subspace to minimize f and in the constrained subspace to minimize g. Across a sequence of such subproblems, the sequence of their solutions $\{\boldsymbol{\delta}^*\}$ approaching $\boldsymbol{\delta} = 0$ corresponds to the origin in $\boldsymbol{\delta}$-space approaching the solution $\boldsymbol{x}^*$ to the constrained problem. Referring to our understanding of each subproblem's solution in Eq. (3.22), this in turn corresponds with the first term being pushed to zero, satisfying the first-order condition of a stationary point for f, and the second term also being pushed to zero, satisfying the constraints.

For equality constrained problems with derivative information available, the multipliers $\boldsymbol{\alpha}$ can even be calculated directly from Eq. (3.17) without solving the actual subproblems. For inequality constrained problems, generalizing the approach relies

on their solution.

3.2.2 Exact Lagrangian for inequality constraints

Recall that the ICP asks for a solution $\boldsymbol{x}$ to the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0 \end{aligned}$$

with all the same notation as detailed in Section B.2, including the *active set* which is the collection of indexed constraints satisfied as equalities at the optimum. The subproblem analogous to Eq. (3.18) for inequality constraints involves solving

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & Q(\boldsymbol{\delta}) = \frac{\omega}{2}\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta} + \boldsymbol{\delta}^{\mathrm{T}}\nabla_{\boldsymbol{x}} f \\ \text{s.t.} \quad & \ell(\boldsymbol{\delta}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\delta} + g \leq 0. \end{aligned} \tag{3.23}$$

Importantly, a true solution in terms of $\boldsymbol{\delta}$ seems required, as using Eq. (3.17) to directly approximate multipliers $\boldsymbol{\alpha}$ is applicable only if the set of active constraints at the optimum is already known.

When using numerical methods, even arriving at this quadratic subproblem greatly simplifies the situation: it is generally easier to deal with a quadratic/linear problem than one in which the objective or constraint functions may be more general or more complex. Indeed, in proposing the extension of the exact Lagrangian to inequality constraints, Fletcher [41] applies a general quadratic programming routine to the subproblems in order to determine the associated Lagrange multipliers. However, for stochastic methods such as evolution strategies, the simplification from nonlinear to quadratic offers no immediate advantage. While the subproblems still provide a valuable framework for understanding the algorithm, actually solving a sequence of inequality subproblems would regress to a variation of the traditional inner/outer loop model of the method of multipliers that expends function evaluations while converging to intermediate values.

The approximations of Eq. (3.17) can still be used if the active constraints are known, so one alternative is to estimate the active set separately from the Lagrange multipliers and use this to calculate $\boldsymbol{\alpha}(\boldsymbol{x})$, allowing the estimated set to change as new constraint information is collected. Since any number of such changes may occur, it is useful to separately consider the *working set* $\mathcal{W}$ as the current best approximation to the true active set at the optimum, with the goal being $\mathcal{W}^{(k)} \to \mathcal{A}$ as $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$. Note that unlike in the original formulation of the exact Lagrangian, our resulting Lagrange multipliers will be discontinuous due to discrete transitions in the members of the working set. This results in the noted symptom of *zigzagging*, where the working set repeatedly adds then removes a constraint (or a set of constraints) so that the resulting multiplier approximations cause the algorithm solving the ICP to be drawn between alternating constraint boundaries instead of towards the constrained optimum. An implementation of an exact Lagrangian algorithm will need to address this concern.

Chapter 4

Augmented and exact Lagrangian evolution strategies

This chapter presents the augmented Lagrangian (AL-ES) and exact Lagrangian evolution strategies (EL-ES) for constrained optimization, respectively framed and justified in terms of the method of multipliers and Fletcher's exact method given in Chapter 3.

Moving from the context of numerical optimization to evolution strategies introduces several challenges. Most importantly, no first- or second-order derivative information is available. The AL-ES adaptively updates a penalty coefficient based on recent changes in the constraint and Lagrangian function values, while the EL-ES relies on expressions such as Eq. (3.17) which require approximating the involved terms. As each algorithm's progress through the search space is governed by stochastic processes, leading to changes in local values between iterations that resemble noise, it is a related concern that the resulting approximations are stable enough to be useful. Finally, the EL-ES requires careful management of the working set, as the automatic means of determining the active set as used in the exact Lagrangian method from numerical optimization cannot be meaningfully adapted for use with evolution strategies.

4.1 AL-ES for one constraint

Early work for this thesis led to the proposal by Arnold and Porter [14] of a novel augmented Lagrangian approach for a $(1 + 1)$-ES which demonstrates good convergence performance on n-dimensional spheres and moderately conditioned ellipsoids with a single constraint. This approach was later extended to handle multiple constraints by Atamna et al. [18, 19]. A key feature of the AL-ES algorithm is the integration of updates for the Lagrangian parameters alongside updates for the internal parameters of the evolution strategy adapted within every iteration.

Under the assumption of only one constraint, the original AL-ES from Arnold and Porter uses the corresponding augmented Lagrangian defined as

$$L_\omega(\boldsymbol{x}, \alpha) = f(\boldsymbol{x}) + \Psi(\boldsymbol{x})$$

where

$$\Psi(\boldsymbol{x}) = \begin{cases} \alpha g(\boldsymbol{x}) + \frac{1}{2}\omega g(\boldsymbol{x})^2 & \text{if } \alpha + \omega g(\boldsymbol{x}) \geq 0 \\ \frac{-\alpha^2}{2\omega} & \text{otherwise.} \end{cases} \tag{4.1}$$

Updates for α follow the method of multipliers, so that in iteration k the lone multiplier is updated according to

$$\begin{aligned} \alpha^{(k+1)} &= \left(\alpha^{(k)} + \omega^{(k)} g(\boldsymbol{x}^{(k)})\right)_+ \\ &= \max\left(0,\ \alpha^{(k)} + \omega^{(k)} g(\boldsymbol{x}^{(k)})\right). \end{aligned} \tag{4.2}$$

In order to permit updating the Lagrangian parameters within every iteration of the evolution strategy, the original AL-ES proposes to adaptively update the penalty coefficient ω based on changes in the constraint violation between parent and offspring candidate solutions. The goal for updating ω this way is to ensure a balance in progress for the evolution strategy between the constrained and unconstrained subspaces. In iteration k, the penalty coefficient is thus calculated as

$$\omega^{(k+1)} = \begin{cases} \omega^{(k)}\chi^{1/4} & \text{if } \omega^{(k)} g(\boldsymbol{x}^{(k)})^2 < k_1 |\Delta L^{(k)}|/n \text{ or } k_2 |\Delta g^{(k)}| < |g(\boldsymbol{x}^{(k)})| \\ \omega^{(k)}\chi^{-1} & \text{otherwise} \end{cases} \tag{4.3}$$

and this is used in defining the augmented Lagrangian of Eq. (4.1). The values of χ, k_1, and k_2 above are control parameters that affect how quickly the Lagrangian parameters (α and ω) are updated. In the original AL-ES of Arnold and Porter, values of $\chi = 2^{1/4}$, $k_1 = 3$, and $k_2 = 5$ are used. The delta values used in Eq. (4.3) represent changes in their respective functions between subsequent iterations with Lagrange parameters held fixed, so

$$\begin{aligned} \left|\Delta L^{(k+1)}\right| &= \left|L_{\omega^{(k)}}(\boldsymbol{x}^{(k+1)}, \alpha^{(k)}) - L_{\boldsymbol{\omega}^{(k)}}(\boldsymbol{x}^{(k)}, \alpha^{(k)})\right| \\ \left|\Delta g^{(k+1)}\right| &= \left|g(\boldsymbol{x}^{(k+1)}) - g(\boldsymbol{x}^{(k)})\right|. \end{aligned}$$

With this in mind, the conditions of the first line in Eq. (4.3) can be broken into two parts: the first aims to increase ω when changes in the augmented Lagrangian are due primarily to changes in the objective function over changes in the Lagrangian parameters, while the second aims to increase ω when overly small changes in constraint violation may signal that progress is slowing down.

Full details of the original $(1+1)$-AL-ES approach are given in Algorithm 4.1.

Algorithm 4.1 Single iteration of $(1+1)$-AL-ES

Require: $f : \mathbb{R}^n \to \mathbb{R},\ g : \mathbb{R}^n \to \mathbb{R}^m,\quad \chi > 1, k_1, k_2 > 0$

1: $\boldsymbol{z} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: $\boldsymbol{y} \leftarrow \boldsymbol{x} + \sigma \boldsymbol{z}$ ▷ Generate offspring
3: $\Delta g \leftarrow g(\boldsymbol{y}) - g(\boldsymbol{x})$
4: $\Delta_x L_\omega \leftarrow L_\omega(\boldsymbol{y}, \alpha) - L_\omega(\boldsymbol{x}, \alpha)$
5: **if** $\Delta_x L_\omega \leq 0$ **then**
6: $\quad \boldsymbol{x} \leftarrow \boldsymbol{y}$
7: $\quad \sigma \leftarrow \sigma \cdot 2^{1/n}$
8: $\quad \alpha \leftarrow \max(0, \alpha + \omega g(\boldsymbol{y}))$
9: $\quad$ **if** $\omega g^2(\boldsymbol{y}) < k_1 |\Delta_x L_\omega| / n$ **or** $k_2 |\Delta g| < |g(\boldsymbol{x})|$ **then** ▷ Update penalty
10: $\quad\quad \omega \leftarrow \omega \chi^{1/4}$
11: $\quad$ **else**
12: $\quad\quad \omega \leftarrow \omega \chi^{-1}$
13: $\quad$ **end if**
14: **else**
15: $\quad \sigma \leftarrow \sigma \cdot 2^{-1/(4n)}$
16: **end if**

A single offspring $\boldsymbol{y}$ is generated in each iteration, and values are calculated for both the constraint $g(\boldsymbol{y})$ and augmented Lagrangian L_ω defined using Eq. (4.4) in Lines 3 - 4. If the offspring $\boldsymbol{y}$ gives an improvement in L_ω over the parent $\boldsymbol{x}$, then the parent and step-size σ are updated in Lines 6 - 7, the Lagrange multiplier is updated in Line 8, and the penalty coefficient ω is updated in Lines 9 - 13. The condition on the update for ω aims to balance the progress of the evolution strategy on improving with respect to the constraints and improving with respect to the objective function, as well as avoiding premature stagnation signalled by rapidly decreasing magnitudes of change in the constraint violation. Convergence is observed on spheres and moderately conditioned

ellipsoids in the presence of a single linear constraint.

4.2 AL-ES for multiple constraints

A direct extension of the AL-ES to handle problems with multiple constraints is proposed by Atamna et al. [18, 19] for multimembered evolution strategies, where convergence properties are investigated analytically using a Markov chain approach. Improved parameter settings are investigated by Dufossé and Hansen [38] and compared experimentally alongside other approaches using CMA and surrogate models.

A summary and synthesis including these two additional approaches is presented here and differences are highlighted. Both of the additional implementations use $(\mu/\mu_W, \lambda)$-ES with weighted recombination and cumulative step-size adaptation. To allow for multiple constraints, the augmented Lagrangian is defined as usual for each implementation. Recall from Eq. (3.9) that this gives

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \Psi(\boldsymbol{x})$$

with

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{m} \begin{cases} \alpha_i g_i(\boldsymbol{x}) + \frac{1}{2}\omega_i g_i(\boldsymbol{x})^2 & \text{if } \alpha_i + \omega_i g_i(\boldsymbol{x}) \geq 0 \\ \frac{-\alpha_i^2}{2\omega_i} & \text{otherwise.} \end{cases} \tag{4.4}$$

Constraints that are inactive will correspond with the condition of the bottom row of Eq. (4.4), while constraints that are active will correspond with the top. Equivalently, constraint i is active when

$$-\alpha_i \leq \omega_i g_i(\boldsymbol{x}). \tag{4.5}$$

The multipliers α_i are written here as elements of the vector $\boldsymbol{\alpha}$, while the penalty coefficients are written as the vector $\boldsymbol{\omega}$ with (possibly distinct) elements ω_i forming the diagonal of $\boldsymbol{\Omega} = \boldsymbol{\omega}\boldsymbol{I}$.

Updating Lagrange multipliers

Each of the AL approaches implement variations on the method of multipliers, so that in iteration k the Lagrange multipliers $\boldsymbol{\alpha}$ are updated according to Eq. (4.2). Both Atamna et al. and Dufossé and Hansen use an additional damping factor $\frac{1}{d_\alpha}$

to slow down the adaptation of $\boldsymbol{\alpha}$, while Atamna et al. do not use the plus operator $(\cdot)_+$, resulting in update rules of

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \frac{1}{d_\alpha} \cdot \boldsymbol{\Omega}^{(k)} g(\boldsymbol{x}^{(k)})$$

for Atamna et al., and

$$\boldsymbol{\alpha}^{(k+1)} = \left(\boldsymbol{\alpha}^{(k)} + \frac{1}{d_\alpha} \cdot \boldsymbol{\Omega}^{(k)} g(\boldsymbol{x}^{(k)}) \right)_+$$

for Dufossé and Hansen, which results in each element being updated as

$$\alpha_i^{(k+1)} = \max \left(0,\ \alpha_i^{(k)} + \frac{1}{d_\alpha} \cdot \omega_i \cdot g_i(\boldsymbol{x}^{(k)}) \right).$$

Updating penalty terms

The original AL-ES assumes only a single penalty coefficient and uses the update rule of Eq. (4.3) in each iteration, while the others allow for multiple distinct penalty coefficients and so use the modified rule

$$\omega_i^{(k+1)} = \begin{cases} \omega_i^{(k)} \chi^{1/4} & \text{if } \omega_i^{(k)} g_i(\boldsymbol{x}^{(k)})^2 < k_1 |\Delta L^{(k)}|/n \text{ or } k_2 |\Delta g_i^{(k)}| < |g_i(\boldsymbol{x}^{(k)})| \\ \omega_i^{(k)} \chi^{-1} & \text{otherwise} \end{cases}$$

with minor differences in the fixed parameters. The AL-ES rule from Arnold and Porter uses $k_1 = 3$, $k_2 = 5$, and $\chi = 2^{1/4}$, while Atamna et al. use $\chi = 2^{1/5n}$ and Dufossé and Hansen use $\chi = 2^{1/\sqrt{n}}$.

Dufossé and Hansen are the only ones to give a recommended initialization for $\boldsymbol{\omega}$, derived by first calculating the inter-decile range (IDR) for the objective and constraint functions among the first set of offspring as

$$\begin{aligned} \underset{i \leq \lambda}{\text{IDR}}(f(\boldsymbol{y}^i)) &= \Delta f, \\ \underset{i \leq \lambda}{\text{IDR}}(g_j(\boldsymbol{y}^i)) &= \Delta g_j, \end{aligned} \tag{4.6}$$

and then setting

$$\omega_i^{(0)} = 10^2 \cdot \frac{\Delta f}{\Delta g_i^2}.$$

Additionally, Dufossé and Hansen add a check whereby the penalty term ω_i is updated only if it is associated with an active constraint according to Eq. (4.5).

Applying CMA

Only Dufossé and Hansen apply CMA to allow for application to a broader selection of problems. Atamna et al. consider CMA elsewhere [17] to determine convergence results, but only in the limited case of a single linear constraint. Dufossé and Hansen note that since CMA should be more adept at handling ill-conditioning, it should be possible to allow the penalty terms ω_i to become larger and hopefully speed up convergence. To this end, they set the fixed parameter $k_1 = 10$, while keeping all others the same. A comprehensive experimental comparison is given using the AL-CMA-ES formulation (identified there as "AL many"), and the result is an apparently widely applicable algorithm.

4.3 EL-ES algorithm

The EL-ES algorithm is presented here, which is the familiar $(\mu/\mu_W, \lambda)$-ES along with calculations for updating the Lagrange parameters and managing the working set in order to implement an exact Lagrangian approach. The algorithm itself is given first, and its main operations are presented as three subroutines: two that update the working set through expansion and pruning, and one that uses local/global approximations to estimate values for the exact Lagrangian parameters $\boldsymbol{\alpha}$ and ω. One additional subroutine is called occasionally in order to maintain linear independence of constraints within the working set.

Throughout, the discussion will focus on the case of inequality constraints, as these pose the greatest difficulty in terms of determining their inclusion in the working set. This is without any loss of generality, as equality constraints may be considered as having been converted to the double-sided inequality constraints $g(\boldsymbol{x}) \leq 0$ and

$-g(\boldsymbol{x}) \leq 0$. In practice of course, an equality constraint should be explicitly guaranteed inclusion in the working set, in which case the rest of the proposed EL-ES algorithm could proceed with only trivial adjustments to accommodate this fact.

4.3.1 Algorithm outline

The main idea of the EL-ES is to use an evolution strategy to solve a constrained optimization problem (GCP) by minimizing an approximation of the unconstrained exact Lagrangian given in Eq. (3.14). In each iteration, local information from the offspring selection process inherent to the ES is combined with historical information from previous iterations to give approximate values for the Lagrangian parameters $\boldsymbol{\alpha}(\boldsymbol{x})$ and ω in terms of the current location in the search space. Although this approach on its own gives good convergence results on some problems, it can encounter difficulties with arrangements of constraints that produce instability in the iterative working set approximations. In the case of inaccurate estimates for the Lagrange multipliers using Eq. (3.17) or oscillating constraints that are repeatedly added and removed from the working set $\mathcal{W}$, a good strategy is to stay close to the current constraint boundaries defined as active by $\mathcal{W}$ until either an optimum is found or else the working set is reliably updated. To encourage this behaviour, the EL-ES evaluates offspring against two separate objective functions to determine their rankings: the Lagrangian $\phi(\boldsymbol{x})$ and a pure penalty function defined as

$$Q_{\text{pen}}(\boldsymbol{x}) = g_{\mathcal{W}}(\boldsymbol{x})^{\mathrm{T}} g_{\mathcal{W}}(\boldsymbol{x}) \tag{4.7}$$

that uses only local evaluations of constraint function $g_{\mathcal{W}}$ evaluated on constraints in $\mathcal{W}$ at the current centroid. Once the offspring rankings are calculated separately for ϕ and Q_{pen} the rankings themselves are summed to establishing a new ranking. This explicitly assigns equal weight to minimizing the violation for all constraints in $\mathcal{W}$ and minimizing the approximated Lagrangian, encouraging progress in the search space within close proximity to the intersection of the active constraint boundaries. This is the desired outcome in most cases; however, it may rarely occur that the offspring rankings are exactly reversed for Q_{pen} and ϕ, in which case the sum of equally

weighted rankings will give identical values for all offspring. Offspring selection in that iteration would then degrade to a random walk. To avoid this disruptive outcome, a tie-breaking process is needed. The result of applying $\oplus_\epsilon$ can thus be summarized as taking the sum of both sets of ranks to be the new rankings, while preferring the rankings given by Q in the case of a tie.

Sums of offspring rankings appear in other constraint-handling methods for evolution strategies proposed by Runarsson and Yao [104] as part of stochastic ranking, and by Sakamoto and Akimoto [105, 106] as part of ARCH. For the EL-ES, the merging of ranks is performed using a simple sum together with the given tie-breaking procedure.

Algorithm 4.2 Single iteration of $(\mu/\mu, \lambda)$-EL-ES with CSA

Require: $f : \mathbb{R}^n \to \mathbb{R},\ g : \mathbb{R}^n \to \mathbb{R}^m,\quad c \in (0, 1),\quad D > 0,\ \sum_i^\mu w_i = 1$

1: ExpandWS()
2: PruneWS()
3: UpdateAlphaOmega()

4: **for** $\ell = 1 \to \lambda$ **do**
5: $\quad \boldsymbol{z}_\ell \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Generate offspring
6: $\quad \boldsymbol{y}_\ell \leftarrow \boldsymbol{x} + \sigma \boldsymbol{z}_\ell$
7: **end for**

8: $\text{sort}([\boldsymbol{z}_{1:\lambda}],\ [\phi(\boldsymbol{y}_1) \oplus_\epsilon Q_{\text{pen}}(\boldsymbol{y}_1), \ldots, \phi(\boldsymbol{y}_\lambda) \oplus_\epsilon Q_{\text{pen}}(\boldsymbol{y}_\lambda)])$ ▷ Combine ranks
9: $\hat{\boldsymbol{z}} \leftarrow \sum_{\ell=1}^{\mu} w_\ell \boldsymbol{z}_\ell$
10: $\boldsymbol{x} \leftarrow \boldsymbol{x} + \sigma \hat{\boldsymbol{z}}$
11: $\boldsymbol{s} \leftarrow (1 - c)\boldsymbol{s} + \sqrt{\mu_{\text{eff}} c(2 - c)}\hat{\boldsymbol{z}}$ ▷ Update $\boldsymbol{s}$
12: $\sigma \leftarrow \sigma \cdot \exp^{\frac{c}{D}} \left(\frac{\|\boldsymbol{s}\|}{\mathbb{E}\left[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|\right]} - 1 \right)$ ▷ Update σ

An outline of the multimembered exact Lagrangian evolution strategy is given in Algorithm 4.2, using calls to subroutines that are defined in subsequent sections of this chapter. In Lines 1 - 3, three subroutines are called for expanding the working set, pruning the working set, and updating the Lagrange parameters. Details for these operations are given in Sections 4.3.2 and 4.3.3. In Lines 4 - 7, λ offspring are generated by sampling from an n-dimensional normal distribution. Both the exact

Lagrangian $\phi(\boldsymbol{x})$ and the penalty function of Eq. (4.7) are evaluated for each offspring, and in Line 8 the operator

$$\phi(\boldsymbol{y}_i) \oplus_\epsilon Q_{\text{pen}}(\boldsymbol{y}_i)$$

is used to indicate the sum of the rank of offspring $\boldsymbol{y}_i$ according to $\phi(\boldsymbol{y}_i)$ (scaled by $(1-\epsilon)$) and the rank according to $Q_{\text{pen}}(\boldsymbol{y}_i)$. This joint ranking is used to sort the offspring, which are then combined in Lines 9 and 10 to form a new parental centroid $\boldsymbol{x}$. Finally, the evolution path $\boldsymbol{s}$ and step size σ are updated in Lines 11 and 12.

4.3.2 Calculating Lagrange parameters

Central to the EL-ES algorithm is using the approximation given by Eq. (3.17) to determine the Lagrange multipliers. This is initially problematic in the context of black-box algorithms like evolution strategies, as none of the gradient information will be readily available. Instead, it is necessary to calculate local approximations to relevant terms. This is done by taking advantage of the objective and constraint function evaluations used for ranking the offspring of a multimembered evolution strategy, reducing the need for extra function evaluations to only evaluating the centroid as part of the approximation process. The subroutine for approximating the Lagrange multipliers $\boldsymbol{\alpha}$ is called once per iteration of the EL-ES and is given in Algorithm 4.3. The first line is a call to the subroutine detailed in Algorithm 4.6 which ensures that the matrix inversion of the next line is operating on a nonsingular matrix; in other words, that the Jacobian of the constraints in the working set are linearly independent, as in Theorem B.6. An explanation of the other components follows.

The multiplier expression in Eq. (3.17) can be expanded as

$$\begin{aligned}\boldsymbol{\alpha}(\boldsymbol{x}) &= -\boldsymbol{J}^{+}\nabla f(\boldsymbol{x}) + \omega \cdot (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1} g(\boldsymbol{x}) \\ &= -\underbrace{(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}}_{\boldsymbol{\alpha}_A(\boldsymbol{x})} \underbrace{\boldsymbol{J}^{\mathrm{T}}\nabla f(\boldsymbol{x})}_{\boldsymbol{\alpha}_B(\boldsymbol{x})} + \omega \cdot \underbrace{(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}}_{\boldsymbol{\alpha}_A(\boldsymbol{x})} g(\boldsymbol{x})\end{aligned}$$

showing the collected terms $\boldsymbol{\alpha}_A(\boldsymbol{x})$ and $\boldsymbol{\alpha}_B(\boldsymbol{x})$ are the only unknowns and are determined with respect to $\boldsymbol{x}$. One approach for reliably approximating these values is to blend together local and historical information taken from offspring evaluations on

Algorithm 4.3 Subroutine for updating $\boldsymbol{\alpha}, \omega$

Require: Current values for $f(\boldsymbol{y}_i), g(\boldsymbol{y}_i)$ for $i = 1 : \lambda$ offspring, $g(\boldsymbol{x})$ for centroid, step size σ, fixed learning rate $c_{\boldsymbol{\alpha}} \leq 1$

function UPDATEALPHAOMEGA
  ENFORCELI() ▷ Algorithm 4.6
  $\boldsymbol{\alpha}_A \leftarrow \frac{1}{\sigma^2} \operatorname{cov}_i (g(\boldsymbol{y}_i))$ ▷ Eq. (4.11)
  $\boldsymbol{\alpha}_B \leftarrow \frac{1}{\sigma^2} \operatorname{cov}_i (g(\boldsymbol{y}_i), f(\boldsymbol{y}_i))$ ▷ Eq. (4.12)
  $\omega \leftarrow \frac{1}{2} \cdot \min\big(\frac{1}{\sigma}\operatorname{std}_i(f(\boldsymbol{y}_i)), \frac{1}{\sigma^2}\operatorname{std}_i(\phi(\boldsymbol{y}_i))\big)$ ▷ Eq. (4.10)

  $\overline{g} \leftarrow (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{g} + c_{\boldsymbol{\alpha}} \cdot g(\boldsymbol{x})$ ▷ Eq. (4.9)
  $\overline{\boldsymbol{\alpha}_A} \leftarrow (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\boldsymbol{\alpha}_A} + c_{\boldsymbol{\alpha}} \cdot \boldsymbol{\alpha}_A$
  $\overline{\boldsymbol{\alpha}_B} \leftarrow (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\boldsymbol{\alpha}_B} + c_{\boldsymbol{\alpha}} \cdot \boldsymbol{\alpha}_B$

  $\overline{\omega} \leftarrow (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\omega} + c_{\boldsymbol{\alpha}} \cdot \omega$
  $\overline{\boldsymbol{\alpha}} \leftarrow -(\overline{\boldsymbol{\alpha}_A})^{-1} \cdot \overline{\boldsymbol{\alpha}_B} + \overline{\omega} \cdot (\overline{\boldsymbol{\alpha}_A})^{-1} \cdot \overline{g}$ ▷ Eq. (4.8)
end function

functions f and g. This results in a modified expression for the Lagrange multipliers

$$\begin{aligned} \overline{\boldsymbol{\alpha}} &= -(\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1} \cdot \overline{\boldsymbol{J}^{\mathrm{T}}\nabla f} + \overline{\omega} \cdot (\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1} \cdot \overline{g} \\ &= -(\overline{\boldsymbol{\alpha}_A})^{-1} \cdot \overline{\boldsymbol{\alpha}_B} + \overline{\omega} \cdot (\overline{\boldsymbol{\alpha}_A})^{-1} \cdot \overline{g} \end{aligned} \tag{4.8}$$

where the bar notation indicates exponential fading is used to combine values from the current iteration with the previous estimate. It is also understood that while these values are accumulated across all constraints in each iteration, only the elements corresponding to constraints in the working set are used in updating $\overline{\boldsymbol{\alpha}}$ in the expression above and in the discussion that follows. Each component of Eq. (4.8) is exponentially faded using the same positive learning rate $c_{\boldsymbol{\alpha}} \leq 1$ as

$$\begin{aligned} \overline{g}^{(k)} &= (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{g}^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot g(\boldsymbol{x}^{(k)}) \\ \overline{\omega}^{(k)} &= (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\omega}^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot \omega \\ \overline{\boldsymbol{\alpha}_A}^{(k)} &= (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\boldsymbol{\alpha}_A}^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot \boldsymbol{\alpha}_A(\boldsymbol{x}^{(k)}) \\ \overline{\boldsymbol{\alpha}_B}^{(k)} &= (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{\boldsymbol{\alpha}_B}^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot \boldsymbol{\alpha}_B(\boldsymbol{x}^{(k)}) \end{aligned} \tag{4.9}$$

so that it only remains to calculate the values given in the right-most terms of each sum above. The learning rate is set to $c_{\boldsymbol{\alpha}} = c_\sigma$ as preliminary work on tuning parameter $c_{\boldsymbol{\alpha}}$ indicates that this value is roughly appropriate on the selection of problems considered in Chapter 5. The value of $g(\boldsymbol{x}^{(k)})$ is simply the constraint violation of the centroid in iteration k, while the value of ω is calculated in each iteration as

$$\omega = \frac{1}{2} \cdot \min\left[\frac{\underset{i}{\text{std}}(f(\boldsymbol{y}_i))}{\sigma}, \frac{\underset{i}{\text{std}}(\phi(\boldsymbol{y}_i))}{\sigma^2}\right]. \tag{4.10}$$

This attempts to maintain locally positive curvature of ϕ with the smallest needed value. From Eq. (3.21), the quadratic subproblems will stagnate and $\boldsymbol{\delta}^{(*)}$ will approach 0 as $\omega \to \infty$, so there is motivation in using a value of the penalty coefficient that is no larger than necessary. The terms $\boldsymbol{\alpha}_A(\boldsymbol{x}^{(k)})$ and $\boldsymbol{\alpha}_B(\boldsymbol{x}^{(k)})$ are each estimates calculated by using information around the centroid in the k-th iteration as

$$\begin{aligned}\boldsymbol{\alpha}_A(\boldsymbol{x}^{(k)}) &= \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} \\ &\approx \frac{1}{\sigma^2} \cdot \underset{i}{\text{cov}}\left(g(\boldsymbol{y}_i^{(k)})\right)\end{aligned} \tag{4.11}$$

and

$$\begin{aligned}\boldsymbol{\alpha}_B(\boldsymbol{x}^{(k)}) &= \boldsymbol{J}^{\mathrm{T}}\nabla f(\boldsymbol{x}^{(k)}) \\ &\approx \frac{1}{\sigma^2} \cdot \underset{i}{\text{cov}}\left(g(\boldsymbol{y}_i^{(k)}), f(\boldsymbol{y}_i^{(k)})\right).\end{aligned} \tag{4.12}$$

Justification for both of these expressions is given by using the definition of covariance taken across the offspring $\boldsymbol{y}_i$ to construct a linear approximation that will be increasingly accurate as σ decreases. The approximation for $\boldsymbol{\alpha}_A$ is derived by starting with

$$\begin{aligned}\underset{i}{\text{cov}}\left(g(\boldsymbol{y}_i)\right) &= \mathbb{E}\left[(g(\boldsymbol{y}_i) - \mathbb{E}[g(\boldsymbol{y}_i)])(g(\boldsymbol{y}_i) - \mathbb{E}[g(\boldsymbol{y}_i)])^{\mathrm{T}}\right] \\ &\approx \mathbb{E}\left[(g(\boldsymbol{y}_i) - g(\boldsymbol{x}^{(k)}))(g(\boldsymbol{y}_i) - g(\boldsymbol{x}^{(k)}))^{\mathrm{T}}\right]\end{aligned}$$

and using the first-order approximation of the differences this becomes

$$\begin{aligned}
&\approx \mathbb{E}\left[\left(\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^{\mathrm{T}}(\boldsymbol{y}_i - \boldsymbol{x}^{(k)})\right)\left(\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^{\mathrm{T}}(\boldsymbol{y}_i - \boldsymbol{x}^{(k)})\right)^{\mathrm{T}}\right] \\
&= \mathbb{E}\left[\left(\sigma^{(k)}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{z}_i\right)\left(\sigma^{(k)}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{z}_i\right)^{\mathrm{T}}\right] \\
&= (\sigma^{(k)})^2 \cdot \boldsymbol{J}^{\mathrm{T}} \cdot \mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right] \cdot \boldsymbol{J} \\
&= \sigma^2 \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}.
\end{aligned}$$

The final line corresponds with Eq. (4.11), and follows from the vectors $\boldsymbol{z}_i$ being standard normally distributed, giving the off-diagonal elements of the matrix $\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}$ expected values of 0 and the diagonal elements χ^2 distributed with mean $k = 1$, thus $\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right] = \boldsymbol{I}$.

By a similar calculation, the approximation for $\boldsymbol{\alpha}_B$ is

$$\begin{aligned}
\operatorname*{cov}_i\left(g(\boldsymbol{y}_i), f(\boldsymbol{y}_i)\right) &= \mathbb{E}\left[(g(\boldsymbol{y}_i) - \mathbb{E}[g(\boldsymbol{y}_i)])(f(\boldsymbol{y}_i) - \mathbb{E}[f(\boldsymbol{x}_i)])^{\mathrm{T}}\right] \\
&\approx \mathbb{E}\left[(g(\boldsymbol{y}_i) - g(\boldsymbol{x}^{(k)}))(f(\boldsymbol{y}_i) - f(\boldsymbol{x}^{(k)}))^{\mathrm{T}}\right] \\
&\approx \mathbb{E}\left[\left(\sigma^{(k)}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{z}_i\right)\left(\sigma^{(k)} \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)})^{\mathrm{T}} \cdot \boldsymbol{z}_i\right)^{\mathrm{T}}\right] \\
&= (\sigma^{(k)})^2 \cdot \boldsymbol{J}^{\mathrm{T}} \cdot \mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right] \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}) \\
&= \sigma^2 \boldsymbol{J}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} f
\end{aligned} \tag{4.13}$$

which matches with Eq. (4.12).

Combining the approximations in Eqs. (4.11) and (4.12) with Eq. (4.9) and $\overline{\omega}$, we have every term needed to calculate the faded multiplier vector in Eq. (4.8). In each iteration, this approximates the Lagrange multipliers of Eq. (3.17) for the inequality subproblem in Eq. (3.23), which are in turn equal to the Lagrange multipliers of the underlying ICP by extension of Theorem 3.1. At each stage of the EL-ES, the value of $\overline{\boldsymbol{\alpha}}$ represents the best estimate for the current Lagrange multipliers which should be minimized against.

Using Eq. (4.8) to write this as part of the Lagrangian seen by the ES, we have

$$\begin{aligned}\phi(\boldsymbol{x}) &= f(\boldsymbol{x}) + g(\boldsymbol{x})^{\mathrm{T}}\big(-(\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1}\cdot\overline{\boldsymbol{J}^{\mathrm{T}}\nabla f} + \overline{\omega}\cdot(\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1}\cdot\overline{g}\big)\\ &= f(\boldsymbol{x}) + g(\boldsymbol{x})^{\mathrm{T}}\big(-\overline{\boldsymbol{\alpha}_A}\cdot\overline{\boldsymbol{\alpha}_B} + \overline{\omega}\cdot\overline{\boldsymbol{\alpha}_A}\cdot\overline{g}\big)\\ &= f(\boldsymbol{x}) + g(\boldsymbol{x})^{\mathrm{T}}\overline{\boldsymbol{\alpha}}. \end{aligned} \tag{4.14}$$

Note that while this is written as an ordinary Lagrange function above, it is also similar in form to the augmented Lagrangian

$$L_\beta(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\alpha}_0 + g(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\beta}g(\boldsymbol{x}). \tag{4.15}$$

if we take the multipliers to be

$$\boldsymbol{\alpha}_0 = -(\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1}\cdot\overline{\boldsymbol{J}^{\mathrm{T}}\nabla f}$$

and augmenting term

$$\boldsymbol{\beta} = \overline{\omega}(\overline{\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}})^{-1}.$$

In comparing the Lagrangian function $\phi(\boldsymbol{x})$ with the augmented Lagrangian $L_\beta(\boldsymbol{x})$ so constructed, the only apparent difference is in their augmenting terms: while the ordinary Lagrangian $\phi(\boldsymbol{x})$ uses the averaged value $\overline{g}$ of constraint violations in the final term, as seen in the first line of Eq. (4.14), the augmented Lagrangian $L_\beta(\boldsymbol{x})$ of Eq. (4.15) uses only the local value $g(\boldsymbol{x})$. In spite of this small difference, the impact is significant. In the former, we use $\overline{g}$ to approximate a continuous function $\boldsymbol{\alpha}(\boldsymbol{x})$ that gives Lagrange multipliers corresponding to the solution of the local subproblem consisting of quadratic and linear approximations of f and g_i, respectively. Assuming the objective function is locally quadratic and the constraints locally linear, calculating the subproblem multipliers can lead to a good estimate for the true multipliers of the ICP. The main effect of $\overline{\omega}$ is in matching the subproblem's quadratic approximation to the underlying objective function f. In the latter, the multipliers $\boldsymbol{\alpha}_0$ are determined without any reference to the value of $g(\boldsymbol{x})$, and instead can be determined using the same approach as in Theorem 3.1 to correspond to an unbounded linear subproblem with no constrained minimum. The effect of $\overline{\omega}$ is to moderate the effect

of the augmenting penalty term.

4.3.3 Working set management

The discussion in Section 4.3.2 assumes that the active set $\mathcal{A}$ is known, consisting of those constraint indices satisfying $g_i(\boldsymbol{x}^*) = 0$ at the optimum. This is not generally a realistic assumption, and so instead calculations like Eq. (4.8) rely on the current working set $\mathcal{W}$ being a reasonable approximation. At any stage of the algorithm, constraints in the working set are treated as needing to be satisfied as equalities, while constraints not in the working set are disregarded both in terms of calculating Lagrange multipliers and in terms of ranking offspring. Since a constraint g_i is considered active at point $\boldsymbol{x}$ if $g_i(\boldsymbol{x}) \geq 0$, and the working set aims to converge to the set of active constraints as $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$, a simple approach would be to define $\mathcal{W} = \{i \ : \ g_i(\boldsymbol{x}^{(k)}) \geq 0\}$. However, this definition is inherently unstable due to the stochastic nature of an evolution strategy that may move unpredictably between feasible and infeasible regions near a constraint boundary. Instead, separate processes are defined below for expanding and pruning the working set. The pruning process attempts to remove a constraint from the working set based on the existence of negative Lagrange multipliers, while the expansion process attempts to add indices of constraints that are recently violated. If either the size of the working set $|\mathcal{W}| \leq m$ becomes greater than the dimension n of the search space, or if a constraint is added that is linearly dependent with the existing working set, then steps are taken to prune $\mathcal{W}$ and restore linear independence.

4.3.4 Normalized constraint violation

Constraint violation, whether for a feasible or infeasible point, is difficult to compare between constraints with potentially different scaling and between iterations while the ES moves stochastically through the search space. In order to allow meaningful comparisons, a single normalized value is calculated for each constraint in order to represent the magnitude of recent violations.

The normalization occurs by using an approximated positive linear scaling factor for each constraint. Between any two points $\boldsymbol{x}$ and $\boldsymbol{y}$, a finite difference of the j-th constraint values $g_j(\boldsymbol{y}) - g_j(\boldsymbol{x})$ divided by the distance $||\boldsymbol{y} - \boldsymbol{x}||$ gives an approximation of the linear scaling of the constraint function between those points. By taking the average of the finite differences of an evaluated constraint across all $\boldsymbol{y}_i$ offspring, we arrive at

$$\frac{1}{\lambda} \sum_{i=1}^{\lambda} \frac{(g_j(\boldsymbol{y}_i) - g_j(\boldsymbol{x}))}{||\boldsymbol{y}_i - \boldsymbol{x}||}$$

as an approximation of the local scaling. Since the distance between offspring and centroid is well approximated by the step size σ of the evolution strategy, the square of the linear scaling is well approximated by the variance divided by σ^2 as

$$\frac{1}{\lambda} \sum_{i=1}^{\lambda} \frac{(g_j(\boldsymbol{y}_i) - g_j(\boldsymbol{x}))^2}{\sigma^2}$$

and so the linear scaling factor itself is well approximated by the standard deviation.

The ratio of the constraint violation $g_j(\boldsymbol{x}^{(k)})$ to the normalizing factor $\underset{i}{\text{std}}(g_j(\boldsymbol{y}_i^{(k)}))$ is therefore a candidate for a normalized constraint violation in iteration k. In order to smooth the value between iterations, accumulation is additionally used in order to calculate

$$\overline{g}_j^{(k)} = (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{g}_j^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot g_j(\boldsymbol{x}^{(k)}),$$

which is given previously in Eq. (4.9), and

$$\overline{d}_j^{(k)} = (1 - c_{\boldsymbol{\alpha}}) \cdot \overline{d}_j^{(k-1)} + c_{\boldsymbol{\alpha}} \cdot \frac{1}{\sigma} \underset{i}{\text{std}}(g_j(\boldsymbol{y}_i^{(k)})).$$

calculated in the same manner. The normalized constraint violation $v_j^{(k)}$ for the j-th constraint in the k-th iteration is then defined as the associated ratio of the faded constraint value to the faded standard deviation, expressed as

$$v_j^{(k)} = \frac{\overline{g}_j^{(k)}}{\overline{d}_j^{(k)}}. \tag{4.16}$$

4.3.5 Expanding the working set

The subroutine for expanding $\mathcal{W}$ is given in Algorithm 4.4 and is called once per iteration of the evolution strategy. Expanding the working set involves potentially adding to its indexed list of constraints. First, an initial set of eligible constraints is constructed, consisting of all violated constraints not already indexed in the working set. The associated normalized constraint violations of Eq. (4.16) are compared and the single constraint with the largest violation is considered for inclusion in $\mathcal{W}$.

Algorithm 4.4 Subroutine for expanding the working set

Require: $\mathcal{W}$, v_i for constraints not in working set

```
1: function ExpandWS
2:     if max_{i∉W}(v_i) > 0 then                ▷ Eq. (4.16)
3:         W = W ∪ {i}
4:     end if
5: end function
```

The most significant potential complication of this subroutine is if a constraint is added that causes the working set to become linearly dependent. Identifying this situation proactively involves matrix operations in $|\mathcal{W}| \leq m$ dimensions. Additionally, a proactive check would have to consider all constraints in the working set, not just those recently added: constraints, especially nonlinear constraints, may be added to the working set while their locally approximated normals are independent, only to approach dependence as the ES approaches the optimum. For these reasons, the difficulty is addressed elsewhere within Algorithm 4.3 by calling the subroutine given by Algorithm 4.6.

4.3.6 Pruning the working set

The subroutine for pruning $\mathcal{W}$ is given in Algorithm 4.5 and is called once per iteration of the evolution strategy. Pruning the working set involves potentially removing one of its indexed constraints, and is comprised of a two-stage process: first deciding whether any constraints at all should be considered for removal, then potentially

choosing which constraint is best to remove. The case where the working set is over-constrained is also considered.

Algorithm 4.5 Subroutine for pruning the working set

Require: $\mathcal{W}$, v_i for $i \in \mathcal{W}$

1: **function** PRUNEWS
2: $\quad\Delta_f \leftarrow |f(\boldsymbol{x}^{(k-1)}) - f(\boldsymbol{x}^{(k)})|$ ▷ k is current iteration
3: $\quad\Delta_h \leftarrow |f(\boldsymbol{x}^{(k-1)}) - f(\boldsymbol{x}^{(e)})|$ ▷ e is iteration of last removal
4: $\quad$**if** $\Delta_f < \Delta_h$ **then**
5: $\quad\quad$**if** $\min_{i\in\mathcal{W}}(\alpha_i) < 0$ **then**
6: $\quad\quad\quad\mathcal{W} \leftarrow \mathcal{W} \setminus \left\{\operatorname{argmin}_{i\in\mathcal{W}}(\alpha_i)\right\}$ ▷ Remove index i from set $\mathcal{W}$
7: $\quad\quad\quad e \leftarrow k$
8: $\quad\quad$**else**
9: $\quad\quad\quad$**if** $|\mathcal{W}| > n$ & $\min_{i\in\mathcal{W}}(v_i) < 0$ **then** ▷ v_i from Eq. (4.16)
10: $\quad\quad\quad\quad\mathcal{W} \leftarrow \mathcal{W} \setminus \left\{\operatorname{argmin}_{i\in\mathcal{W}}(v_i)\right\}$
11: $\quad\quad\quad\quad e \leftarrow k$
12: $\quad\quad\quad$**end if**
13: $\quad\quad$**end if**
14: $\quad$**end if**
15: **end function**

The initial decision is based on a simple idea of Fletcher's [43], which is to compare the historical change in f over recent iterations to the expected change in f for the next iteration if the current working set were to be maintained. When the expected change in f under the current working set is sufficiently large, the working set is locked and no removals are allowed; the heuristic principle here is that in order to avoid unnecessary oscillations in the working set, a constraint should be removed from $\mathcal{W}$ only when there is evidence that better progress on minimizing f can be made without it. For an evolution strategy, the expected change can be calculated as a direct difference between candidate solutions using the same values of $\boldsymbol{\alpha}$, as

$$\Delta_f = |f(\boldsymbol{x}^{(k-1)}) - f(\boldsymbol{x}^{(k)})|$$

and then compared to the historical change in f since the last iteration e in which a

constraint was removed from the working set

$$\Delta_h = |f(\boldsymbol{x}^{(k-1)}) - f(\boldsymbol{x}^{(e)})|.$$

Whenever the expected future change in f is less than the historical change in f since the last constraint removal, the working set is eligible for constraint removal. Otherwise, no constraint is removed. Note that this incurs one extra function evaluation per iteration on the candidate solution $\boldsymbol{x}^{(k)}$ calculated before updating the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$; selection and recombination is performed first using the older values of $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$, then again after those values have been updated.

Once $\mathcal{W}$ is marked eligible for a removal, the values α_i for the Lagrange multipliers indexed by $i \in \mathcal{W}$ are compared and the constraint with the most negative Lagrange multiplier is removed. If no negative multiplier is found, an additional check is made whether the working set is over-constrained with $|\mathcal{W}| > n$. If so, the constraint with the minimal normalized constraint violation v_j from Eq. (4.16) is removed.

4.3.7 Enforcing linear independence

The subroutine for enforcing linear independence within $\mathcal{W}$ is given in Algorithm 4.6. As it can be a computationally expensive operation involving matrix operations based on the number of constraints, it is called only when a singular matrix is encountered, such as while attempting to calculate the inverse in Eq. (4.11) as part of Algorithm 4.3.

Algorithm 4.6 Subroutine for enforcing linear independence in the working set

Require: $\mathcal{W}$, tolerance $\epsilon > 0$

1: **function** EnforceLI
2: **while** $\mathcal{W}$ is linearly dependent **do**
3: $(u_1, \dots), (\boldsymbol{w}_1, \dots) = \text{eig}\left[\underset{i}{\text{cov}}(g(\boldsymbol{y}_i))\right]$ ▷ Eq. (4.11)
4: **for** $j = 1 \to |\mathcal{W}|$ **do**
5: **if** $|u_j| < \epsilon$ **then**
6: $\mathcal{B} = \{k : |[\boldsymbol{w}_j]_k| > \epsilon\}$
7: $\mathcal{W} = \mathcal{W} \setminus \underset{k \in \mathcal{B}}{\text{argmin}}(v_k)$
8: **end if**
9: **end for**
10: **end while**
11: **end function**

The subroutine relies on the fact that linearly dependent vectors in the column-space of a matrix will have corresponding zero eigenvalues. This fact is used to identify and consider for removal those constraints in $\mathcal{W}$ that are causing the set to be linearly dependent.

Line 3 of Algorithm 4.6 uses the covariance approximation of $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ from Eq. (4.11), calculated using constraints in the current working set. If any collection of the constraints in $\mathcal{W}$ are linearly dependent, then the matrix is singular and there should be corresponding zero eigenvalues. Let $u = 0$ be one such eigenvalue with associated eigenvector $\boldsymbol{w}$ of the approximated matrix $\boldsymbol{\alpha}_A \approx \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ that is not of full rank. Then the entries of $\boldsymbol{w}$ also give the coefficients of a linear combination of the columns of $\boldsymbol{\alpha}_A$ that equal zero, since

$$\boldsymbol{\alpha}_A \cdot \boldsymbol{w} = u = 0$$

by the definition of an eigenvector. If we consider the indices of the non-zero entries of $\boldsymbol{w}$, then these give the indices of the columns of $\boldsymbol{\alpha}_A$ appearing in the linear combination, and thus give a collection of columns that form a linearly dependent set. If there are multiple zero eigenvalues, then the same process can be repeated by analyzing the respective associated eigenvectors to retrieve indices of columns that form a linearly dependent set. Note that since $\boldsymbol{\alpha}_A \approx \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$, the column indices of this matrix correspond to the column indices of J which in turn correspond to the indices of constraints in the working set, so that the i-th column of $\boldsymbol{\alpha}_A$ corresponds with the

i-th constraint in $\mathcal{W}$. By finding a set of columns that form a linearly dependent set, we have found a set of constraints that can be compared and considered for removal in order to restore linear independence in $\mathcal{W}$.

In order to implement this in practice, we first note that the eigenvalues and eigenvectors will usually not contain entries that can be identified as exactly zero due to numerical inaccuracies, so a pre-selected tolerance value $\epsilon > 0$ is used throughout instead of 0. This value does not appear to be overly sensitive, and in experiments $\epsilon = 10^{-6}$ has been found to work well. The implementation for the rest of the process is largely straightforward. When the subroutine is called, the eigenvalues u_j and eigenvectors $\boldsymbol{w}_j$ of the singular matrix are calculated, and for each eigenvalue with $|u_j| < \epsilon$ (indicating linear dependence, within the selected tolerance) the indices k are collected from within the associated eigenvector where the absolute values satisfy $|[\boldsymbol{w}_j]_k| > \epsilon$ (indicating non-zero, within the tolerance). The normalized constraint violations v_k are compared across the collected indices k, and the constraint associated with the smallest value of v_k is removed from the working set. If necessary, this process is repeated until either the needed matrix is invertible, or else $\mathcal{W}$ is empty. In practice, only one constraint is usually observed being removed at a time.

4.4 Connections between exact and augmented Lagrangians

As already remarked, the basic expression of the exact Lagrangian can in some contexts be treated as an augmented Lagrangian, so it is natural to consider connections between the two. The EL-ES approach proposed in Section 4.3 can similarly be connected with previous implementations of the AL-ES.

In order to motivate the derivation of these connections, we begin with considering how to improve on the results of the AL-ES as described Section 4.2 through re-examination of the justification used [14] for the original multiplier update rule, as presented in Eq. (4.2). There, the update rule for $\boldsymbol{\alpha}$ is tied with updating the penalty coefficient ω in a way that balances terms of the Lagrangian so that the ES is able to make balanced progress in all dimensions of the search space. This balance is expressed as part of a single-step analysis of the evolution strategy's expected behaviour. In the discussion of Section B.4, the augmented Lagrangian is understood

to be an unconstrained function that shares its minimum with a related constrained problem (a GCP) so long as the elements of the penalty term Ω are large enough to ensure locally positive curvature at the optimum. At the same time, the discussions of Sections 3.1 and B.5 portray the choice of penalty terms ω_i as being good step-sizes for applying the method of gradient ascent on the negative dual function $-\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})$ with respect to the Lagrange multipliers. Apparently, updating the penalty coefficient in the AL-ES serves several overlapping purposes, and a good update rule for ω should:

1. modify the Lagrangian function so that the ES can make balanced progress,

2. be large enough to ensure appropriate positive curvature of the Lagrangian function in the search space, and

3. provide a good choice of step-size for updating the Lagrange multipliers $\boldsymbol{\alpha}$.

It is desirable to determine if any of these criteria can be relaxed or removed, so that the impact of the penalty coefficient update rule can be better understood for each component separately. The easiest to relax is perhaps the condition on positive curvature, which can instead be achieved by for instance limiting the objective and constraint functions to being convex. This is because the second-order necessary condition of Eq. (B.17) becomes a sufficient condition whenever f and g are convex, and this in turn guarantees locally positive curvature of the augmented Lagrangian at the constrained optimum for any values of ω_i. In particular, the ordinary (non-augmented) Lagrangian

$$L_0(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}) \tag{4.17}$$

is positive-definite in an open neighbourhood $\mathcal{N}_r(\boldsymbol{x}^*)$ of the optimum and corresponds to the augmented Lagrangian with penalty term $\boldsymbol{\omega}$ chosen to be the zero vector.

As desired, this simplifying assumption results in eliminating the need for ω updates to enforce positive curvature, as well as decoupling the penalty coefficient update from the Lagrange multiplier update; since there is no penalty term included in the Lagrangian $L_0(\boldsymbol{x}, \boldsymbol{\alpha})$, we need to approach the update step for $\boldsymbol{\alpha}$ in a different way. The Lagrange multipliers are themselves defined in terms of the linear basis

of constraint normals active at the optimum, so it seems natural to consider how constraint information can be used. Indeed, as demonstrated in Section 5.2, one motivation for developing the EL-ES is the observed poor performance of the AL-ES on certain linearly constrained sphere problems, even when using CMA, with particular arrangements of the constraints resulting in narrow feasible regions.

The Newton update step for $\boldsymbol{\alpha}$ as given in Eq. (B.33) includes constraint information which could be helpful for devising a new update rule for the Lagrange multipliers, but it requires approximating local derivative information. Using the process given by Eq. (4.11) introduced as part of the EL-ES, and assuming an approximation for the Hessian matrix of f is available, say $\boldsymbol{A}$, then a quasi-Newton update for the multipliers is given by

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \left[\boldsymbol{J}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{J}\right]^{-1} \cdot g(\boldsymbol{x}^{(k)}). \tag{4.18}$$

In the simplest case that the Hessian approximation $\boldsymbol{A} \approx a \cdot \boldsymbol{I}$, then the objective function is locally spherical and the above calculation is greatly simplified as

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + a \cdot \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\right)^{-1} \cdot g(\boldsymbol{x}^{(k)}).$$

As in the EL-ES, this formulation of a quasi-Newton update rule for $\boldsymbol{\alpha}$ conveniently allows for local approximations of necessary terms simply by using constraint function evaluations that will already be performed for the regular ES updates. However, convex objective functions that are not spherical (that is, those functions with Hessian not equal to a scalar multiple of the identity matrix) are unlikely to perform well unless a better approximation is made for the Hessian, or information about the objective function is included through other means.

A similar update rule that does include objective function information can be arrived at by generalizing the single-step argument used to derive the original $(1+1)$-AL-ES update rule for $\boldsymbol{\alpha}$. To do so, we first reproduce the original single-step analysis for the AL-ES.

4.4.1 Single-step analysis for $(1+1)$-AL-ES

Following the argument given as part of the proposal of the AL-ES [14], the objective function

$$f(\boldsymbol{x}) = a\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$$

is spherically symmetric and only a single linear constraint is considered. Thus, the location of the parent candidate solution can be written without loss of generality as the vector $\boldsymbol{x} = [x_1, R, 0, \ldots, 0]$ and the function for the lone (active) constraint as

$$g(\boldsymbol{x}) = bx_1 + c.$$

By writing the coordinates this way, only changes along the x_1 axis will affect constraint violation, and the value R gives the distance from the optimum in the (unconstrained) subspace spanned by the remaining $x_2, \ldots, x_n$ axes.

A single step of the $(1+1)$-ES considers the value of the augmented Lagrangian for the offspring $\boldsymbol{y}$, expressed as

$$L_\omega(\boldsymbol{y}) = L_\omega(\boldsymbol{x} + \sigma\boldsymbol{z})$$

which after expansion gives

$$= a\sum_{i=1}^{n}(x_i + \sigma z_i)^2 + \alpha\left[\sum_{i=1}^{n} b(x_i + \sigma z_i) + c\right] + \frac{\omega}{2}\left[\sum_{i=1}^{n} b(x_i + \sigma z_i) + c\right]^2.$$

The elements of the mutation vector $\boldsymbol{z}$ are sampled independently from a standard normal distribution, and recalling that we can write the parent $\boldsymbol{x}$ in terms of only x_1 and R, the augmented Lagrangian for the offspring becomes

$$L_\omega(\boldsymbol{y}) = L_\omega(\boldsymbol{x}) + 2\sigma a x_1 z_1 + 2R\sigma a z_2 + \alpha\sigma b z_1 + \omega\sigma b(c + bx_1)z_1 + \frac{\omega}{2}\sigma^2 b^2 z_1^2 + \sigma^2 a\sum_{i=1}^{n} z_i^2.$$

This expression can be simplified considerably if we introduce the normalized step size $\sigma^* = \sigma n/R$, then assuming this approaches a finite value in the limit as the

dimension $n \to \infty$ and after collecting terms, we can write

$$L_\omega(\boldsymbol{y}) \approx L_\omega(\boldsymbol{x}) + \frac{2R^2 a}{n}\left[\sigma^* \frac{\Xi}{R} z_1 + \sigma^* z_2 + \frac{\sigma^{*2}}{2}\right] \tag{4.19}$$

where

$$\Xi = x_1 + \frac{b}{2a}\left(\alpha + \omega g(\boldsymbol{x})\right).$$

The $(1+1)$-ES will accept the offspring only when $L_\omega(\boldsymbol{y}) \leq L_\omega(\boldsymbol{x})$, yet examining the expression above reveals that only the first two terms within the square brackets may be non-positive, and their sign depends on the sampled values for z_1 and z_2. In order for the ES to make balanced progress on both the distance from the (active) constraint boundary and in the remaining $n-1$ dimensions, we can therefore conclude that the first and second terms within the square brackets should be of approximately equal magnitude. Thus, $\frac{\Xi}{R}$ should be of unit order magnitude, meaning the magnitude of Ξ should decrease approximately in proportion with the decrease in the distance R from the optimum in the $n-1$ dimensional unconstrained space. Approaching the optimum point implies that $R \to 0$, yet neither term of Ξ does so on its own. In order for rates of decrease to remain proportional, the two terms of Ξ should be of approximately equal magnitude and opposite sign. Since changes in α are already determined by the method of multipliers update rule, this implies in particular that changes to $\omega g(\boldsymbol{x})$ should be approximately proportional to $2a/b$. Given that the constraint function is linear, this is equivalent to desiring

$$\omega \approx \frac{2a}{b^2}.$$

Although the explicit values of a and b are not available to the algorithm, the AL-ES uses an update rule for the penalty coefficient that aims to approximate this value in order to maintain balanced progress of the evolution strategy in both the constrained and unconstrained subspaces.

4.4.2 Single-step analysis for multimembered ES

This analysis can be extended to problems with multiple constraints and applied to evolution strategies beyond the $(1+1)$-ES. Multirecombinative evolution strategies

were analyzed on conical feasible regions by Porter and Arnold [92], and predicted behaviour was shown to match well with experimental results under certain assumptions. In two dimensions, these conical feasible regions align with the narrow feasible regions discussed in Section 5.2, and have a similar structure in higher dimensions. As in both the analysis for conical constraints as well as in Section 4.4.1, it makes sense to consider balanced progress of the evolution strategy in the constrained as well as the unconstrained subspaces of the problem. For Lagrangian functions, this can be expressed in terms of the complementary linear maps given by matrices $\boldsymbol{P}$ and $(\boldsymbol{I} - \boldsymbol{P})$, as defined in Eq. (B.3).

For a constrained problem with potentially multiple constraints, the single step equation for an evolution strategy operating on the augmented Lagrangian is given similar to before by writing the function value for a selected centroid $L(\boldsymbol{x} + \sigma\hat{\boldsymbol{z}}) = L(\boldsymbol{y})$ in terms of the change in parameter values. Let R be the distance from the optimum in the unconstrained subspace of $\mathbb{R}^n$ and normalize the step-size as $\sigma^* = \sigma n/R$. By assuming σ^* approaches a stationary value and taking a second-order Taylor expansion of $L_{\boldsymbol{\omega}}$ for the next selected step $\boldsymbol{y} = \boldsymbol{x} + \sigma\hat{\boldsymbol{z}}$ around the current centroid $\boldsymbol{x}$, we have the single-step equation given by

$$\begin{aligned} L_{\boldsymbol{\omega}}(\boldsymbol{y}) &= L_{\boldsymbol{\omega}}(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^{\mathrm{T}} \cdot \nabla^2_{\boldsymbol{xx}} L_{\boldsymbol{\omega}}(\boldsymbol{x}) \cdot (\boldsymbol{y} - \boldsymbol{x}) \\ &= L_{\boldsymbol{\omega}}(\boldsymbol{x}) + \frac{\sigma^* R}{n} \hat{\boldsymbol{z}}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}) + \frac{\sigma^* R}{n} \hat{\boldsymbol{z}}^{\mathrm{T}} \cdot \nabla^2_{\boldsymbol{xx}} L_{\boldsymbol{\omega}}(\boldsymbol{x}) \cdot \frac{\sigma^* R}{n} \hat{\boldsymbol{z}} \\ &= L_{\boldsymbol{\omega}}(\boldsymbol{x}) + \frac{\sigma^* R^2}{n} \left[\frac{\hat{\boldsymbol{z}}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x})}{R} + \frac{\sigma^*}{n} \hat{\boldsymbol{z}}^{\mathrm{T}} \cdot \nabla^2_{\boldsymbol{xx}} L_{\boldsymbol{\omega}}(\boldsymbol{x}) \cdot \hat{\boldsymbol{z}} \right]. \end{aligned} \tag{4.20}$$

For the evolution strategy to improve in this iteration, we require $L_{\boldsymbol{\omega}}(\boldsymbol{y}) \leq L_{\boldsymbol{\omega}}(\boldsymbol{x})$, implying the bracketed term in the last line of Eq. (4.20) needs to evaluate to a negative value. In the situation where the Hessian of the Lagrangian $L_{\boldsymbol{\omega}}$ is positive-definite, then the second half of the bracketed term will always be positive. The elements of $\hat{\boldsymbol{z}}^{\mathrm{T}}$ are equivalent to those drawn from a weighted sum of standard normal variables, and these determine the sign of the first half of the bracketed term. Note that if we let $\Xi = \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}$, then Eq. (4.20) is similar to Eq. (4.19) given for the simpler case of one linear constraint on the sphere.

We claimed in the introduction of Section 4.4 that a generalization of the single-step

analysis for multiple constraints would lead to a new update rule for $\boldsymbol{\alpha}$, and we arrive at this update now.

Multiplier update for ordinary Lagrangian

Consider the single step equation of Eq. (4.20) in the context of the ordinary Lagrangian in Eq. (4.17). In order to separately consider the progress of the evolution strategy in the constrained and unconstrained subspaces, we can use the orthogonal decomposition of $\nabla_{\boldsymbol{x}} L_0$ by projection matrices as given in Eq. (B.3) so that

$$\begin{aligned}\nabla_{\boldsymbol{x}} L_0 &= \boldsymbol{P} \cdot (\nabla_{\boldsymbol{x}} f + \nabla_{\boldsymbol{x}} g \cdot \boldsymbol{\alpha}) + (\boldsymbol{I} - \boldsymbol{P}) \cdot (\nabla_{\boldsymbol{x}} f + \nabla_{\boldsymbol{x}} g \cdot \boldsymbol{\alpha}) \\ &= (\boldsymbol{P} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{J}\boldsymbol{\alpha}) + (\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f \\ &= \boldsymbol{J}\left(\boldsymbol{J}^{+} \nabla_{\boldsymbol{x}} f + \boldsymbol{\alpha}\right) + (\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f,\end{aligned}$$

where the second line follows by recalling that $\nabla_{\boldsymbol{x}} g = \boldsymbol{J}$ is the Jacobian, and this is unaffected by the projection matrix $\boldsymbol{P}$ so that $\boldsymbol{P}\boldsymbol{J}\boldsymbol{\alpha} = \boldsymbol{J}\boldsymbol{\alpha}$. Using this decomposition in Eq. (4.20), we can write the first half of the bracketed term as

$$\begin{aligned}\frac{\hat{\boldsymbol{z}}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} L_{\boldsymbol{0}}(\boldsymbol{x})}{R} &= \hat{\boldsymbol{z}}^{\mathrm{T}}\Big(\frac{1}{R}\left(\boldsymbol{P} \cdot \nabla_{\boldsymbol{x}} f + \boldsymbol{J}\boldsymbol{\alpha}\right) + \frac{1}{R}(\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f\Big) \\ &= \hat{\boldsymbol{z}}^{\mathrm{T}}\Big(\frac{1}{R}\boldsymbol{J}\left(\boldsymbol{J}^{+} \nabla_{\boldsymbol{x}} f + \boldsymbol{\alpha}\right) + \frac{1}{R}(\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f\Big).\end{aligned} \tag{4.21}$$

Since R measures the distance from the optimum in the unconstrained space, then $R \to 0$ as $\|(\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f\| \to 0$, and in particular

$$\frac{1}{R}\|(\boldsymbol{I} - \boldsymbol{P}) \cdot \nabla_{\boldsymbol{x}} f\|$$

is of approximately unit order of magnitude as the optimum is approached. In order for the evolution strategy to make balanced progress then, the term

$$\boldsymbol{J}\left(\boldsymbol{J}^{+} \nabla_{\boldsymbol{x}} f + \boldsymbol{\alpha}\right)$$

must go to zero along with R so that

$$\frac{1}{R}\boldsymbol{J}\left(\boldsymbol{J}^{+} \nabla_{\boldsymbol{x}} f + \boldsymbol{\alpha}\right)$$

has an approximately unit order of magnitude. In order for the two inner terms to progressively cancel out as the optimum is approached, we must have $\boldsymbol{\alpha} \approx -\boldsymbol{J}^+ \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. The same result is reached by using the first-order condition that $\nabla_{\boldsymbol{x}} L_0(\boldsymbol{x}) = 0$ to give

$$\begin{aligned} 0 &= \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) \cdot \boldsymbol{\alpha} \\ &= \boldsymbol{J}^+ \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) + \boldsymbol{J}^+ \boldsymbol{J} \cdot \boldsymbol{\alpha} \\ &= \boldsymbol{J}^+ \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) + \boldsymbol{\alpha}, \end{aligned} \tag{4.22}$$

where the second line follows by multiplying through by the pseudo-inverse $\boldsymbol{J}^+$ of the Jacobian of g.

4.4.3 Derivation from inexact solutions

Similar conclusions were reached by Miele et al. [87], Haarhoff and Buys [49], and Buys [30], and highlighted by Bertsekas [26], in the context of numerical optimization routines achieving inexact solutions. Rather than expending extra calculations precisely solving for $\boldsymbol{x}(\boldsymbol{\alpha}^{(k)})$ for a given $\boldsymbol{\alpha}^{(k)}$, the idea is to allow points $\boldsymbol{x}^{(k)}$ that are close to stationary within some bounds, say $||\nabla L_\omega(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)})|| < \epsilon$, yet for which the method of multipliers can still proceed.

Recalling that a local minimum requires a point where the derivative of the ordinary Lagrangian L_0 is zero with respect to $\boldsymbol{\alpha}$, we can construct the quadratic function

$$\begin{aligned} Q(\boldsymbol{x}, \boldsymbol{\alpha}) &= (\nabla_{\boldsymbol{x}} L_0)^{\mathrm{T}} (\nabla_{\boldsymbol{x}} L_0) \\ &= \left(\nabla_{\boldsymbol{x}} f + \nabla_{\boldsymbol{x}} g \cdot \boldsymbol{\alpha}\right)^{\mathrm{T}} \left(\nabla_{\boldsymbol{x}} f + \nabla_{\boldsymbol{x}} g \cdot \boldsymbol{\alpha}\right) \end{aligned}$$

that measures the "error" in estimation of the optimum, and which will obviously equal 0 with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*,\ \boldsymbol{x} = \boldsymbol{x}^*$ by Theorem B.6. Taking the derivative of this function with respect to $\boldsymbol{\alpha}$ and solving gives

$$\begin{aligned} 2\nabla_{\boldsymbol{x}} g^{\mathrm{T}} \nabla_{\boldsymbol{x}} f + 2\nabla_{\boldsymbol{x}} g^{\mathrm{T}} \nabla_{\boldsymbol{x}} g \cdot \boldsymbol{\alpha} &= 0 \\ \boldsymbol{\alpha} &= -\left(\nabla_{\boldsymbol{x}} g^{\mathrm{T}} \nabla_{\boldsymbol{x}} g\right)^{-1} \nabla_{\boldsymbol{x}} g^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} f. \end{aligned} \tag{4.23}$$

Note that this expression matches the one given in Eq. (4.22). If the solution is exact, then both $||\nabla L_\omega(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)})|| = 0$ and $\boldsymbol{x}^{(k)} = \boldsymbol{x}(\boldsymbol{\alpha}^{(k)})$, and Eq. (4.23) then reduces to the usual multiplier update of Eq (3.6). However, when the solution $\boldsymbol{x}^{(k)}$ only satisfies the stationary condition $||\nabla L_\omega(\boldsymbol{x}^{(k)}, \boldsymbol{\alpha}^{(k)})||$ within the $\epsilon > 0$ bound, the usual procedure for the method of multipliers is no longer appropriate. In this case, Eq. (4.23) gives the proper correction to the multiplier update.

4.4.4 Summary and resulting exact Lagrangian

From the above discussion, we have arrived at an ordinary Lagrangian given by

$$L_0(\boldsymbol{x}) = f(\boldsymbol{x}) + \left(-\left(\nabla_{\boldsymbol{x}} g^{\mathrm{T}} \nabla_{\boldsymbol{x}} g\right)^{-1} \nabla_{\boldsymbol{x}} g^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} f \right)^{\mathrm{T}} g(\boldsymbol{x}) \tag{4.24}$$

which combines Eq. (4.17) and update step Eq. (4.23) into a single expression. Despite having eliminated the penalty coefficient $\boldsymbol{\omega}$, Eq. (4.22) suggests the Lagrangian function will be modified such that the ES can make balanced progress, and Eq. (4.23) suggests that $\boldsymbol{\alpha}$ will be a good approximation to the optimal Lagrange multipliers. There is no augmenting penalty term, but for fully convex problems we might expect this approach to suffice; however, this may not be the case. As noted by both Fletcher [43] and Bertsekas [27], if we consider a problem defined with quadratic objective function f having Hessian $\boldsymbol{A}$ and linear constraints g with Jacobian $\nabla_x g = \boldsymbol{J}$, then the Hessian of the ordinary Lagrangian in Eq. (4.24) can be written as

$$\begin{aligned} \nabla^2_{\boldsymbol{xx}} L_0 &= \boldsymbol{A} - \boldsymbol{A}\boldsymbol{J}(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}} - \boldsymbol{J}(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{A} \\ &= \boldsymbol{A}\left(\boldsymbol{I} - \boldsymbol{P}\right) - \boldsymbol{P}\boldsymbol{A} \\ &= (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{P}) - \boldsymbol{P}\boldsymbol{A}\boldsymbol{P} \end{aligned}$$

where the last line follows by using the definition of Eq. (B.3) and then completing the square with respect to $(\boldsymbol{I} - \boldsymbol{P})$. This shows that the Hessian of the ordinary Lagrangian has the same curvature as the objective f in the unconstrained directions, but *opposite curvature* in the subspace spanned by the constraint normals. Therefore, setting $\boldsymbol{\alpha}$ according to Eq. (4.23) will give a Lagrangian function with an appropriate

unconstrained minimum if and only if the curvature of f is negative exactly in the directions of the constraint normals and positive elsewhere. In the simplest case of minimizing on the sphere with linear constraints, as with many other convex problems, this is obviously not the case.

The solution is to once again include an augmenting penalty term, and doing so results in Eq. (3.14) and the subsequent exact Lagrangian method of Section 3.2. The penalty term ω is once again responsible for ensuring locally positive curvature, but now in a different way. While the penalty term (and thus the step size of the multiplier update) needed to increase for the method of multipliers and related AL-ES approaches of Section 4.2 in order to balance against less positive curvature of f in the directions of the constraint normals, the penalty term for the exact Lagrangian method needs to increase in order to balance against *more* positive curvature of f in those same directions.

An instructive visualization of this effect is given in Figure 4.1 for the TR2 sphere problem having objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ (blue contour lines) and both the infeasible region (shaded grey) and constraint boundary (dashed lines) for the linear inequality constraint $2 - x_1 - x_2 \leq 0$. The exact Lagrangian defines $\boldsymbol{\alpha}(\boldsymbol{x})$ in terms of position $\boldsymbol{x}$ in the search space, so the only parameter is ω. Shaded contour regions are given for three exact Lagrangian functions $\phi(\boldsymbol{x})$ when using a roughly appropriate value of $\omega = 2$ (top right), a value of $\omega = 2 \cdot 10^{-2}$ that is comparatively very small (bottom left), and a value of $\omega = 2 \cdot 10^{2}$ that is very large. For the roughly appropriate value, the contours for the exact Lagrangian are similar to the circles seen for $f(\boldsymbol{x})$ and have a minimum corresponding with the constrained optimum at $\boldsymbol{x}^* = [1, 1]$. When the value of ω is very small, it is insufficient to maintain positive curvature in the directions of the positive and negative constraint normals, and the constrained optimum becomes a saddle point for $\phi(\boldsymbol{x})$. When the value of ω is very large, the curvature is locally positive and the minimum again corresponds with the constrained optimum but with increased ill-conditioning in $\phi(\boldsymbol{x})$. Compare with Figure 1.2 which shows the same effects on changing parameters for the augmented Lagrangian $L_\omega(\boldsymbol{x}, \alpha)$.

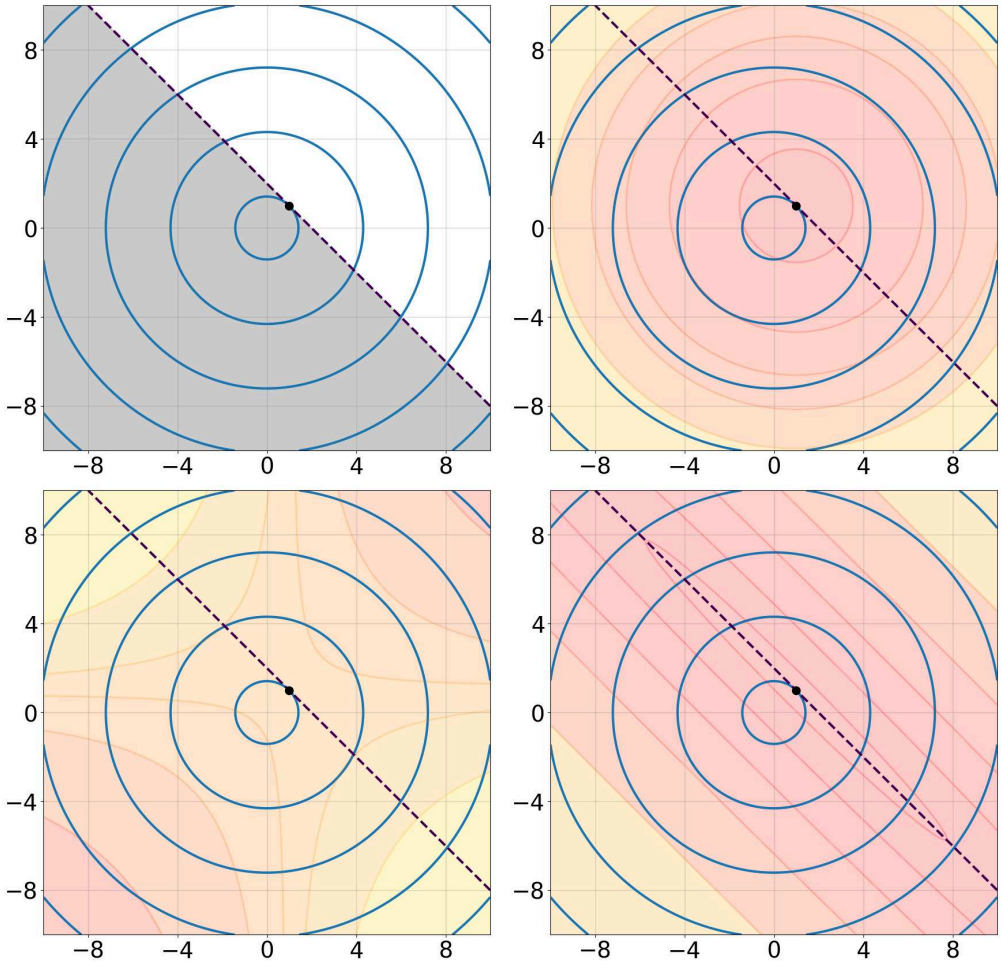

Figure 4.1: Visualization in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with inequality constraint $g(\boldsymbol{x}) = 2 - x_1 - x_2 \leq 0$. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2$. Bottom left: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{-2}$. Bottom right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{2}$. The constrained optimum is marked throughout at $\boldsymbol{x}^* = [1, 1]$.

Chapter 5

Experimental evaluation

This chapter evaluates the proposed exact Lagrangian evolution strategy (EL-ES) by comparing its performance with that of existing constraint-handling methods using data collected experimentally. Comparisons are split into three main problem sets, consisting first of archetypal problems formed by combining sphere and ellipsoid objective functions with linear constraints, followed by a selection of commonly used constrained optimization benchmark problems from the literature, and finally using a recently proposed scalable benchmark problem with multiple linear constraints. We will demonstrate by these results that the EL-ES outperforms the augmented Lagrangian approach (AL-ES) on all problems considered. The improvement resulting from using the exact Lagrangian approach is also significant enough that the EL-ES (without CMA) will be seen to outperform the AL-CMA-ES on a majority of the selected problems while still being closely competitive on the others.

Section 5.1 introduces experimental criteria that will be used for evaluating the performance of the different algorithms. Certain concerns are highlighted with how best to make comparisons between approaches. Importantly, we summarize the concept of an empirical cumulative distribution function (ECDF) that will be used to plot and visually compare performance results throughout this chapter, including how target sets are defined and evaluated against multiple runs on a chosen problem.

In Section 5.2, spherical and ellipsoidal objective functions are combined with linear constraints and evaluated against. The particular case of a sphere with constraints that form a narrow feasible region (NFR) highlights a situation found difficult by AL-ES implementations, even when including CMA. Linear constraints of random orientation are also considered, generated in such a way that each constraint will be active and have a positive Lagrange multiplier.

In Section 5.3, benchmark problems from the literature are selected and used to demonstrate that the EL-ES is also applicable to problems with various combinations of linear and non-linear features. As these problems are all commonly used to benchmark the performance of constrained optimization algorithms, this places the EL-ES among a variety of published results.

In Section 5.4, the linear Rotated Klee-Minty problem is considered which is scalable in both dimension and number of linear constraints. Performance comparisons are made between the EL-ES and AL-CMA-ES as previously described and the ϵMA-ES and lcCMSA-ES. Both of the latter algorithms have published competitive results when comparing on the Rotated Klee-Minty problem as well as other problem sets, and so serve as useful comparative benchmarks of performance for the Lagrangian approaches.

5.1 Methods of comparison

Problem benchmarks such as those used in competitions from the IEEE Congress on Evolutionary Computation (CEC) [77, 82, 129] aim to rank algorithms by comparing solution quality under fixed budgets of function evaluations. The results can be difficult to compare and extrapolate [62], and Hansen et al. [55] argue that comparisons on fixed budgets are not in general usefully interpretable: analyzing quantitative relationships between quality indicators (such as observing that one metric is twice as small as another) need not indicate a similar relationship between the algorithms used to reach them. Instead, Hansen et al. advocate comparing the number of function evaluations needed by each algorithm to reach a set of fixed targets. This approach forms part of the **Co**mparing **C**ontinuous **O**ptimizers (COCO) benchmark [57].

An important method of comparison between algorithms for COCO relies on empirical cumulative distribution functions (ECDFs) that can be plotted as visualizations. These are in turn a generalization of single-target data profiles [88], which aggregate multiple runs of an algorithm and yield the proportion of those runs meeting a fixed target on a specified optimization problem after an elapsed measure of runtime. Targets are usually chosen to be the distance from a known optimal value, and runtime is typically measured in function evaluations or iteration count. Targets are evaluated

against a chosen function used through the algorithm's operation (such as the objective function), and are considered met if the value returned by the function does not exceed the target value. Once a target is met, the entirety of the run is considered as successfully meeting the target, and the associated runtime for a successful target is the earliest runtime for which the target was met. A data profile, or equivalently a single-target ECDF, therefore measures the experimental success rate on the chosen target with respect to runtime. Plotting a single-target ECDF as a graph such as in Figure 5.1, with runtime on the x-axis and proportion of successful runs on the y-axis, gives a curve that visually represents the range of performance on the selected problem: runs that meet the target with relatively few function evaluations form the left portion of the curve, while runs that take relatively more function evaluations (indicative of worst-case performance) form the right portion. With enough runs to give a representative sample, the extremes of the curve are thus indicative of best-case and worst-case performance respectively, with the slope of the curve correlating inversely with variance in performance between successful runs. In the example given by Figure 5.1, the observed best-case performance is seen to correspond with meeting the fixed target using fewer than $10^{2.5}$ function evaluations, while worst-case performance corresponds with taking just over $10^{3.0}$ evaluations.

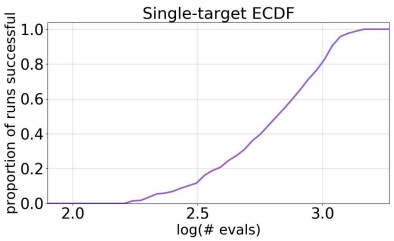

Figure 5.1: Example of a single-target ECDF plot showing proportion of successful runs (y-axis) for a single algorithm with respect to function evaluations (x-axis) scaled logarithmically.

It is straightforward to generalize this idea to runtime ECDFs for multiple targets by yielding the proportion of a *target set* that has been met after an elapsed measure of runtime across a set of runs on a problem. The target set constitutes a sequence of fixed targets, usually of increasing difficulty. Evaluating a single run with respect to a fixed runtime as in a single-target ECDF, the number of targets met from the sequence is a measure of its performance. Extending this to evaluate across a set of

R runs in aggregate with respect to a fixed runtime, if we write the target set T as having size $|T|$, and $|T_i|$ as being the count of targets met by the i-th run from the set, then the proportion of met targets across all runs is expressed as the ratio

$$\frac{1}{R}\sum_i \frac{|T_i|}{|T|} = \frac{\sum_i |T_i|}{R \cdot |T|}.$$

This can be thought of as the average proportion of targets achieved across all R runs, or as the total number of targets achieved from within the total set of $|T| \cdot R$ targets as considered across all runs. Plotting an ECDF for multiple targets as a graph with runtime on the x-axis and proportion of met targets on the y-axis gives a curve that represents algorithm performance in a manner analogous to that of the single-target data profile: the left portion of the curve indicates performance on easier targets, the right portion indicates performance on more difficult targets, and the slope correlates inversely with variance in convergence speed between targets.

Generating ECDFs for constrained optimization problems introduces extra complexity, since targets can then reasonably be defined for the objective function as well as for each of the constraints. It is therefore necessary to either consider target sets separately for constraints and objective, or else combine them in a way that gives meaningful results.

5.1.1 Target definitions

The latest COCO benchmark for single-objective constrained optimization recommends[1] the use of 41 targets defined as $t_i = f(\boldsymbol{x}^*) + 10^{e_i}$ with exponents e_i evenly distributed in the closed interval $[2, -6]$. For a constrained optimization problem (GCP), these are evaluated against the combined function

$$\tilde{f}(\boldsymbol{x}) = \max[f(\boldsymbol{x}^*), f(\boldsymbol{x})] + \sum_{i=0}^{m} \max[0, g_i(\boldsymbol{x})] \tag{5.1}$$

so that a run is successful on target t_i in iteration k if $\tilde{f}(\boldsymbol{x}^{(k)}) \leq t_i$. This additively combines a measure of success for the objective function with a measure of

[1] Taken from the COCO outline for bbob-constrained at `http://numbbo.github.io/coco-doc/bbob-constrained/`, retrieved Apr 25, 2022.

feasibility.

A related approach is used by Hellwig et al. [63] and Spettel et al. [117] that instead defines targets separately for objective and constraints. Respectively, these are two sequences of values defined as

$$t_i^f = f(\boldsymbol{x}^*) + 10^{e_i}, \qquad t_j^g = g(\boldsymbol{x}^*) + 10^{e_j} \tag{5.2}$$

with distinct exponents evenly distributed in the closed intervals $e_i \in [0, -8]$ for t_i^f and $e_j \in [2, -6]$ for t_j^g. The set of constraint targets additionally contains the value 0. The objective targets are simply evaluated against $f(\boldsymbol{x})$ while the constraint targets are evaluated against the sum of violated constraints

$$g_\Sigma(\boldsymbol{x}) = \sum_{i=0}^{m} \max[0, g_i(\boldsymbol{x})]. \tag{5.3}$$

A run is therefore successful in its k-th iteration on the i-th f-target if $f(\boldsymbol{x}^{(k)}) \leq t_i^f$ and successful on the j-th g-target if $g_\Sigma(\boldsymbol{x}^{(k)}) \leq t_j^g$. The runtime for a successful run is measured as number of function evaluations consumed by the algorithm, either for f or g, up to the first successful iteration.

Both of these approaches have their drawbacks. The sum defined in Eq. (5.1) and used by COCO obfuscates the distinction between convergence to the feasible region and convergence to the optimal objective function value. The use of Eq. (5.3) can also be problematic, both because it may contain spurious information (a constraint being violated which is inactive at the optimum may be an irrelevant feature of an algorithm's progress towards that optimum) and because its information is "lossy" (initialization within, or a single step made into, the feasible region is enough to universally satisfy all g-targets for the remainder of a run). Recall that the runtime for a target is evaluated according to the first iteration in which that target is met, so that if $g_\Sigma(\boldsymbol{x}^{(k)}) = 0$ in iteration k, then clearly $g_\Sigma(\boldsymbol{x}^{(k)}) \leq t_j^g$ will also be satisfied for all j indexing the g-target set.

I propose an alternative to address these concerns, which is to instead define the (ℓ_1)

active constraint distance

$$g_{\mathcal{A}}(\boldsymbol{x}) = \sum_{i \in \mathcal{A}} |g_i(\boldsymbol{x})| \tag{5.4}$$

as the sum of absolute values of constraint function values $g_i(\boldsymbol{x})$ limited to those indexed by the optimal active set $\mathcal{A}$. When used to evaluate against a set of g-targets near the optimum, Eq. (5.4) approaches the same value as Eq. (5.3), and using either equation there is equivalently appropriate. However, for evaluating g-targets farther from the optimum, the active constraint distance gives a more meaningful value that is not strictly a measure of feasibility. In particular, algorithms are not rewarded for remaining feasible with respect to constraints that become irrelevant in a neighbourhood of the optimum. One possible drawback with this alternative is that the active constraint distance becomes a meaningless measure if there are no constraints active at the optimum; however, this situation is not a common test case.

5.1.2 Staggered ECDFs

Considering separate target sets on even a moderate number of problems can lead to ECDF plots that contain all of the relevant information, but are difficult to interpret quickly. As an alternative, I will primarily use *staggered* ECDFs that visually represent combined performance by showing success on the full set of f-targets together with one of two fixed g-target values. An example is given in Figure 5.2 with the proportion of met targets plotted against the count of $(f + g)$ evaluations. Runs on the first target set (shown in solid lines) are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_\Sigma \leq 10^0$ are satisfied, while runs on the second target set (shown in dashed lines) are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_\Sigma \leq 10^{-6}$ are satisfied. By using fixed g-targets staggered at roughly opposing edges of the difficulty range, a clear picture is given for the two extremes of algorithm performance while sacrificing minimal detail.

Additional figures beyond the staggered ECDFs show progress on the full range of f-targets and the full range of g-targets. As in the example given by Figure 5.3, these regular ECDF plots are grouped together by problem and arranged into pairs of rows

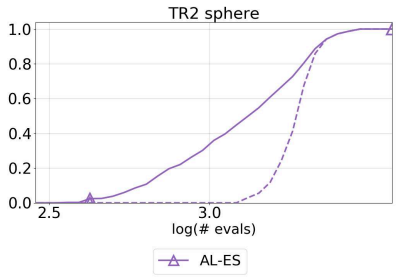

Figure 5.2: Example of a staggered ECDF plot containing a single pair of curves for the targets met by one algorithm.

representing the proportion of successful f-targets plotted against the count of f-evaluations (top) and successful g-targets plotted against the count of g-evaluations (bottom). The horizontal axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets. In the example figure, one pair of rows is given, labeled for the TR2 sphere problem.

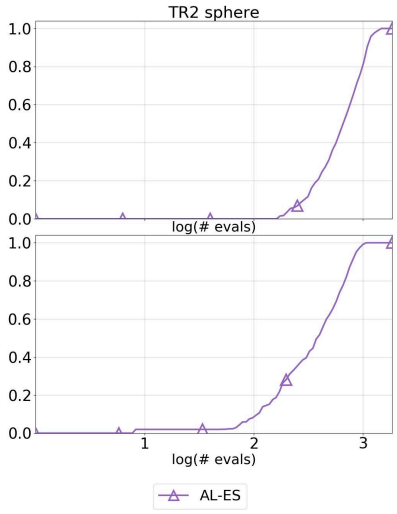

Figure 5.3: Example of ECDF plots paired vertically by problem (indicated by the label) showing f-evals vs. f-targets (top plot of pair) and g-evals vs. g-targets (bottom plot of pair).

5.2 Spheres and ellipsoids

Spheres and ellipsoids constitute a class of functions with search space features that are relatively simple to describe. In n-dimensional space, these can be parameterized

as the sphere function

$$f_{\text{sph}}(\boldsymbol{x}) = a \cdot \sum_{i=1}^{n} x_i^2 \tag{5.5}$$

with single coefficient a, and generalized to the axis-aligned ellipsoid function

$$f_{\text{gen}}(\boldsymbol{x}) = \sum_{i=1}^{n} a_i x_i^2 \tag{5.6}$$

with n coefficients a_i not all equal. Common examples of the latter with single parameters $\xi > 1$ include the discus function

$$f_{\text{dis}}(\boldsymbol{x}) = \xi x_1^2 + \sum_{i=2}^{n} x_i^2$$

and cigar function

$$f_{\text{cig}}(\boldsymbol{x}) = x_1^2 + \xi \sum_{i=2}^{n} x_i^2,$$

while an ellipsoid with varying parameters is

$$f_{\text{ell}}(\boldsymbol{x}) = \sum_{i=1}^{n} \xi^{\left(\frac{i-1}{n-1}\right)} x_i^2. \tag{5.7}$$

It will be noted that each of these objective functions as given are highly separable, in that the optimal value of the i-th coordinate does not depend on the chosen values for other coordinates. However, the selected algorithms to be evaluated are all additionally invariant to rotations of the coordinate system, so the results will be unaffected. Adding linear constraints to any of these functions gives a simple constrained optimization problem, and different variations have been used to evaluate augmented Lagrangian ES approaches both without [14, 16, 18] CMA and with [17, 38].

5.2.1 Fixed constraints

It is argued by Arnold and Porter [14] that any effective constraint-handling technique for evolution strategies should necessarily be able to achieve log-linear convergence on

convex quadratic problems subject to a single constraint. In this section, we demonstrate that this is a feature of the EL-ES by performing experimental comparisons on a sphere and moderately conditioned ellipsoid. Using these problems has the additional benefit of allowing comparisons with other published results for evolution strategies in the literature. We further consider a specially constructed type of linearly constrained sphere with two constraints that form a narrow feasible region (NFR) and show that this type of problem poses difficulties for existing AL-ES implementations, even when using CMA, but on which the EL-ES is able to converge effectively to the optimum.

To begin, a set of problems is generated by combining objective functions with fixed constraints, resulting in multiple instances of ICPs. Both the unit sphere function

$$f_{\text{sph}} = \sum_{i=1}^{n} x_i^2$$

and ellipsoid function f_{ell} as in Eq. (5.7) are used, with $\xi = 10$ giving moderate conditioning for the ellipsoid. In both problems, a single linear inequality constraint function

$$g_1(\boldsymbol{x}) = \boldsymbol{b}_1^{\mathrm{T}}\boldsymbol{x} + c_1 \leq 0$$

is used with $\boldsymbol{b}_1 = -[1, 0]$ and $c_1 = 1$. Thus the constraint boundary is orthogonal to the x_1 axis, and the optimal point is located at $\boldsymbol{x}^* = [1, 0]$. With respect to the isotropic sphere function f_{sph}, this is equivalent (up to a rotation and shift of the constraint boundary, centered on the origin) to the TR2 sphere problem introduced by Kramer and Schwefel [72] and used by both Arnold and Hansen [6] and Dufossé and Hansen [38], for which $\boldsymbol{b}_1 = -[1, 1]$ and $c_1 = 2$ with optimal point at $\boldsymbol{x}^* = [1, 1]$.

To highlight performance on a problem found difficult by the standard AL-ES, an additional sphere problem for $m = 2$ is constructed with a narrow feasible region (NFR). The first constraint is fixed using g_1 as above, and the second constraint generated by negating the normal vector $\boldsymbol{b}_1$ then rotating by $\frac{-1}{100}$-th of a right angle about the origin; equivalently, rotating $\boldsymbol{b}_1$ by $\pi\left(1 - \frac{1}{200}\right)$ radians. The resulting constraint

normal vectors are

$$\boldsymbol{b}_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{b}_2 = -\begin{bmatrix} \cos\left(\pi\left(1 - \frac{1}{200}\right)\right) \\ \sin\left(\pi\left(1 - \frac{1}{200}\right)\right) \end{bmatrix}$$

defining constraint functions

$$g_1(\boldsymbol{x}) = \boldsymbol{b}_1^{\mathrm{T}}\boldsymbol{x} + c_1 \leq 0$$
$$g_2(\boldsymbol{x}) = \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{x} + c_2 \leq 0$$

with $c_1 = c_2 = 1$. Note that both constraints are active at the optimum with non-zero Lagrange multipliers.

Figures 5.4 and 5.5 give visualizations of the NFR sphere problem, with scaled axes to highlight relevant details, using both augmented and exact Lagrangians. The plots display contours of the objective function $f(\boldsymbol{x})$ (blue lines) as well as both the infeasible region (shaded grey) and constraint boundaries (dashed lines) for the two inequality constraints. Shaded contour regions are given for three augmented Lagrangian functions in Figure 5.4 defined by using the optimal $\boldsymbol{\alpha}^*$ and unit $\omega = 1$ (top right), by increasing the penalty coefficient ω by a factor of 20 (bottom left), and by increasing the Lagrange multipliers $\boldsymbol{\alpha}$ by a factor of 20 (bottom right).

For optimal $\boldsymbol{\alpha}$ and unit ω, the resulting augmented Lagrangian is well-conditioned and its unconstrained minimum corresponds with the constrained optimum at $\boldsymbol{x}^* \approx [1, 127.321]$. However, as ω increases the ill-conditioning also significantly increases, and non-optimal values for $\boldsymbol{\alpha}$ move the unconstrained minimum far from the constrained optimum. Perturbations in either of these Lagrangian parameters will have significant impacts on the underlying augmented Lagrangian.

Similar shaded contour regions are given in Figure 5.5 for three exact Lagrangian functions defined by using $\omega = 2$ (top right), smaller $\omega = 2 \cdot 10^{-2}$ (bottom left), and larger $\omega = 2 \cdot 10^{2}$ (bottom right). The values $\omega = 2$ and $\omega = 2 \cdot 10^{2}$ both give exact Lagrangians with almost no ill-conditioning (compare also with Figure C.2 in the appendix, which uses equal scaling for both axes) and an unconstrained minimum corresponding to the constrained optimum $\boldsymbol{x}^*$. The value of $\omega = 2 \cdot 10^{-2}$ however is

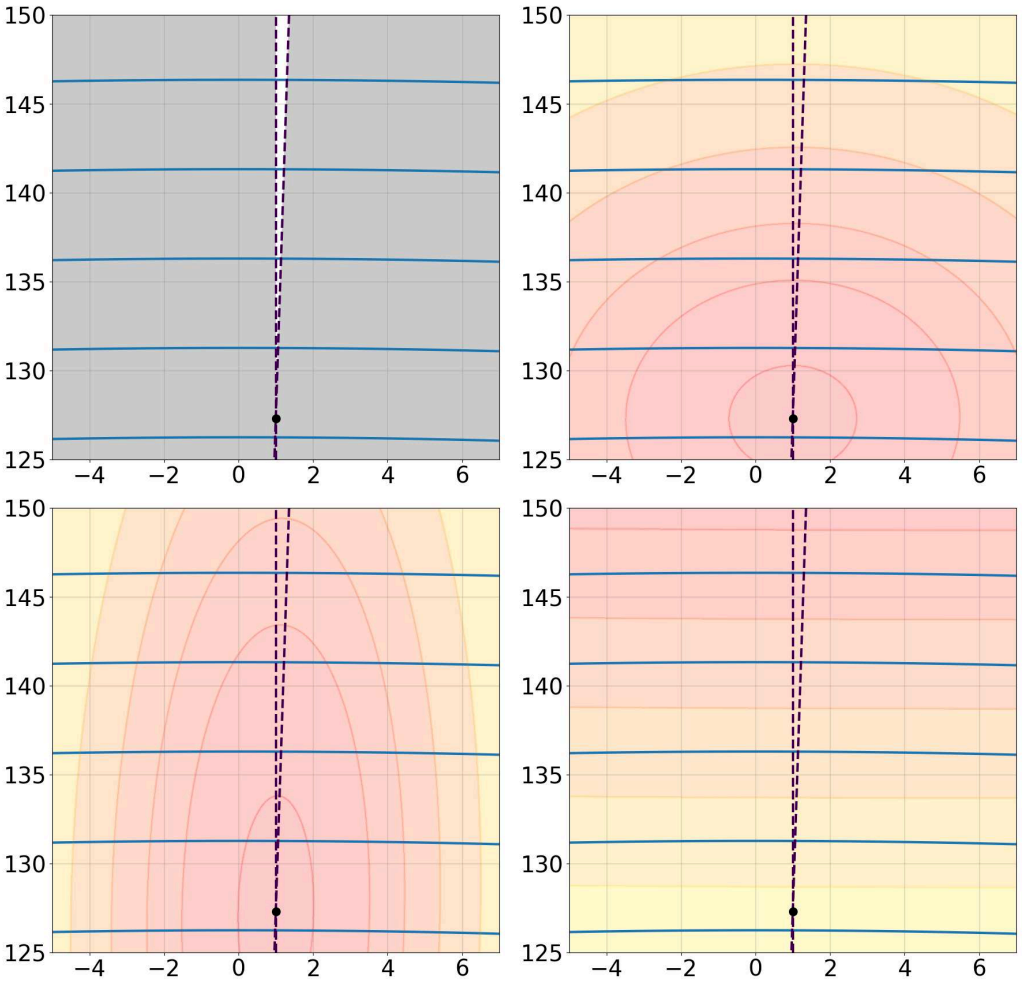

Figure 5.4: Visualization in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with NFR inequality constraints. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*,\ \omega = 1$. Bottom left: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*,\ \omega = 20$. Bottom right: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = 20\boldsymbol{\alpha}^*,\ \omega = 1$. The constrained optimum is marked throughout at $\boldsymbol{x}^* \approx [1, 127.321]$. Figure C.1 gives a similar version with equal axis scaling.

not large enough to ensure locally positive curvature, and the resulting Lagrangian has an unconstrained maximum at $\boldsymbol{x}^*$.

With simple, convex quadratic objective functions and linear constraints all active, each of the sphere, ellipsoid, and NFR problems as given above can be described entirely by their respective Hessian and Jacobian matrices $\boldsymbol{H}$ and $\boldsymbol{J}$, and the objective

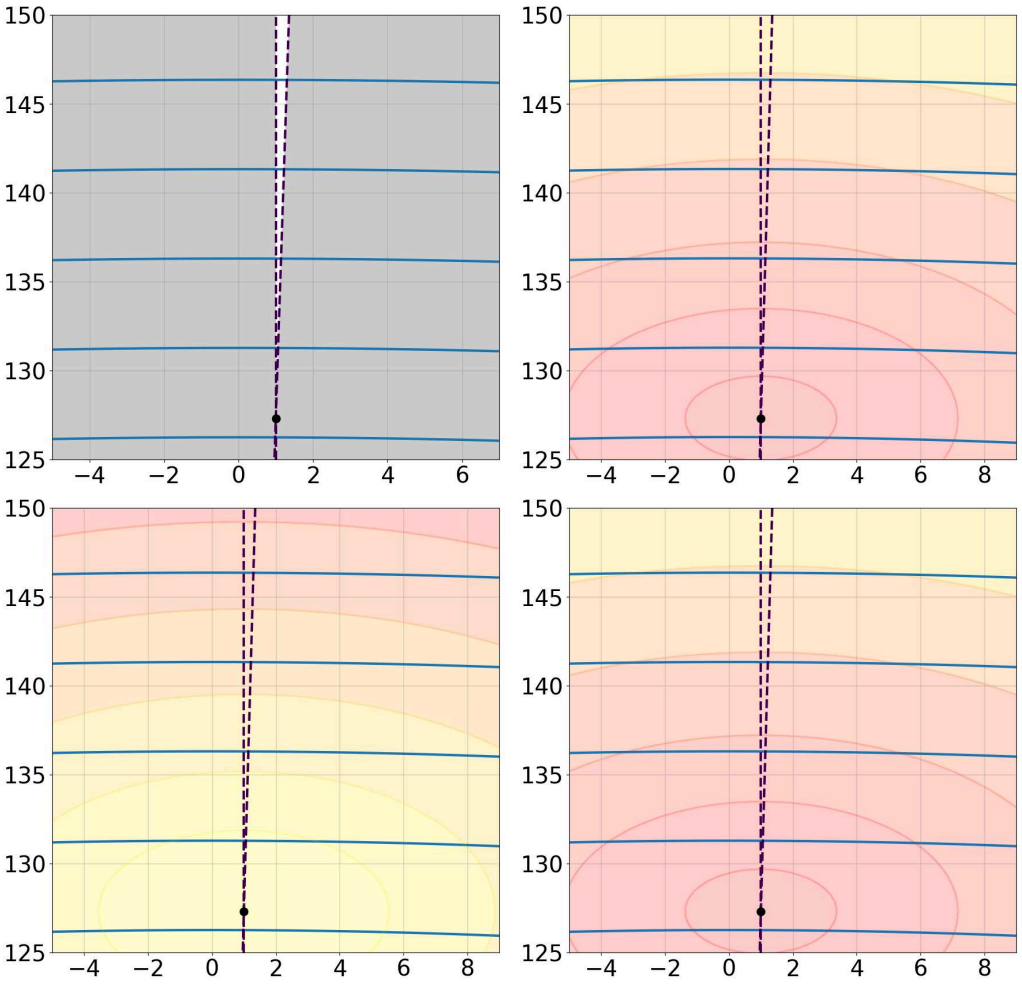

Figure 5.5: Visualization in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with NFR inequality constraints. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2$. Bottom left: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{-2}$. Bottom right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{2}$. The constrained optimum is marked throughout at $\boldsymbol{x}^* \approx [1, 127.321]$. Figure C.2 gives a similar version with equal axis scaling.

and constraint functions can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}, \\ g(\boldsymbol{x}) &= \boldsymbol{J}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c} \leq 0. \end{aligned} \tag{5.8}$$

By combining these definitions with the first-order necessary conditions given in Eqs. (B.15) and (B.16), we can give analytic descriptions of the optimal KKT pair

as

$$\begin{aligned}
\boldsymbol{x}^* &= -\boldsymbol{H}^{-1}\boldsymbol{J} \cdot \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{J}\right)^{-1} \cdot \boldsymbol{c}, \\
\boldsymbol{\alpha}^* &= \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{J}\right)^{-1} \cdot \boldsymbol{c} \\
&= -\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{J}\right)^{-1} \boldsymbol{J}^{\mathrm{T}}\boldsymbol{x}^* \\
&= -\boldsymbol{J}^{+}\boldsymbol{H}\boldsymbol{x}^*.
\end{aligned}$$

The equivalence of the last line for $\boldsymbol{\alpha}^*$ follows either by using identities given by Fletcher [43] for deriving the optimal KKT pair for the method of multipliers, or by relying on the identity

$$\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{J}\right)^{-1} = \left(\boldsymbol{J}^{+}\boldsymbol{H}(\boldsymbol{J}^{\mathrm{T}})^{+}\right)$$

which can be verified through explicit calculation.

Experimental results

These three problems consisting of the $m = 1$ sphere, $m = 1$ ellipsoid, and $m = 2$ NFR sphere are experimentally tested with problem dimensions of both $n = 2$ and $n = 20$, creating six problems in total. To begin with, the problems are used as defined and with no modifications, then later in Section 5.2.1 variations are considered with changes to the scaling between the objective and constraint functions.

Data sets are generated for each problem by performing 25 runs on each of four algorithms. The **aCMA-ES** [6] is a $(1 + 1)$ evolution strategy that uses a covariance matrix for generating offspring which is actively updated away from constraint violations. The **AL-ES** is a $(\mu/\mu_W, \lambda)$ evolution strategy that follows the outline given in Section 4.2 and otherwise implements Algorithm 2.2, which is the implementation given by Atamna et al. [18, 19] using parameter settings suggested by Dufossé and Hansen [38]. The **AL-CMA-ES** is also a $(\mu/\mu_W, \lambda)$-ES, but with CMA used instead for offspring generation following the outline and parameter recommendations of Section 4.2 and otherwise implementing Algorithm 2.3. Duffosé and Hansen [38] apply surrogate modeling to an augmented Lagrangian approach with CMA-ES, and

additionally implement AL-CMA-ES without surrogate modeling in order to investigate improved parameter settings. Their implementation of AL-CMA-ES without surrogate modeling largely matches the implementation given in Algorithm 2.3.

Population parameters for each algorithm are set according to Eq. (2.11). Offspring weights are generated numerically for AL-ES according to the recommendations of Arnold [7, 8] for infinite-dimensional spheres, matching the implementation of Atamna et al., while the AL-CMA-ES uses the default weights (including negative weights) recommended for CMA-ES, matching the implementation of Dufossé and Hansen. Finally, the **EL-ES** is the exact Lagrangian approach proposed in Section 4.3. Other than the EL-ES, the performance of each chosen algorithm has been previously studied on either or both of the $m = 1$ sphere and ellipsoid problems, providing a point of comparison with results in the literature.

Runs are terminated only when $f(\boldsymbol{x})$ and $g_{\mathcal{A}}(\boldsymbol{x})$ are both within 1.0e-8 of optimum values, given by $f(\boldsymbol{x}^*)$ and $g_{\mathcal{A}}(\boldsymbol{x}^*) = 0$, respectively. Starting points are initialized randomly using coordinates drawn uniformly from the interval $[-10, 10]$, with the exception of the aCMA-ES which requires a feasible starting point. In order to accommodate a reasonable comparison, a set of feasible points is generated for each problem in a pre-processing step similar to [38] by using CMA-ES to minimize the sum of constraint violations function $g_{\Sigma}(\boldsymbol{x})$ in a series of 200 runs. The result is a set of 200 feasible points within the search space for each problem and an associated count of g-evaluations used to find each point. This set is then sampled from uniformly at random in order to initialize each run of the aCMA-ES, and the count of g-evaluations is initialized to the number used to locate the feasible starting point.

Runtimes throughout are measured as evaluation counts for the objective function f or constraint function g, or as a sum of both; these are referred to respectively as f, g, and $(f + g)$ evaluations (or evals). In the given ECDF plots, the base-10 logarithm of the count of function evaluations is used as the unit for the x-axes. On all problems, only a single constraint evaluation is considered needed to return $g_i(\boldsymbol{x})$ for all i.

Convergence plots are given in Figure 5.6 showing the distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ from the constrained optimum and in Figure 5.7 showing the step size σ for all algorithms, both

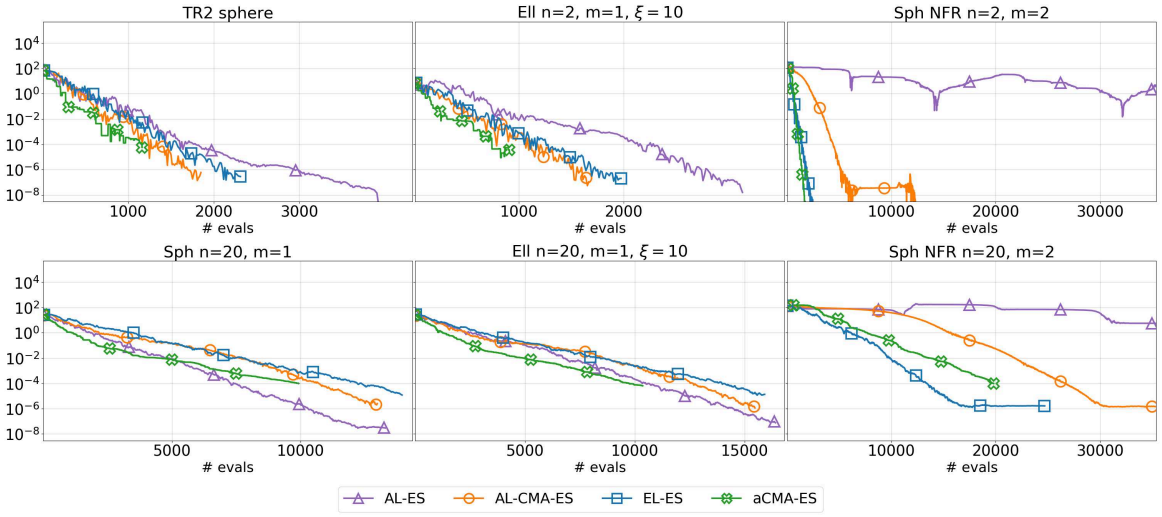

Figure 5.6: Convergence plots showing distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ from the constrained optimum with respect to the first 3.5×10^4 $(f + g)$-evals for median runs from each of four algorithms. The x-axes are scaled to present as much detail as possible without obscuring relevant data points.

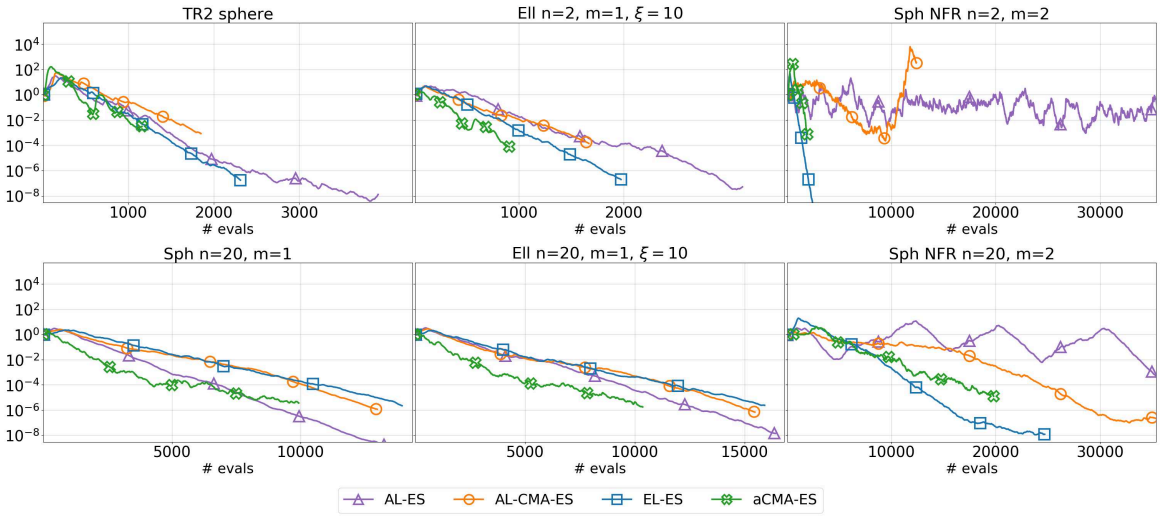

Figure 5.7: Convergence plots showing step size σ with respect to the first 3.5×10^4 $(f + g)$-evals for median runs from each of four algorithms. The x-axes are scaled to present as much detail as possible without obscuring relevant data points.

plotted against the total number of $(f+g)$ function evaluations. Figure 5.8 additionally shows the normalized distance $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$ between the current approximation of $\boldsymbol{\alpha}$ and the optimal Lagrange multiplier vector $\boldsymbol{\alpha}^*$ for the three Lagrangian methods. Each of the plots is generated from median runs of the corresponding 25 runs set, and displays behaviour only across the first 3.5×10^4 total function evaluations in order to highlight relevant details. The problems on the NFR sphere are made evident, in particular for AL-ES and AL-CMA-ES on the $n = 2$ variant in Figures 5.7 where the adapted step sizes are erratic and far too large. After an initialization period for each problem in Figure 5.8, all three Lagrangian algorithms appear to exhibit log-linear convergence of the Lagrange multiplier vector on the $m = 1$ sphere and ellipsoids problems. The EL-ES appears to converge with slightly fewer overall function evaluations compared to the other two algorithms. This difference is much more pronounced for the NFR spheres, where the EL-ES convergence is notably faster, and the AL-ES approximation is seen to be overall very poor. Both the AL-CMA-ES and EL-ES also appear to enter a final period of oscillation with no further improvements near the end of these runs on the NFR sphere.

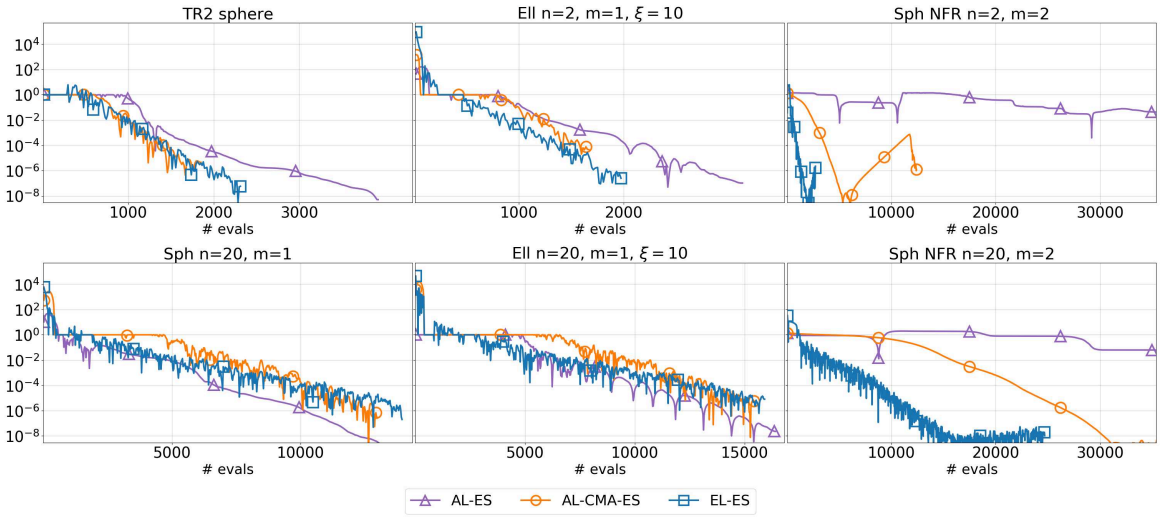

Figure 5.8: Convergence plots showing distance $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$ from the optimal Lagrange multiplier vector with respect to the first 3.5×10^4 $(f+g)$-evals for median runs from the three Lagrangian methods. The x-axes are scaled to present as much detail as possible without obscuring relevant data points.

In order to generate ECDF data for the same six problems, target sets are fixed according to Eq. (5.2) with f-target and g-target values evenly logarithmically spaced

in the closed intervals $[10^0, 10^{-8}]$ and $[10^2, 10^{-6}]$, respectively. The function $g_{\mathcal{A}}$ defined in Eq. (5.4) is used to evaluate against g-targets. Note that the termination condition used for this data set is more strict than any in the target sets, thus ensuring each algorithm is permitted to succeed on as many targets as it is able.

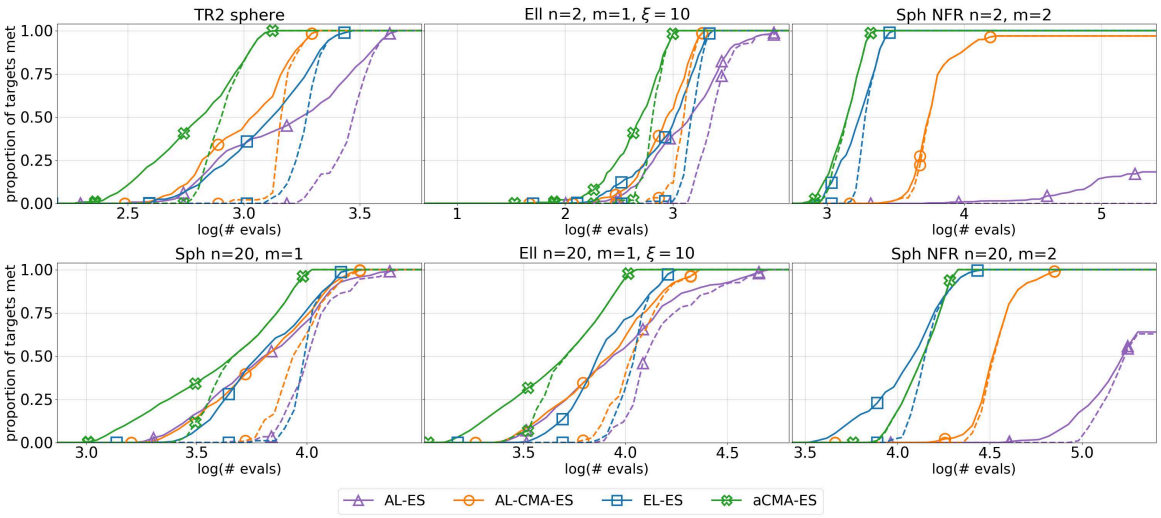

Figure 5.9: ECDF plots showing $(f + g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines). The x-axes are scaled to present as much detail as possible without obscuring data points.

Plots are given for each problem in Figure 5.9 representing combined performance using staggered ECDFs as described in Section 5.1.2. In these curves, the poor performance of AL-ES on the NFR spheres is made apparent; for the $n = 2$ case, fewer than 20% of the targets are reached. The NFR spheres in both dimensions also show the largest differences in performance overall, with the EL-ES converging significantly faster than either of AL-CMA-ES or AL-ES, and AL-CMA-ES even failing to reach all targets for the $n = 2$ case. For $n = 20$, EL-ES also out-performs the aCMA-ES on some targets. Across all $m = 1$ problems, the performance is roughly equivalent between AL-CMA-ES and EL-ES, with a small advantage for AL-CMA-ES when $n = 2$ and a small advantage for EL-ES when $n = 20$. This may be explained by the EL-ES requiring one additional f evaluation per iteration; both approaches evaluate $f(\boldsymbol{y}_i^{(k)})$ for the λ offspring in the k-th iteration, however only the EL-ES needs to additionally evaluate $f(\boldsymbol{x}^{(k)})$ while approximating $\boldsymbol{\alpha}^{(k+1)}$.

Additional ECDF plots are given in Figure C.3 of Appendix C that separately show

progress on the full range of f- and g-targets. On the NFR sphere, the difference in performance between aCMA-ES and EL-ES on $n = 2$ versus $n = 20$ can be attributed to the performance on f-targets, as the g-target plots are almost identical for both dimensions.

Overall, the EL-ES is at least a close competitor throughout to the AL-CMA-ES, and draws roughly even with the aCMA-ES performance on some measures, in spite of not using CMA for generating offspring. Specifically on the NFR problems, the EL-ES appears to be the clearly superior choice; the aCMA-ES has better performance in smaller dimensions, but this advantage appears to decrease markedly with increasing dimensionality of the problem. The ability of the EL-ES to deal with narrow feasible regions is due in part to the construction of its Lagrange multiplier, which accounts for both the magnitudes of and correlations between the various constraints. For the approaches based on AL-ES, the NFR sphere problems result in both increased ill-conditioning for the Lagrangian functions as well as increased difficulty in converging to an optimal Lagrange multiplier, and these issues are only partly addressed by the addition of CMA.

Experimental results with varied scaling

The linearly constrained spheres of Section 5.2.1 involve objective and constraint functions with equal scaling, in the sense that both the magnitudes of the constraints' normal vectors and the eigenvalues of $\frac{1}{2}H$ from Eq. (5.8) are equal to one. For the linearly constrained ellipsoids, the eigenvalues of $\frac{1}{2}H$ vary because of the parameter ξ, but the smallest eigenvalue is still equal to one. In order to observe the algorithms' behaviour on problems with different scaling factors between the objective and constraint functions, two new problem sets are considered with increased objective function scaling (termed large A) and with increased constraint function scaling (termed large B). Formally, the large A problems use coefficient $A = 10^3$ and re-define

the sphere and ellipsoid functions as

$$f_{\text{ell}}(\boldsymbol{x}) = A \cdot \sum_{i=1}^{n} \xi^{\left(\frac{i-1}{n-1}\right)} x_i^2, \tag{5.9}$$
$$f_{\text{sph}}(\boldsymbol{x}) = A \cdot \sum_{i=1}^{n} x_i^2$$

while the large B problems set $B = 10^3$ and re-define the constraint functions as

$$g_i(\boldsymbol{x}) = B \cdot \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x} + c_i \leq 0. \tag{5.10}$$

In both variations, only the related scaling factor is changed, and no other aspects of the problem definitions are modified.

Large B scaling

The methodology for generating data for the large B problem variants is the same as in Section 5.2.1, except the six problem sets use the updated constraint function definitions of Eq. (5.10). Figure 5.10 gives convergence plots for the distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$, while additional Figures C.4 - C.5 in Appendix C give convergence plots for the step size σ and normalized distance $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$. As before, each plot is with respect to the total number of $(f + g)$ function evaluations consumed by median runs of the corresponding 25 runs set, and truncated to the first 3.5×10^4 evaluations to ensure relevant details are visible.

As with the unit scaled problems, the performance of all four algorithms exhibits varying degrees of log-linear convergence on the $m = 1$ problems. On these median runs, the most significant differences are again on the NFR spheres, with AL-ES in particular showing erratic step size adaptation and poor convergence towards the optimum. Convergence to the optimal Lagrange multiplier by the EL-ES is also notably faster for the NFR spheres.

Figure 5.11 gives staggered ECDFs for the large B variants. Performance on the $m = 1$ sphere and ellipsoid problems gives a similar overall ranking of the algorithms as in Section 5.2.1, with the EL-ES drawing closer to the AL-ES performance on the

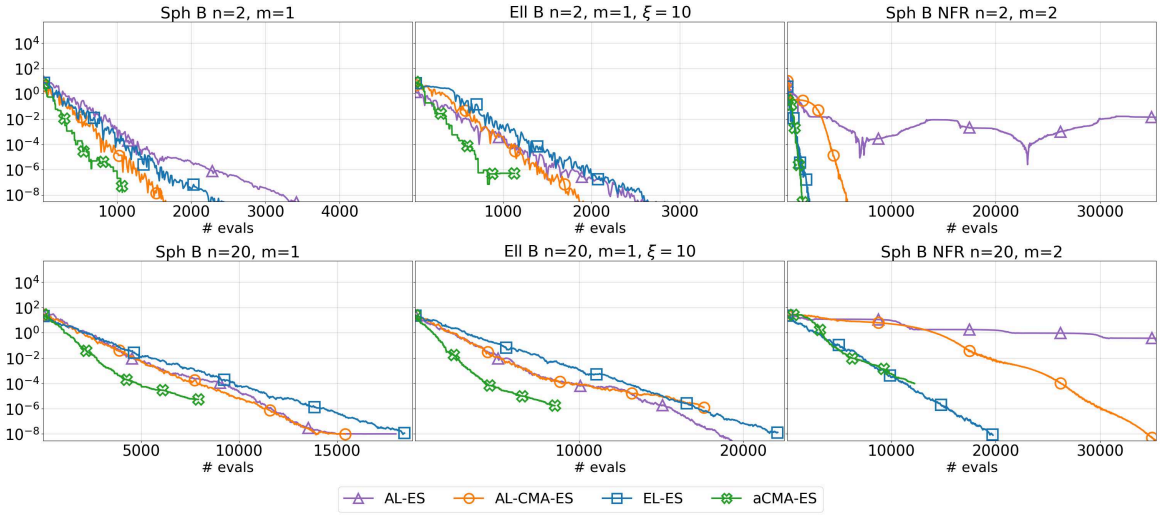

Figure 5.10: Convergence plots showing distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ from the constrained optimum with respect to the first 3.5×10^4 $(f + g)$-evals for median runs from each of the four algorithms on large B variants. The x-axes are scaled to present as much detail as possible without obscuring relevant data points.

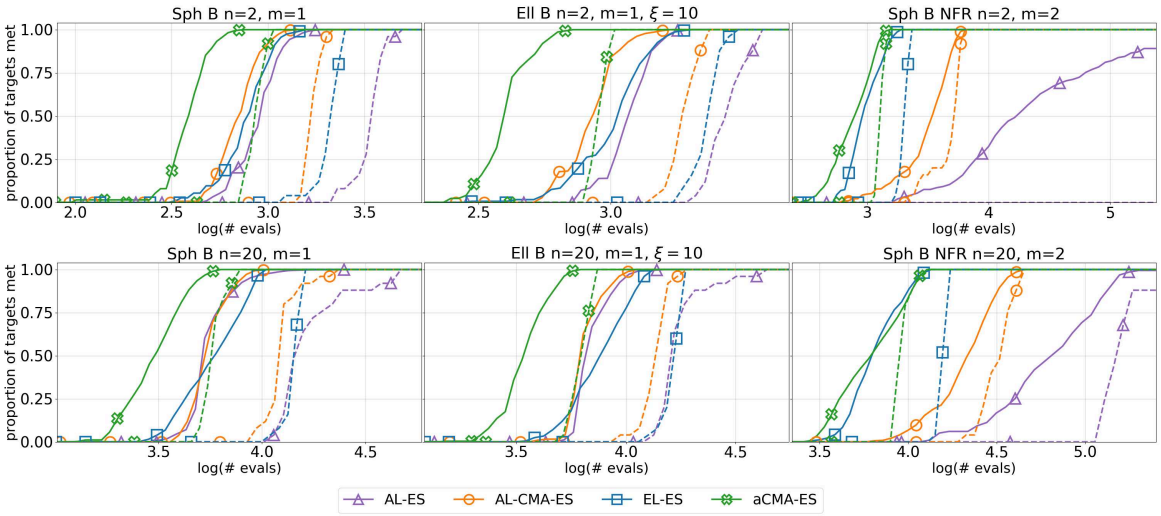

Figure 5.11: ECDF plots showing $(f + g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines) on large B problem variants. The x-axes are scaled to present as much detail as possible without obscuring data points.

moderately conditioned ellipsoids. As with the unit-scaled problems, the aCMA-ES is overall superior. However, on the $m = 2$ large B NFR sphere problems, the EL-ES is superior to both of the other Lagrangian methods, with the AL-ES failing to meet all f-targets even with the easiest staggered g-target.

Additional ECDF plots are given in Figure C.6 of Appendix C that show progress on the separate f- and g-targets for the large B problem variants, similar to Figure C.3. For the $m = 2$ NFR spheres in particular, the EL-ES is seen to be roughly equal in performance to the aCMA-ES on g-targets and superior to the other two Lagrangian methods on both target types.

Large A scaling

The methodology for generating data for the large A problem variants is the same as in Section 5.2.1, except the six problem sets use the updated objective function definitions of Eq. (5.9). Figure 5.12 gives convergence plots for the distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$, while additionally Figures C.7 - C.8 in Appendix C give convergence plots for the step size σ and normalized distance $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$. As before, each plot is with respect to the total number of $(f + g)$ function evaluations consumed by median runs of the corresponding 25 runs set, and truncated to the first 3.5×10^4 evaluations to ensure relevant details are visible

Roughly log-linear convergence is shown again by all of the algorithms' median runs on the $m = 1$ problems, while the NFR spheres are again problematic for the AL-CMA-ES and especially the AL-ES. The flattening of the curves at the end of each median run for the $n = 20$ NFR sphere appears to be a result of the limitations of numerical accuracy caused by selecting both large A and larger n.

Staggered ECDF plots are shown in Figure 5.13 for the four algorithms evaluated on the six large A problem sets. The aCMA-ES remains dominant on the four $m = 1$ problems, although by a more narrow margin than in either the unit scaled or large B problem variants. The performance of the EL-ES and AL-CMA-ES is again comparable on the $m = 1$ problems, with a slight advantage on the $n = 2$ sphere. The advantage of the EL-ES is pronounced on the NFR sphere in both dimensions,

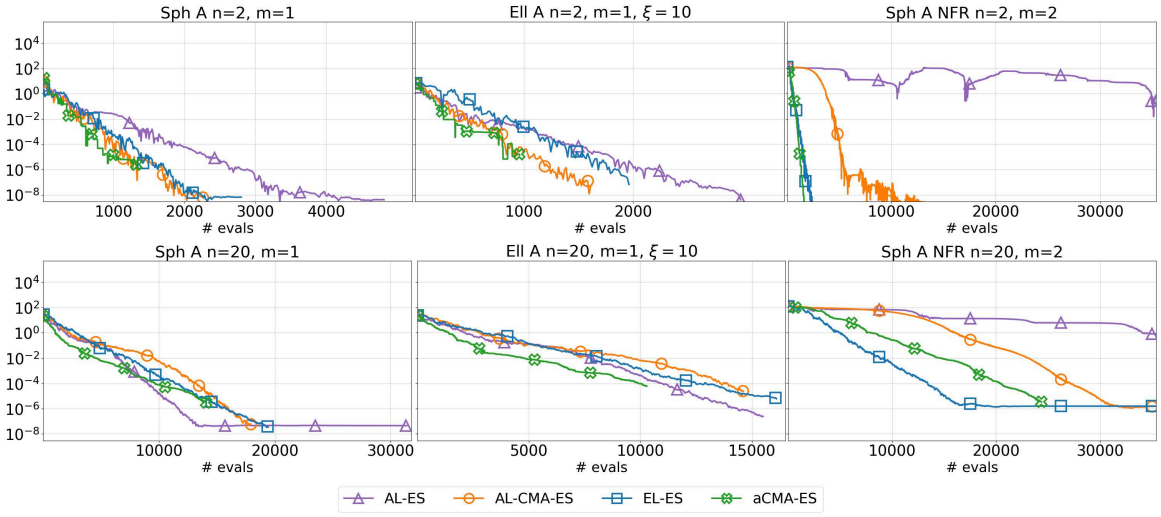

Figure 5.12: Convergence plots showing distance $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ from the constrained optimum with respect to the first 3.5×10^4 $(f + g)$-evals for median runs from each of the four algorithms. The x-axes are scaled to present as much detail as possible without obscuring relevant data points.

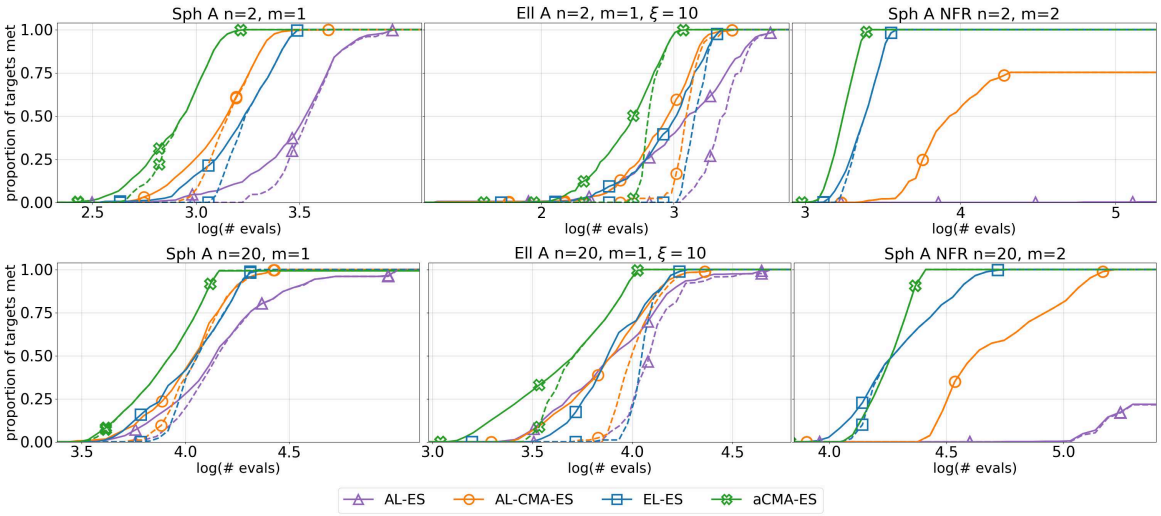

Figure 5.13: ECDF plots showing $(f + g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines) on large A problem variants. The x-axes are scaled to present as much detail as possible without obscuring data points.

with its performance even exceeding the aCMA-ES on some of the easier targets with $n = 20$. Both the AL-ES and AL-CMA-ES struggle significantly on the two NFR sphere problems, and for $n = 2$ neither are able to converge to all targets even within 10^0 of feasible. Additional ECDF plots are given in Figure C.9 of Appendix C showing progress on the separate f- and g-targets for the large A problem variants. These highlight the difficulty of the augmented Lagrangian methods on meeting the f-targets for the NFR spheres in particular.

Summary for fixed constraints

The EL-ES is seen to converge reliably on the eighteen problem variations tested with fixed constraints (six each for unit, large B, and large A scaling), with measured performance on f- and g-targets generally close to that of the AL-CMA-ES on single-constrained problems in spite of the use of CMA for improved offspring generation. Convergence plots for these problems also show that the step size and distance from optimum decrease log-linearly with respect to function evaluations, in line with the other evolution strategies. The NFR sphere with two constraints is seen to be a difficult problem for existing augmented Lagrangian methods; in some contexts, they are not able to converge at all, even while using CMA. The EL-ES is consistently successful on this problem however, and in larger dimensions is even able to exceed the performance of the aCMA-ES. Convergence plots for the Lagrangian methods demonstrate that the Lagrange multiplier approximations of the EL-ES can be significantly more accurate after an equal number of function evaluations when compared to either the AL-ES or AL-CMA-ES.

5.2.2 Random constraints

Fixed linear constraints as determined by the experimenter provide a valuable baseline for performance, but do not accurately encompass the full range of possible constraint combinations. One possibility for doing so is to construct a parameterization for the constraints that allows their random generation in a reliable and unbiased manner.

Atamna et al. [18] generate constraints by means of normal vectors $\boldsymbol{b}_i$, where each constraint function is linear and represented as

$$g_i(\boldsymbol{x}) = \boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x} + c_i \leq 0.$$

A fixed point $\boldsymbol{x}^*$ is chosen to serve as the constrained optimum, then the constraint normals are generated so that the selected point lies on the boundary of each feasible region. The first constraint is always determined by setting $\boldsymbol{b}_1 = -\nabla f(\boldsymbol{x}^*)$ and $c_1 = -\nabla f(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{x}^*$, giving a constraint that immediately satisfies the first-order KKT condition of Eq. (B.11) with $\alpha_1 = 1$ and guarantees the chosen $\boldsymbol{x}^*$ will be a constrained optimum for convex f. Any additional constraints are then added by sampling from a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to form additional vectors $\boldsymbol{b}_i$ and setting the matching $c_i = -\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}^*$ to maintain $\boldsymbol{x}^*$ as the optimum. For each additional vector so constructed, a simple check is performed to ensure that the point $\boldsymbol{x}^* + \nabla f(\boldsymbol{x}^*)$ remains feasible, setting $\boldsymbol{b}_i = -\boldsymbol{b}_i$ and $c_i = -c_i$ if needed.

The result of this process is a set of linear constraints that are all active at the optimum in the sense that all $g_i(\boldsymbol{x}^*) = 0$, but for which only g_1 has a nonzero Lagrange multiplier. In the context of the complementary slackness condition in Eq. (B.13), all constraints beyond the first one are *weakly* active. This construction makes sense for the context in which it was originally used, but is a somewhat limited problem formulation for testing a Lagrangian algorithm in that only a single Lagrange multiplier ever needs to be approximated. The same method is also used by the COCO benchmark for constrained optimization[2], which does not evaluate algorithms based on Lagrange multipliers. I propose an alternative here that duplicates desirable features of the method used by Atamna et al. (such as fixing $\boldsymbol{x}^*$ at a chosen point), but ensures that each active constraint will have an associated positive Lagrange multiplier.

Given a convex objective function f, the goal is to generate $m \leq n$ active linear constraints with respect to a chosen point $\boldsymbol{x}^*$ such that at that point:

[2]Taken from the COCO outline for bbob-constrained at `http://numbbo.github.io/coco-doc/bbob-constrained/`, retrieved Apr 25, 2022.

1. all constraints $g_i(\boldsymbol{x}) = \boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x} + c_i$ have associated positive Lagrange multipliers α_i, and

2. the gradient $\nabla f(\boldsymbol{x}^*)$ can be written as a linear combination of the constraint normals $\nabla g_i = \boldsymbol{b}_i$ using these α_i.

These conditions together satisfy the KKT necessary conditions of Theorem B.6, and since the functions are convex the conditions are in fact sufficient for $\boldsymbol{x}^*$ to be a constrained optimum of the ICP. These can equivalently be combined to require at $\boldsymbol{x}^*$ the lone condition that

1. the gradient $\nabla f(\boldsymbol{x}^*)$ can be written as a linear combination of constraint normals using positive α_i, and there is no proper subset of the constraint normals for which this is also true.

This essentially excludes the possibility of any linear dependence between the constraint normals, as in the LICQ of Definition B.2. For any constraint g_i with normal $\boldsymbol{b}_i$, we will ensure $||\boldsymbol{b}_i|| = 1$ by normalizing vectors as needed, and always set $c_i = -\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}^*$ so that $\boldsymbol{x}^*$ lies on the boundary of the constraint as desired. It therefore suffices to determine how the directions of the constraint normals should be generated.

To begin, sample independently from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to create $(m-1)$ random vectors $\boldsymbol{b}_i$ constituting the constraint normal vectors, and normalize them to be of unit length. This is equivalent to creating unit vectors from sampling $(m-1)$ angles independently and uniformly from $[0, 2\pi)$, corresponding with rotations about the origin. Additionally, these vectors are linearly independent with probability 1 since the dimension n is strictly greater than $(m-1)$. The final constraint normal $\boldsymbol{b}_m$ must then be chosen carefully so as to satisfy the first-order KKT condition Eq. (B.11). This can be re-arranged to write

$$\boldsymbol{b}_m = a_0(-\nabla f) + \sum_{i=1}^{m-1} a_i(-\boldsymbol{b}_i)$$

in terms of positive scalars a_i. With appropriate normalizations, this is equivalent

to

$$\boldsymbol{b}_m = \frac{1}{a} \cdot \sum_{i=1}^{m-1} [(z_i)(-\nabla f) + (1 - z_i)(-\boldsymbol{b}_i)] \tag{5.11}$$

where the z_i lie in the open interval $(0, 1)$, and $a > 0$ is simply a normalization constant ensuring $\boldsymbol{b}_m$ is a unit vector. Thus the final constraint normal $\boldsymbol{b}_m$ can be generated by sampling z_i from the open uniform distribution $\mathcal{U}(0, 1)$ and then normalizing by the resulting vector's length to arrive at the value given in Eq. (5.11).

The result of this process is m vectors that satisfy the needed KKT condition: the objective gradient can be written as a linear combination of the constraint normals at the optimum using positive coefficients, which are the Lagrange multipliers. With probability 1 they are linearly independent, and so no smaller subset of the $\boldsymbol{b}_i$ could be used to represent the gradient of f as a linear combination.

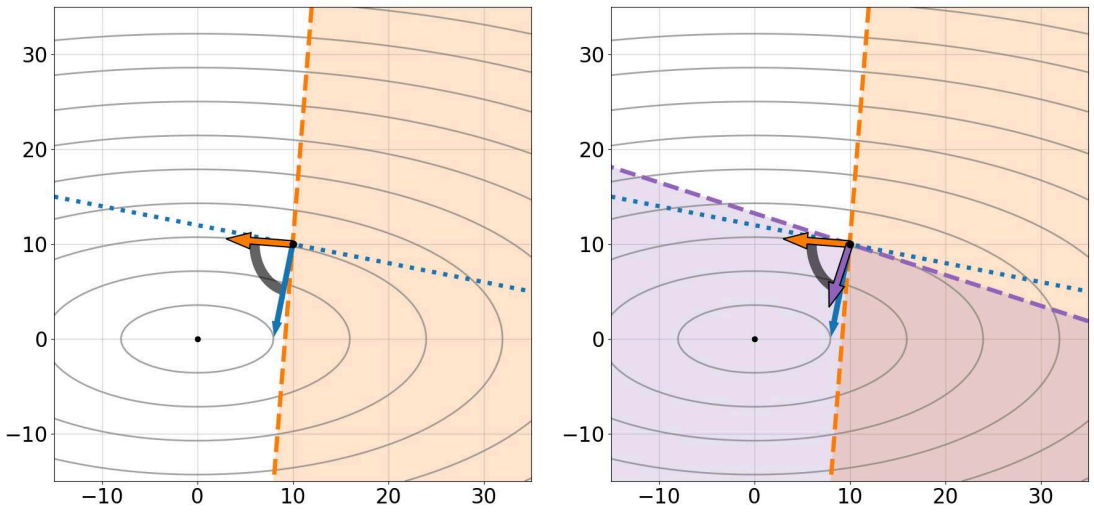

Figure 5.14: Visualized stages for generating two active linear constraints in $n = 2$. Arrows correspond to vectors $-\nabla f(\boldsymbol{x}^*)$ (blue), $-\boldsymbol{b}_1$ (orange), and $\boldsymbol{b}_2$ (purple). The line $\boldsymbol{x}^{\mathrm{T}} \nabla f(\boldsymbol{x}^*) = 0$ is given by a dotted line, while both constraint boundaries are given by dashed lines and their infeasible regions shaded. Contour lines shown are for the ellipsoid objective function $f(\boldsymbol{x})$.

The process can be made clear through a simple example in $n = 2$ for generating $m =$

2 active linear constraints. Let f be a convex quadratic function with unconstrained minimum at the origin[3] and $\boldsymbol{x}^* = [10, 10]$ be the selected optimum. A visualization of the resulting process is given in Figure 5.14. In order for the selected point $\boldsymbol{x}^* = [10, 10]$ to be the constrained optimum under linear constraints, all points superior to $\boldsymbol{x}^*$ with respect to f must end up in the infeasible region. The linear open boundary of the region containing all those points with an objective function value smaller than $f(\boldsymbol{x}^*)$ is given by $\boldsymbol{x}^{\mathrm{T}} \nabla f(\boldsymbol{x}^*) = 0$ (dotted blue lines in the figure), which is the line orthogonal to the gradient ∇f at $\boldsymbol{x}^*$ and tangent to the objective function f at $\boldsymbol{x}^*$. This line divides the plane into two half-planes, a *feasible half-plane* and an *infeasible half-plane*, the latter of which includes the origin for our chosen f. Clearly, all points in the infeasible half-plane must end up being infeasible under our constructed linear constraints (or else there would be a feasible point $\boldsymbol{y}$ for which $f(\boldsymbol{y}) < f(\boldsymbol{x}^*)$) and the feasible half-plane must end up being non-empty (or else there would be no feasible solutions to f)[4]. After randomly generating the first constraint normal (left image in the figure), the infeasible region associated with $\boldsymbol{b}_1$ is seen to cover only part of the infeasible half-plane. The range of possible choices for the second constraint normal is therefore restricted to the highlighted arc between the vectors $-\nabla f(\boldsymbol{x}^*)$ and $-\boldsymbol{b}_1$, which visualizes the relationship given in Eq. (5.11). So long as the second constraint normal is selected from within this range (right image in the figure), the resulting infeasible region will encompass the remainder of the infeasible half-plane, as desired.

Experimental results

By following the given random process for generating active constraints, problems are created based on the $n = 2$ sphere with $m = 2$, and on the $n = 10$ sphere with $m = 2, 5$, and 10. Data sets are generated for each selected combination of parameters n and m by performing 100 runs using each of the four algorithms described in Section 5.2.1. The larger number of runs is chosen to help account for the fact that problem definitions will change between each run. In order to ensure a fair

[3]For this example, I specifically use f_{cig} with $\xi = 5$, but for any other appropriate objective function the process is the same.

[4]Taken together, these give the overall restriction that neither constraint normal $\boldsymbol{b}_i$ can be equal to either of $\nabla f(\boldsymbol{x}^*)$ or $-\nabla f(\boldsymbol{x}^*)$, and this will be true with probability 1.

comparison, each combination of parameters n and m is assigned a random seed that is then used to generate in advance 100 constraint sets for all 100 runs, resulting in each algorithm being given the same sequence of randomly-generated constraints for each problem set. As in Section 5.2.1, target sets for the ECDFs rely on the definition in Eq. (5.2) with f-target and g-target values evenly logarithmically spaced in the closed intervals $[10^0, 10^{-8}]$ and $[10^2, 10^{-6}]$, respectively, and runtimes are again measured as f-evals, g-evals, or $(f+g)$-evals, depending on the target set.

A run is terminated when $f(\boldsymbol{x})$ and $g_{\mathcal{A}}(\boldsymbol{x})$ are both within 10^{-8} of optimum values, represented by $f(\boldsymbol{x}^*)$ and $g_{\mathcal{A}}(\boldsymbol{x}^*) = 0$, respectively, or when more than 10^5 total $(f+g)$ function evaluations have been used. It will be seen that convergence for these problems typically occurs with not much more than 10^4 total function evaluations. Starting points are initialized randomly using coordinates drawn independently and uniformly from the interval $[-10, 10]$. To avoid the computational expense involved in generating a full set of 200 feasible points for each of the 100 distinct problems in each problem set, a smaller scale approach is instead undertaken for enabling the aCMA-ES to have a feasible starting point. At the start of each run, a pre-processing step uses the CMA-ES to minimize the sum of constraint violations $g_{\Sigma}(\boldsymbol{x})$ in a small series of 15 runs. These runs are sorted by g-function evaluations and the median entry is selected to serve as the starting point for the aCMA-ES algorithm. The number of g-evals is also initialized to the number of evaluations used by the median entry to locate the feasible point.

Staggered ECDF plots are given for each problem in Figure 5.15 representing combined performance plotted against the sum of f- and g-evaluations. The interpretation of the lines and staggered f- and g-targets is otherwise the same as in Figure 5.9, where the TR2 sphere $(n = 2, m = 1)$ serves as a point of performance comparison. Despite only adding one additional constraint, the performance of each of the aCMA-ES, AL-CMA-ES, and AL-ES algorithms is significantly degraded with the $m = 2$ random constraints, while the performance of the EL-ES is visually quite similar to that seen in the previous $m = 1$ case of Figure 5.9. The EL-ES strictly dominates the performance of the three other algorithms on the $n = 10$ sphere for both $m = 5$ and $m = 10$, and is closely competitive with the aCMA-ES for $m = 2$. As the only

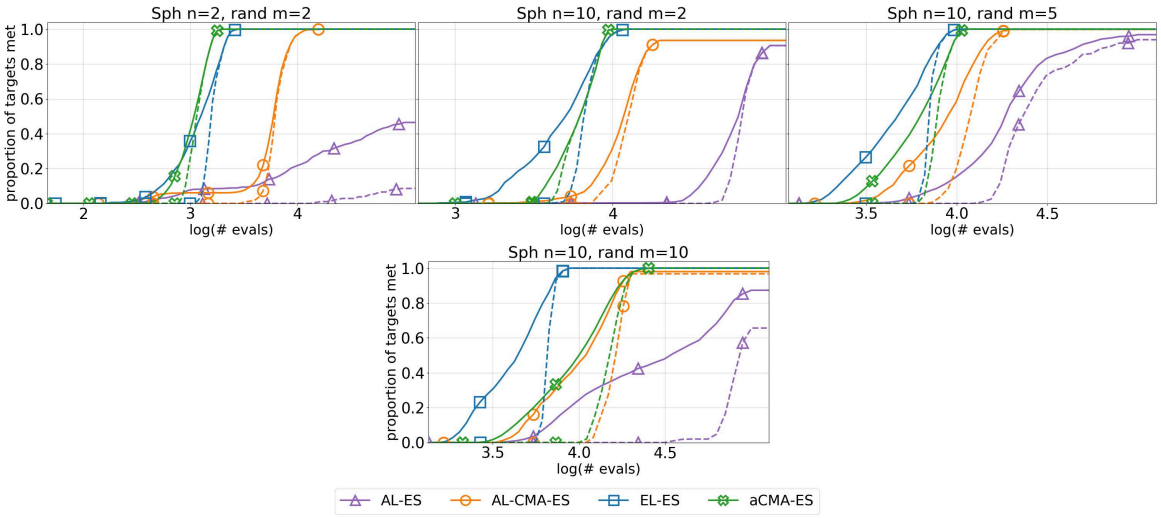

Figure 5.15: ECDF plots showing $(f+g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines) for randomly generated constraints on $n = 10$ spheres. The x-axes are scaled to present as much detail as possible without obscuring data points.

other non-CMA algorithm, AL-ES performs quite poorly across all problems.

Similar staggered ECDF plots for $n = 20$ spheres are presented in Figure 5.16 with $m = 2, 10$, and 20. The process for generating 100 runs and other criteria are identical to those used to generate Figure 5.15, and the single staggered plot for the $n = 2, m = 2$ random sphere is included again verbatim to facilitate comparison.

As in the case of the $n = 10$ spheres, the performance benefits of the Exact Lagrangian approach appear to increase along with the number of constraints. The dimension of the search space is now large enough that the EL-ES is superior to the aCMA-ES in all cases, and is also strictly superior to both other Lagrangian approaches throughout.

Additional plots for the separate f- and g- targets are given in Figures C.10 and C.11 for the $n = 10$ and $n = 20$ problems, respectively. The main advantages of EL-ES over other algorithms is seen here to be due primarily to its performance with respect to convergence on the g-targets, where it is strictly dominant over all other algorithms for the most difficult 80% of the targets. In the plot for the AL-ES on the sphere with $n = 10$ and $m = 2$, the associated line is almost invisible because only 11 of 25

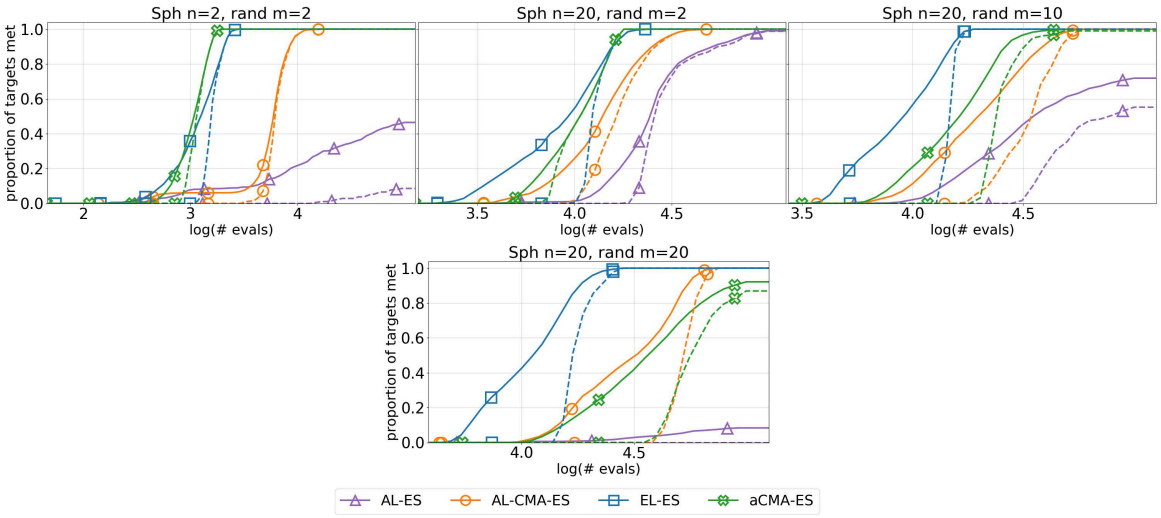

Figure 5.16: ECDF plots showing $(f+g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines) for randomly generated constraints on $n = 20$ spheres. The x-axes are scaled to present as much detail as possible without obscuring data points.

runs are recorded as able to reach even the first f-target (resulting in the proportion of met targets being slightly under 0.9%) and none came closer than 10^{-1}.

Summary for random constraints

The overall performance of the EL-ES on randomly generated active constraints is seen to be notably superior to the two other Lagrangian methods tested. With increasing constraint number, the performance of the EL-ES also appears to improve relative to all other algorithms. On even a moderate $n = 10$ dimensional problem, the EL-ES is able to out-perform the aCMA-ES with $m = 5$ and $m = 10$ constraints by almost a factor of two with respect to total function evaluations. On the $n = 20$ sphere problems, the superior performance of the Exact Lagrangian approach is evident across all chosen numbers of constraints.

5.3 Benchmarks from the literature

To exhibit broader applicability of the proposed Exact Lagrangian algorithm, it is necessary to compare performance against other algorithms in the literature. A set

of constrained optimization problems has been selected with published experimental results for evolution strategies [6, 38] that includes S240 and S241 taken from [107], and G04 (also referred to as HB or Himmelblau's problem), G06, G07, and G09 from the 2006 CEC competition [77]. In addition, Rosenbrock's Parcel problem is taken from [43]. These are all unimodal constrained problems with a mixture of linear and nonlinear inequality constraints. Problems S240 and S241 do not have upper bound constraints, while all other problems have both upper and lower bound constraints defined in addition to their other inequality constraints. Table 5.1 gives a summary of the various problem attributes. Note that for problem G04, Dufossé and Hansen [38] identify an error in previous definitions of the non-linear constraints, which has been corrected here. Additionally, the identified number of constraints active at the optimum has previously been in error [6, 38]. The correct constraint definition [65] is used in experiments throughout this section, and the corrected value of m_{act} for G04 is given in Table 5.1. Full definitions for all of these functions are given in Appendix A.

Problem	n	m	m_{act}	f	g_i
S240	5	6	5	lin.	lin.
S241	5	6	5	lin.	lin.
Parcel	3	7	1	non-lin.	lin.
G04 (HB)	5	16	5	non-lin.	non-lin.
G06	2	6	2	non-lin.	non-lin.
G07	10	28	6	non-lin.	both
G09	7	18	2	non-lin.	non-lin.

Table 5.1: Summary of problem attributes used in benchmark including dimension n, total number of constraints m, number of active constraints m_{act}, linearity or non-linearity of objective function f, and whether non-bound constraints g_i are linear, non-linear, or a mix of both.

5.3.1 Experimental results

Data sets are generated for each problem by performing 25 runs using each of the four algorithms used in Section 5.2, which are the **aCMA-ES** $(1+1)$ evolution strategy, the **AL-ES** $(\mu/\mu_W, \lambda)$ evolution strategy outlined in Section 4.2 and implementing Algorithm 2.2, the **AL-CMA-ES** $(\mu/\mu_W, \lambda)$-ES with covariance matrix adaptation outlined in Section 4.2 and implementing Algorithm 2.3, and **EL-ES** as proposed in

Chapter 4. The same target sets are fixed according to Eq. (5.2) with f-target and g-target values evenly logarithmically spaced in the closed intervals $[10^0, 10^{-8}]$ and $[10^2, 10^{-6}]$, respectively. As before, the function $g_{\mathcal{A}}$ defined in Eq. (5.4) is used to evaluate against g-targets, and runtimes are measured as counts of f, g, or $(f + g)$ evaluations.

All runs are terminated only when $f(\boldsymbol{x})$ and $g_{\mathcal{A}}(\boldsymbol{x})$ are both within 10^{-8} of optimum values, represented by $f(\boldsymbol{x}^*)$ and $g_{\mathcal{A}}(\boldsymbol{x}^*) = 0$, respectively, ensuring that each algorithm is permitted to succeed on as many targets as it is able. Problems S240 and S241 use the recommended initial starting points, while all other problems have their initial starting points generated by sampling randomly and uniformly within the region defined by the problem's bound constraints. As the aCMA-ES requires a feasible starting point, a set of feasible points is generated in a pre-processing step similar to [38] and mirroring Section 5.2 by using CMA-ES to minimize the sum of constraint violations function $g_{\Sigma}(\boldsymbol{x})$ in a series of 200 runs. The resulting set of 200 feasible points within the search space is sampled from uniformly to initialize the aCMA-ES, and the count of g-evaluations is initialized to the number used to locate the feasible starting point.

ECDF plots are given for each problem in Figure 5.17 representing combined performance by showing success on staggered targets plotted against the sum of f- and g-evaluations. As for the staggered plots in Section 5.2, the combined performance is displayed for f-targets with fixed g-target of 10^0 (solid lines) and fixed g-target of 10^{-6} (dashed lines). Thus, runs on the first target set are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_{\mathcal{A}} \leq 10^0$ are satisfied, and runs on the second target set are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_{\mathcal{A}} \leq 10^{-6}$ are satisfied.

In these staggered target curves, the overall performance of the top three algorithms is seen to be generally competitive. The combined performance for EL-ES and aCMA-ES is seen to be nearly equivalent on S240, S241, and G04, particularly for the fixed target $g_{\mathcal{A}} \leq 10^0$ with small differences visible elsewhere. The combined performance of aCMA-ES is superior on Rosenbrock's Parcel, G06, and G09, while the Exact Lagrangian is narrowly superior on G07. The EL-ES is superior to the other augmented

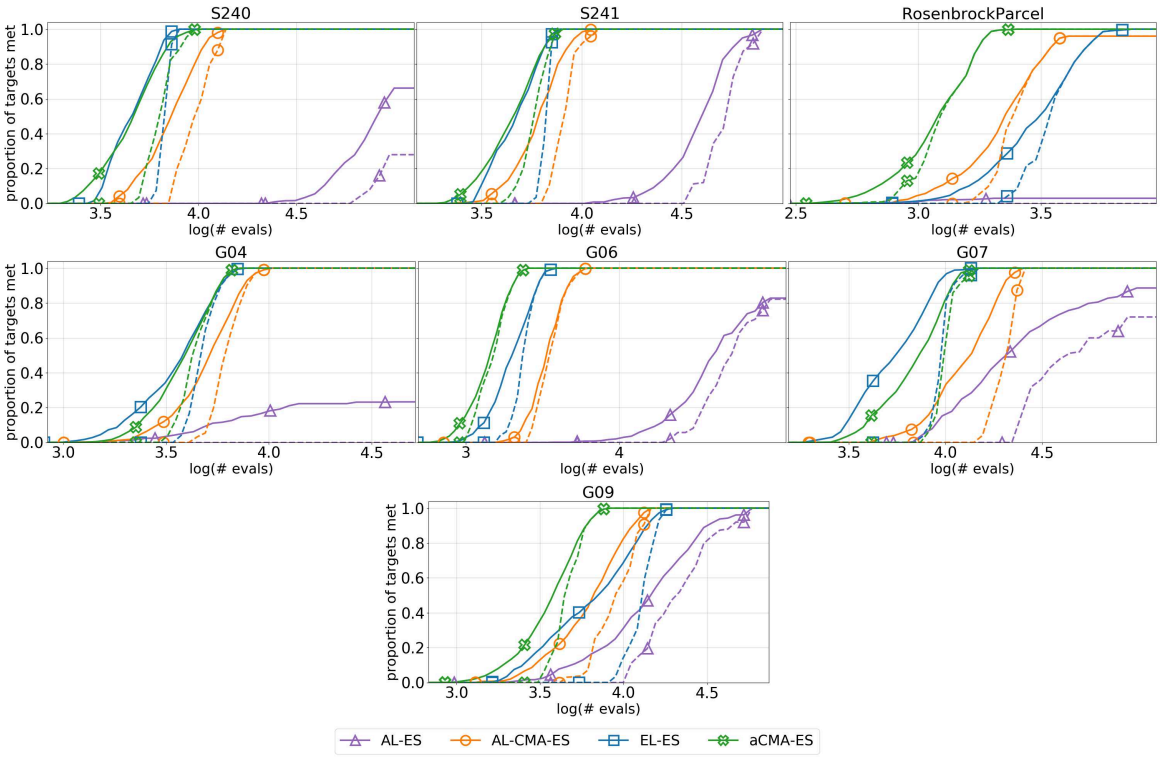

Figure 5.17: ECDF plots showing $(f + g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines). The x-axes are scaled to present as much detail as possible without obscuring data points.

Lagrangian algorithms on all problems except Rosenbrock's Parcel and G09, where for both problems it is slightly behind the AL-CMA-ES but still notably superior to the AL-ES.

Additional ECDF plots are given in Figure 5.18 showing separate performance on f- and g-targets. The plots are grouped together by problem and arranged into pairs of rows representing the proportion of successful f-targets plotted against the count of f-evaluations (top) and successful g-targets plotted against the count of g-evaluations (bottom). A distinctive feature of aCMA-ES is evident as it is seen to require significantly fewer f-evaluations to succeed on all targets, visible in the top rows with the best performance on f-targets for each problem.

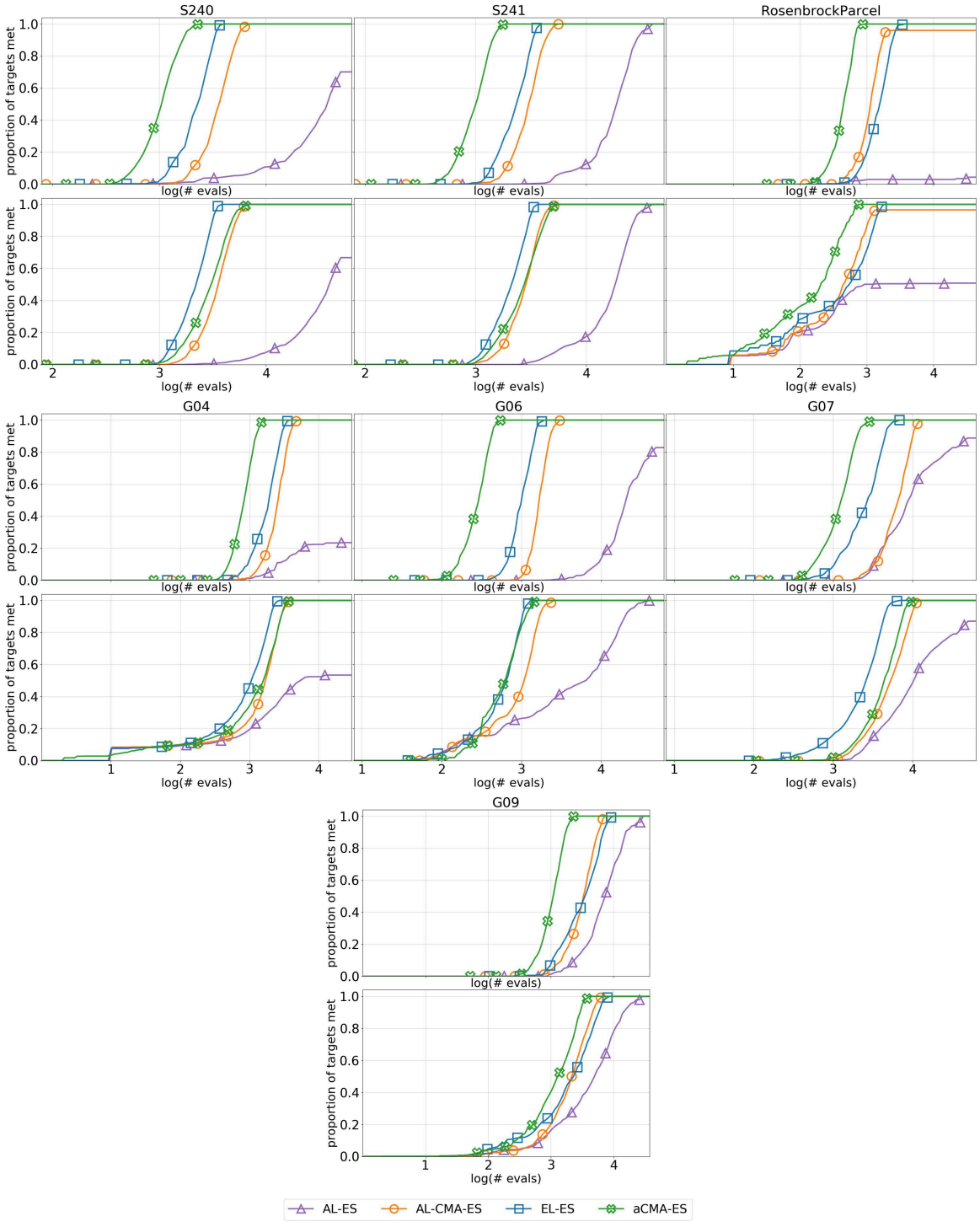

Figure 5.18: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom). The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

5.3.2 Summary for literature benchmarks

The overall performance of EL-ES is favourable on the selected problems, even while comparing against ES algorithms enabled by covariance matrix adaptation that exploit more information from the search space and avoid issues from ill-conditioning. On four of the seven problems (S240, S241, G04, and G07), the combined performance of EL-ES is approximately equal or superior to that of all other algorithms. Only on two problems (RosenbrockParcel and G09) is the EL-ES performance not strictly superior to that of both the other Lagrangian methods, and these both have relatively ill-conditioned objective functions. While there are evident advantages from including CMA with the AL-ES approach, which is to be expected, the EL-ES approach on its own is able to deal with some the Lagrangian ill-conditioning, likely due to the inclusion of constraint information in its multiplier update.

5.4 Rotated Klee-Minty problem

The Klee-Minty problem is a scalable constrained optimization problem with linear objective function and linear constraints. It was proposed originally as a pathological case for which the simplex algorithm exhibits worst-case performance [71] and more recently modified by Hellwig and Beyer [59] with the inclusion of a translation and rotation to make it suitable as a potential benchmark for probabilistic search algorithms. They provide initial experimental results for both the CEC2017 competition winner L-SHADE [90] based on differential evolution and their own algorithm ϵMAg-ES [60] based on a reduced form of CMA-ES with ϵ-level constraint handling. The benchmark has also been used Spettel et al. [117] while introducing the lcCMSA-ES algorithm, a CMA-ES variant specifically designed for solving problems with linear constraints, and by the same authors [63] in a broader survey of stochastic algorithms, including various CEC competition winners as well as active-set-ES [112].

The original Klee-Minty problem defines an ICP in n dimensions having $2n$ inequality constraints. Geometrically, the feasible space is contained within a hypercube with slightly distorted corners and the linear objective function is given by

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x}$$

where $\boldsymbol{c} = [0, 0, \ldots, 1]$ so the optimal point is located at the origin. The Rotated Klee-Minty problem applies a transformation to the constraints by means of a rotation about the origin by angle $\frac{-350}{180}\pi$ and translation by the vector $[n^3, n^3, \ldots, n^3]$. Bound constraints are also applied, primarily to limit the region for generating initial (feasible, if necessary) points.

5.4.1 Experimental results

Comparative data sets are generated for four algorithms on Rotated Klee-Minty problems. The ϵ**MAg-ES** [60] algorithm implements a reduced variant of CMA-ES alongside ϵ-level comparisons and gradient-based repairs from Takahama and Sakai [123]. The authors note that within each iteration that uses the repair operation, extra constraint function evaluations are consumed making the action more expensive. The ϵMAg-ES method was retroactively ranked third in 2019 among all submissions to the CEC 2017 problem set, using the prescribed $2n \cdot 10^4$ budget of function evaluations for problems of dimension n. The **lcCMSA-ES** [117] implements another reduced variant of CMA-ES with a special focus on repair and projection of points into the unconstrained subspace. The assumption of constraints being linear is strict, and the first step of the algorithm is to gather a large number of sample points to be used in the linear projection of infeasible points. Both ϵMAg-ES and lcCMSA-ES are implemented using code from the authors[5]. Additionally, both the **AL-CMA-ES** and **EL-ES** are used as described in Sections 5.2 and 5.3. Both the lcCMSA-ES and ϵMAg-ES algorithms have published experimental results on the Rotated Klee-Minty problem that compare favourably against other competitive algorithms from the literature.

Existing work on the Rotated Klee-Minty problem has an established process for performance comparison that is distinct from elsewhere in the literature. For each algorithm, 1000 bootstrapped samples are generated from 15 run sets. A brief overview of the bootstrapping process is given in Section A.2 of the appendix. Target sets are fixed according to Eq. (5.2) and the function g_Σ defined in Eq. (5.3) is used to evaluate

[5]Retrieved April 25, 2022, from `https://github.com/patsp/RotatedKleeMintyProblem/tree/ea_comparison/lcCMSA-ES`

against g-targets. Runtimes are measured as evaluation counts for either f or g, depending on the target set being evaluated against, before being divided by the problem dimension. Monotonicity is enforced during the bootstrapping process, giving plots comparable to those previously published [63]. Repeating, near-horizontal plateaus in some plots are indicative of the effects of enforcing the monotonic condition.

To allow a more direct and alternative comparison, data is generated for ECDF plots here without any bootstrapping. In addition, unlike previous results, our function evaluation counts are not scaled by dimension. To generate this data, 100 runs were performed for the Rotated Klee-Minty problem in each of the dimensions $n = 2, 3, 5, 10, 15$, and 20 for each of the four algorithms. Target sets for the ECDFs rely on the definition in Eq. (5.2) with f-target and g-target values evenly logarithmically spaced in the closed intervals $[10^0, 10^{-8}]$ and $[10^2, 10^{-6}]$, respectively, and runtimes are again measured as f-evals, g-evals, or $(f + g)$-evals, depending on the target set. In order to facilitate comparisons with published results on these particular algorithms, the value given by Eq. (5.3) is used for evaluating g-targets across all algorithms.

A run is terminated when $f(\boldsymbol{x})$ and $g_\Sigma(\boldsymbol{x})$ are both within 1.0e-8 of optimum values, represented by $f(\boldsymbol{x}^*)$ and $g_\Sigma(\boldsymbol{x}^*) = 0$, respectively. The lcCMSA-ES and ϵMAg-ES, following their default parameters, also terminate when more than $2n \cdot 10^4$ total $(f+g)$ function evaluations have been used. Starting points are initialized randomly using coordinates drawn independently and uniformly from the interval $[0, 5n^3]$.

Staggered ECDF plots are given in Figure 5.19 plotted against the sum of f- and g-evaluations. As for the staggered plots in Section 5.2, the combined performance is displayed for f-targets with fixed g-target of 10^0 (solid lines) and fixed g-target of 10^{-6} (dashed lines). Runs on the first target set are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_\Sigma \leq 10^0$ are satisfied, and runs on the second target set are considered successful on the i-th target if both $f(\boldsymbol{x}) \leq t_i^f$ and $g_\Sigma \leq 10^{-6}$ are satisfied.

The performance of lcCMSA-ES is given throughout by only a single line because both of the fixed g-targets are the first targets satisfied during its pre-processing step. The performance of EL-ES is seen to be broadly superior to all other algorithms

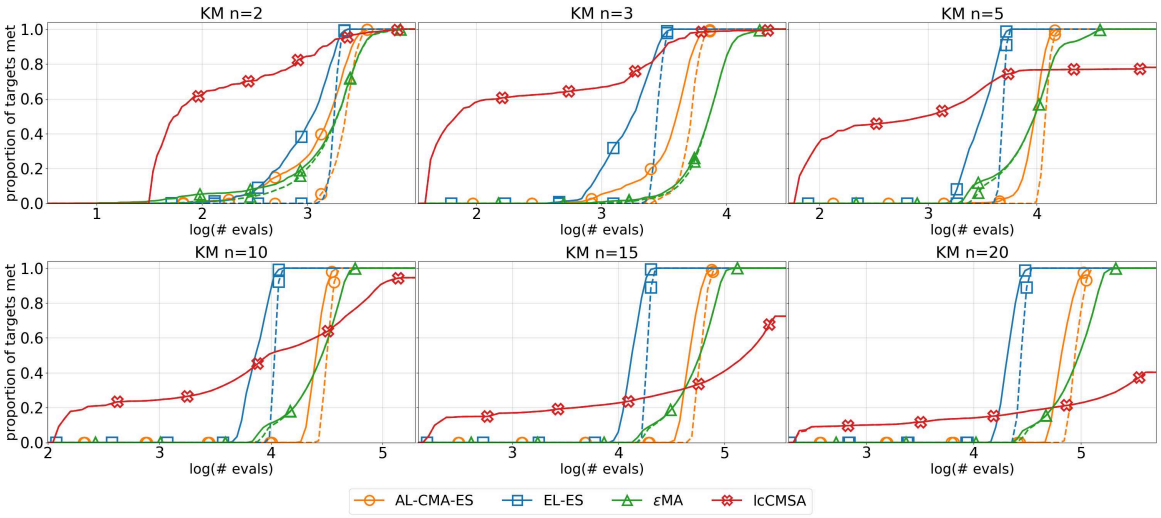

Figure 5.19: ECDF plots showing $(f + g)$-evals vs. f-targets for fixed g-target 10^0 (solid lines) and fixed g-target 10^{-6} (dashed lines) for the Rotated Klee-Minty problem in varying dimensions n. The x-axes are scaled to present as much detail as possible without obscuring data points.

for dimensions $n = 10$ and above, and is additionally superior on the most difficult targets for $n = 5$ and $n = 3$. The EL-ES is also strictly superior to the AL-CMA-ES on all problems. The lcCMSA-ES appears very competitive on easier targets and in smaller dimensions, but this deteriorates significantly with increasing n and it fails to reach all targets within the allocated budget for problems in dimension $n = 10$ and above.

Additional ECDF plots for the separate f- and g- targets are given in Figure 5.20, where as before, the top rows are associated with success on f-targets plotted against f-evaluation counts, while the bottom rows show success on g-targets plotted against g-evaluation counts. The value of the pre-processing step of the lcCMSA-ES is evident, as it is able to succeed on all g-targets for all problems within only a few hundred g-evaluations. The advantage of the EL-ES algorithm in higher dimensions appears mostly due to success on f-targets, as it is otherwise comparable to the ϵMAg-ES in satisfying g-targets for problems with $n = 5$ and above. On all problems, the EL-ES is also seen to strictly dominate the three other algorithms on the most difficult f-targets.

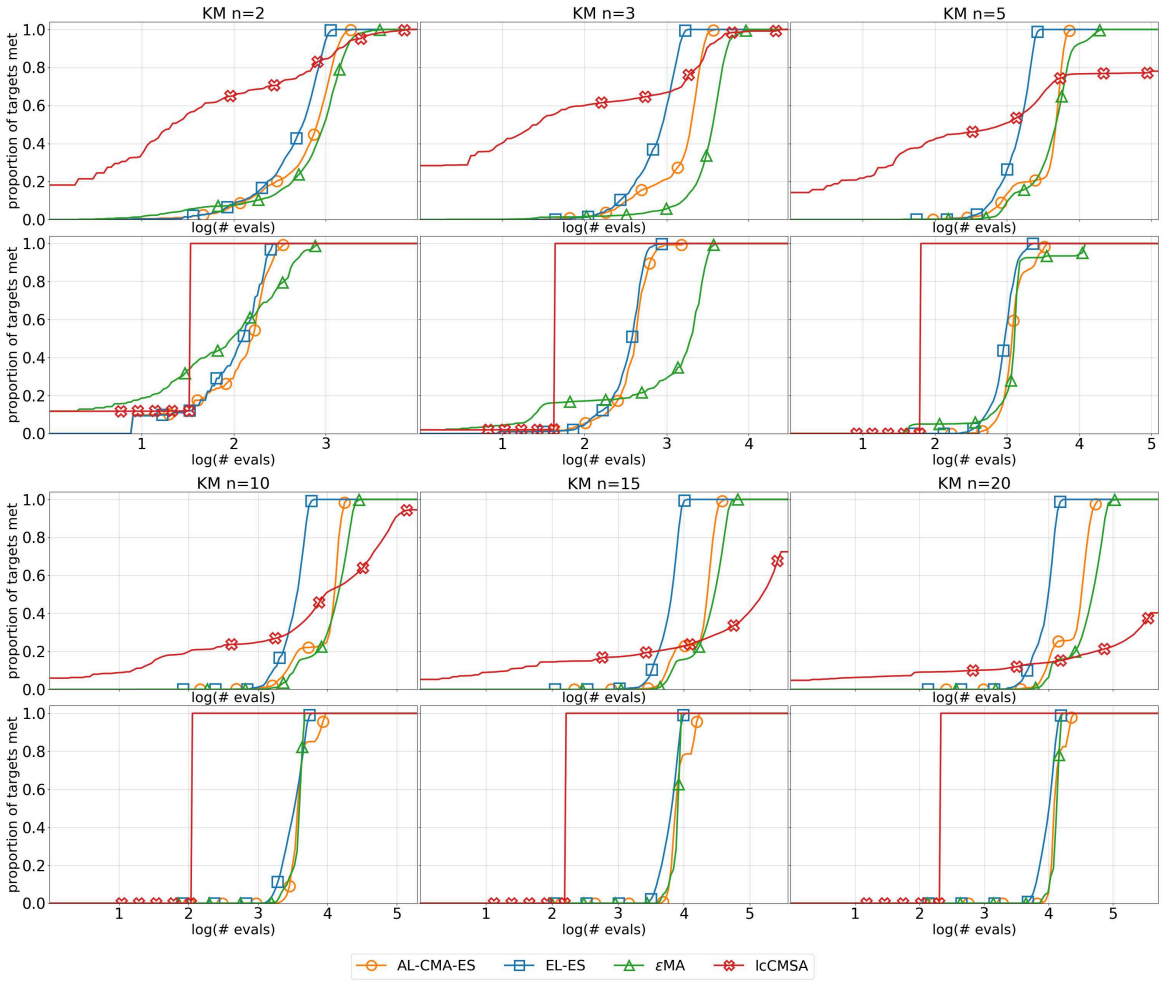

Figure 5.20: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom) for the Rotated Klee-Minty problem. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

It should be noted that in trial runs for generating data on the $n = 20$ Rotated Klee-Minty problem, the EL-ES was observed to fail in very rare cases ($< 1\%$ of runs), likely due to instability in the working set caused by the large number of constraints. These issues were not encountered while performing the 100 independent runs to generate the data used for the plots given above.

5.4.2 Summary for Rotated Klee-Minty

Across all tested dimensions for the Rotated Klee-Minty problem, the performance of the EL-ES algorithm is seen to largely out-perform three algorithms that employ various forms of covariance matrix adaptation for generating improved offspring, one of which was designed specifically for solving linear problems with linear constraints. The Exact Lagrangian approach also appears to scale much better with increasing dimension and number of constraints. This is potentially limited by rare cases of failure in the $n = 20$ case due to difficulties in managing the working set.

Chapter 6

Discussion and future work

We have proposed through this research a novel approach for constrained continuous optimization with stochastic black-box algorithms by adapting for the first time an exact Lagrangian method from numerical optimization for use with evolution strategies. Previous study and subsequent proposal of the augmented Lagrangian approach (AL-ES) with a $(1 + 1)$-ES showed that using the Lagrange multiplier update rule from the method of multipliers alongside careful adaptation of the penalty coefficient could lead to log-linear convergence on certain sphere and ellipsoid problems with a single constraint. This desirable behaviour is understood as resulting from adapting the Lagrangian parameters in order to balance progress of the evolution strategy in the constrained and unconstrained subspaces, as demonstrated by single-step analysis.

Extensions of the AL-ES method [17, 16, 18, 38] exhibit good performance on certain problems, but our experimental investigation highlighted that it appears necessary to include covariance matrix adaptation in order to arrive at a widely applicable algorithm. Without CMA, the ill-conditioning of the augmented Lagrangian was often too significant to allow convergence, primarily due to the update rules for the Lagrangian's parameters. Additionally, from the discussion in Section B.5, the multiplier update used by the AL-ES is seen to be a form of gradient ascent for the dual function with step size determined by the penalty coefficient ω. This raises two potential concerns: first, that the implicit maximization of the dual is based on only first-order derivative information, and second, that the adaptation of the penalty coefficient is performed without regard for its role as the step size.

A particular example of difficulty was demonstrated, by showing that even well-conditioned spherical objective functions with linear constraints that create narrow

feasible regions (NFR) result in much slower convergence (or even non-convergence) when using existing implementations of the AL-ES. In part, this is because of poor values arising from the multiplier update. In the dual formulation, this can be interpreted as a result of the multiplier update not accounting for second-order features of the dual function, like curvature and ill-conditioning.

In Section 4.4 and in connection with Section B.5, we defined a multiplier update rule that did include second-order information, based on quasi-Newton maximization of the dual. One observed difficulty was that implementing the rule would require knowledge of the derivatives of the objective and constraint functions. Instead, we derived a multiplier update rule that only relies on first-order information. As such, the proposed expression for the multiplier does not account for any ill-conditioning in the objective function, but crucially includes information about the constraints and the correlations between them. This rule was justified in two complementary ways, both by application of the first- and second-order KKT conditions, as well as through step-size analysis on the Lagrangian. Exactly how this information should be included for use with stochastic algorithms like evolution strategies is not immediately straightforward.

Our approach was to adapt an exact Lagrangian penalty method from Fletcher that continuously defines Lagrange multipliers with respect to position in the search space, rather than as part of an external update rule. Doing so allows approximating constraint information that is then included within the Lagrangian function in a way that is usable by an evolution strategy, resulting in the EL-ES. By applying single-step analysis to this new method, and taken together with theoretical insight from the literature on numerical optimization, we showed that the multiplier update for the EL-ES rule balances the progress of the evolution strategy in the constrained and unconstrained subspaces, in a manner analogous to that of the original multiplier rule for the AL-ES.

In order to validate our proposed approach, experimental data was generated for multiple runs on spheres and moderately scaled ellipsoids with a single constraint. The results showed that the EL-ES was able to perform almost as well as the AL-CMA-ES on these archetypal problems, except where it performed significantly better

on NFR spheres with two constraints. This increased performance is in spite of the advantage given by improved offspring generation from CMA.

Results were also generated for varied scaling between objective and constraint functions, as well as for constraints with random orientation. In order for evaluation on the random constraints to be reliable and unbiased, an approach was proposed for generating active constraints based on the relationship between their normal vectors and the gradient of the objective function. This process makes use of the KKT conditions that describe an optimum point in order to allow as much freedom in selecting constraint orientations as possible while guaranteeing a pre-selected point will become the optimum. These constraints were additionally guaranteed to not merely be weakly active, and instead would have an associated non-zero Lagrange multiplier. Performance of the EL-ES on all random constraint problems was observed to be superior to that of either of the existing AL implementations, and even performed better than the active-CMA-ES on problems with increasing number of constraints and dimension. On problems with the number of generated constraints equal to the dimension for $n = 10$ and $n = 20$, the EL-ES converged to the optimum with 2-4 times fewer function evaluations than the other algorithms.

Additional experimental results on benchmarks from the literature and the Rotated Klee-Minty problem showed that the EL-ES is also competitive on certain problems beyond archetypal spheres and ellipsoids, even when compared against algorithms using CMA for generating offspring. On the standard benchmark problems, the number of function evaluations required for convergence using the EL-ES was smaller than that required for the AL-ES by approximately a factor of 10. The overall performance of the exact Lagrangian approach was also observed to never be far behind the methods using CMA. The benchmark problems were selected to match those previously used [6, 38] for evaluating evolution strategies on constrained optimization problems, allowing the EL-ES to be evaluated in light of those published results.

The Rotated Klee-Minty problem was selected as an additional benchmark that has previously been used for evaluating performance of constrained optimization algorithms, including the ϵMA-ES and lcCMSA-ES [59, 117, 63]. In addition to having published results for the Rotated Klee-Minty problem, the ϵMA-ES has been

favourably compared to leading algorithms when evaluated against the CEC 2017 benchmark problems [60], and the lcCMSA-ES has published very encouraging results using an early variant of the BBOB COCO framework for constrained optimization [117]. In our results on the Rotated Klee-Minty problem, the EL-ES outperformed both other algorithms on the most difficult evaluated targets in all dimensions but $n = 2$ and $n = 20$. In the smaller case, the EL-ES was roughly comparable to the performance of the lcCMSA-ES algorithm. In the larger case, the EL-ES converged with 2-4 times fewer function evaluations than the other algorithms. In rare cases, the EL-ES may have difficulties converging for the $n = 20$ case due to instability in the working set caused by the large number of constraints.

From this collection of encouraging empirical comparisons together with the justifications given by both step-size analysis and consideration of the KKT conditions, the exact Lagrangian method for evolution strategies is seen to offer an attractive approach for continuous constrained black-box optimization.

Future work

An immediate and obvious improvement for the EL-ES would be the inclusion of CMA for generating offspring, allowing for faster convergence on resulting Lagrangian functions in spite of a certain degree of ill-conditioning. This needs to be done with some care, as both the working set management and the approximation of Lagrange multiplier terms assume to some extent that offspring are sampled isotropically from the search space. If this can be properly accounted for, then an EL-CMA-ES implementation could be an algorithmic approach with very promising properties.

A limitation of the current implementation of the EL-ES relates to reliably managing the working set, as on some problem instances the current approach appears to be insufficient. In the case of the Rotated Klee-Minty problem in high ($n = 20$) dimension, there are a large number of similar constraints active near the optimum, and in rare instances the working set will oscillate between adding and removing a subset of constraints. This leads to a form of zigzagging, which in the case of an evolution strategy can result in poor adaptation of the step size. Similarly, the problem G10 has been previously used for evaluating evolution strategies [6, 38], yet the EL-ES

progresses far too slowly to result in reliable convergence. A key feature of this problem is the relatively large number of constraints with different weights. Deriving a more reliable method of working set management would allow broader application of the EL-ES.

Application of the EL-ES approach, either in its current form or using future improvements, should be evaluated on the COCO bbob-constrained benchmark. Comparative results for the AL-CMA-ES using this benchmark have recently been published by Dufossé and Atamna [36]. Similar benchmark performance could be investigated for the EL-ES with the addition of surrogate models for problems with expensive constraint evaluations, similar to the evaluation performed by Dufossé and Hansen [38] for the AL-CMA-ES.

For work farther in the future, it would be beneficial to further investigate a broader range of approaches previously used in numerical optimization, and consider how they might be applied to stochastic algorithms like evolution strategies. The success of the EL-ES is evidence that there are approaches in the literature that might benefit from a second look. Work like that of Glad and Polak [48], which expands on the work of Fletcher for exact Lagrangians, should in particular be investigated.

Finally, it would be interesting to consider whether extension of the Markov chain analysis used to originally give convergence results for variations of the AL-ES [16, 17, 18] would be possible for the case of the EL-ES.

Appendices

Appendix A

Experimental details

A.1 Function definitions

Explicit function definitions are given here for the selected literature benchmarks used in Section 5.3 for experimentally comparing the performance of the EL-ES with other evolution strategies for constrained optimization.

Problem TR2

(Kramer & Schwefel [72])
Minimize

$$f(\boldsymbol{x}) = x_1^2 + x_2^2$$

subject to

$$g_1(\boldsymbol{x}) = 2 - x_1 - x_2 \leq 0.$$

The lone constraint is active at $\boldsymbol{x}^* = [1, 1]$ with $f(\boldsymbol{x}^*) = 2$. The starting point is fixed as $\boldsymbol{x} = [50, 50]$.

Problem S240

(Schwefel [109])
Minimize

$$f(\boldsymbol{x}) = -\sum_{i=1}^{5} x_i$$

subject to

$$g_1(\boldsymbol{x}) = -50000 + \sum_{i=1}^{5}(9+i)x_i \leq 0$$

and lower bound constraints $0 \leq x_i$ for $i = 1, \ldots, 5$, with no upper bound constraints. Constraint g_1, along with the lower bounds on x_2, x_3, x_4, x_5, are all active at $\boldsymbol{x}^* = [5000, 0, 0, 0, 0]$ with $f(\boldsymbol{x}^*) = -5000$. The starting point is fixed as $\boldsymbol{x} = [250, 250, 250, 250, 250]$.

Problem S241

(Schwefel [109])
Minimize

$$f(\boldsymbol{x}) = -\sum_{i=1}^{5} ix_i$$

subject to

$$g_1(\boldsymbol{x}) = -50000 + \sum_{i=1}^{5}(9+i)x_i \leq 0$$

and lower bound constraints $0 \leq x_i$ for $i = 1, \ldots, 5$, with no upper bound constraints. Constraint g_1, along with the lower bounds on x_1, x_2, x_3, x_4, are all active at $\boldsymbol{x}^* = [0, 0, 0, 0, 25000/7]$ with $f(\boldsymbol{x}^*) = -125000/7$. The starting point is fixed as $\boldsymbol{x} = [250, 250, 250, 250, 250]$.

Problem RosenbrockParcel

(Rosenbrock [102], Fletcher [41])
Minimize

$$f(\boldsymbol{x}) = -x_1 x_2 x_3$$

subject to

$$g_1(\boldsymbol{x}) = x_1 + 2x_2 + 2x_3 - 72 \leq 0$$

and bound constraints $0 \leq x_i \leq 42$ for $i = 1, \ldots, 3$. Constraint g_1 is active at $\boldsymbol{x}^* = [24, 12, 12]$ with $f(\boldsymbol{x}^*) = -3456$.

Problem G04 (HB)

(Himmelblau [65])
Minimize

$$f(\boldsymbol{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

subject to

$$\begin{aligned}
g_1(\boldsymbol{x}) &= h_1(\boldsymbol{x}) - 92 \leq 0 \\
g_2(\boldsymbol{x}) &= -h_1(\boldsymbol{x}) \leq 0 \\
g_3(\boldsymbol{x}) &= h_2(\boldsymbol{x}) - 110 \leq 0 \\
g_4(\boldsymbol{x}) &= 90 - h_2(\boldsymbol{x}) \leq 0 \\
g_5(\boldsymbol{x}) &= h_3(\boldsymbol{x}) - 25 \leq 0 \\
g_6(\boldsymbol{x}) &= 20 - h_3(\boldsymbol{x}) \leq 0
\end{aligned}$$

where

$$\begin{aligned}
h_1(\boldsymbol{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \\
h_2(\boldsymbol{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \\
h_3(\boldsymbol{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4
\end{aligned}$$

and bound constraints

$$78 \leq x_1 \leq 102$$
$$33 \leq x_2 \leq 45$$

and $27 \leq x_i \leq 45$ for $i = 3, 4, 5$. Both g_1 and g_6 are active, along with the lower bounds on x_1 and x_2 and the upper bound on x_4, at

$$\boldsymbol{x}^* \approx [78, 33, 29.99525602, 45, 36.77581290]$$

with $f(\boldsymbol{x}^*) \approx -30665.53867178$.

Problem G06

(Floudas & Pardalos [46])
Minimize

$$f(\boldsymbol{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to

$$g_1(\boldsymbol{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$
$$g_2(\boldsymbol{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

and bound constraints

$$13 \leq x_1 \leq 100$$
$$0 \leq x_2 \leq 100.$$

Both g_1 and g_2 are active at $\boldsymbol{x}^* \approx [14.095, 0.84296]$ with $f(\boldsymbol{x}^*) \approx -6961.81387558$.

Problem G07

(Hock & Schittkowski [66])
Minimize

$$\begin{aligned} f(\boldsymbol{x}) = x_1^2 + x_2^2 + x_1 x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\ + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 \\ + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \end{aligned}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leq 0 \\ g_2(\boldsymbol{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\boldsymbol{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(\boldsymbol{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\ g_5(\boldsymbol{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_6(\boldsymbol{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1 x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(\boldsymbol{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_8(\boldsymbol{x}) &= (x_1 - 8)^2 + 4(x_2 - 4)^2 + 6x_5^2 - 2x_6 - 60 \leq 0 \end{aligned}$$

and bound constraints

$$-10 \leq x_i \leq 10$$

for $i = 1, \ldots, 10$. Each of $g_1, g_2, g_3, g_5, g_6, g_7$ are active at

$$\begin{aligned} \boldsymbol{x}^* \approx &[2.17199638, 2.36368294, 8.77392572, 5.09598444, 0.99065475, \\ &1.43057395, 1.32164423, 9.82872583, 8.28009174, 8.37592676] \end{aligned}$$

with $f(\boldsymbol{x}^*) \approx 24.30620906$.

Problem G09

(Hock & Schittkowski [66])
Minimize

$$f(\boldsymbol{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\ g_2(\boldsymbol{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\ g_3(\boldsymbol{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\ g_4(\boldsymbol{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \end{aligned}$$

and bound constraints

$$-10 \leq x_i \leq 10$$

for $i = 1, \ldots, 7$. Both of g_1 and g_4 are active at

$$\begin{aligned} \boldsymbol{x}^* \approx &[2.33049932, 1.95137235, -0.47754169, 4.36572630, \\ &-0.62448696, 1.03813102, 1.59422672] \end{aligned}$$

with $f(\boldsymbol{x}^*) \approx 680.63005737$.

A.2 Bootstrapping

In order to simulate restarts and allow comparisons between algorithms with failures on certain targets, the COCO benchmark [57] uses a method inspired by statistical *bootstrapping* for extending results in a viable way without requiring excessive experimental runs. Instead, a small number of runs (COCO recommends 15) are performed and these are used to generate a larger number of bootstrapped results. The

operation for generating a bootstrapped runtime proceeds in the same way for each target in the fixed set: if at least one of the experimental runs succeeded, then the set of runs is drawn from uniformly at random and with replacement until a success is found. The bootstrapped runtime is then the sum of the runtime of the successful run and the runtimes for all unsuccessful runs, if any. A full set of bootstrapped samples is generated in this way for each target. The outline of this process is given in pseudo-code by Algorithm A.1. To represent this as an ECDF graph, the proportion of successful targets among the bootstrapped samples is plotted against the runtime.

Algorithm A.1 Generating bootstrapped results for ECDFs

Require: Indexed sets of runs R and targets T, max runtime m, bootstrap sample size b

Initialize B of size b
for $i = 1 \to |T|$ **do**
    for $j = 1 \to |B|$ **do**
        $B_{i,j} = 0$
        while no successful run found **do**
            $k \leftarrow \mathcal{U}[1, |R|]$ ▷ Sample from random uniform distribution
            if run R_k succeeded for target T_i **then**
                $B_{i,j} \leftarrow B_{i,j} + R_k(T_i)$ ▷ Add sample's runtime on success
            else
                $B_{i,j} \leftarrow B_{i,j} + m$ ▷ Add max runtime on failure
            end if
        end while
    end for
end for
return B

This differs from the process used in previous work on the Klee-Minty problem [63, 117] which additionally enforces monotonicity[1]. In this process, each of the bootstrapped samples for a fixed target is considered as part of a contiguous virtual run, where runtimes for subsequent (more difficult) targets are not permitted to be less than runtimes for prior (easier) targets. More concretely, for the set of bootstrapped samples $B_{t,i}$ corresponding to target t and with $i = 1, \ldots$ indexing the samples, the authors enforce the condition that $B_{t,i} = \max(B_{t,i}, B_{t-1,i})$. This has the

[1]Based on the authors' code, retrieved April 25, 2022, from `https://github.com/patsp/RotatedKleeMintyProblem/tree/ea_comparison/lcCMSA-ES`

effect of propagating forward the runtime of the worst run on simpler targets to later, more difficult targets, regardless of the likelihood that the target would result in that runtime. In published ECDF plots, this effect is visualized by horizontal plateaus followed by sharp vertical increases, that overall shift the apparent performance of algorithms to appear worse. In the experimental results throughout Chapter 5, we aim to largely avoid these issues by relying on raw data instead of using a bootstrapping process.

Appendix B

Theory of optimization

The theory of numerical optimization is well covered by many authors. In order to summarize key concepts that are important for understanding and justifying my own work, I provide here an amalgam of standard work taken from the literature and adapted for brevity and notational coherence, based primarily on the works of Fletcher [44, 43], Bertsekas [27, 28], and Nocedal and Wright [128]. Throughout the following I will use the ∇ and ∇^2 operators to refer to the gradient and Hessian matrix, respectively defined as

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix}, \quad \nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} \end{bmatrix}$$

where the variable under differentiation is understood. Where clarity is required, subscripts such as $\nabla_{\boldsymbol{x}}$ and $\nabla^2_{\boldsymbol{x}}$ will be used. In some expressions that benefit from notational simplicity, I also write f, g, and related matrices and derivatives with the understanding that they are to be evaluated at the current point $\boldsymbol{x}$, unless otherwise specified.

B.1 Unconstrained optimization

Optimization is the study of algorithms for finding extrema. For continuous numerical optimization, this can be understood through the simple example given by an unconstrained optimization problem where we are given a function

$$f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$$

and desire a solution $\boldsymbol{x}^*$ that satisfies either

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{or} \quad \max_{\boldsymbol{x}} f(\boldsymbol{x})$$

across all $\boldsymbol{x}$ in the domain $\mathbb{R}^n$. Since one optimization problem is identical to the other up to the sign on f, we will throughout refer to *minimization* as the optimization operation, without loss of generality.

Without restrictions on the function f, it may not be realistic to determine a *global minimum* across the entire domain, so we are frequently satisfied with finding a *local minimum*. This is a solution $\boldsymbol{x}^*$ satisfying $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all other points in the neighbourhood $\mathcal{N}_r$ with $||\boldsymbol{x}^* - \boldsymbol{x}|| < r$ for some $r > 0$, and is a *strict* local minimum if the inequality is made strict. With mild restrictions on f, a local minimum within a neighbourhood can be characterized with the following propositions.

Proposition B.1 (First-order Necessary)**.** *Let f be continuously differentiable within neighbourhood $\mathcal{N}_r = \{\boldsymbol{x} : ||\boldsymbol{x}^* - \boldsymbol{x}|| < r\}$ with $r > 0$, having associated strict local minimum $\boldsymbol{x}^*$. Then the gradient satisfies*

$$\nabla f(\boldsymbol{x}^*) = 0.$$

Proposition B.2 (Second-order Necessary)**.** *Let f be twice continuously differentiable in $\mathcal{N}_r$, having associated local minimum $\boldsymbol{x}^*$. Then the Hessian satisfies*

$$\boldsymbol{z}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{z} \geq 0$$

for all $\boldsymbol{z} \in \mathcal{N}_r$ and is therefore positive semidefinite.

Proposition B.3 (Second-order Sufficient)**.** *Let f be twice continuously differentiable*

in $\mathcal{N}_r$ around point $\boldsymbol{x}^*$. If both $\nabla f(\boldsymbol{x}^*) = 0$ and the Hessian satisfies

$$\boldsymbol{z}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{z} > 0$$

for all $\boldsymbol{z} \in \mathcal{N}_r$ and is therefore positive definite, then $\boldsymbol{x}^*$ is a strict local minimum.

In the particular case that the neighbourhood $\mathcal{N}_r$ is extended to include the entire domain, then these conditions refer to global rather than local solutions. The implications of these three propositions taken together typically play a foundational role in the design of any numerical optimization algorithm. In the simplest cases, it may be possible to even solve the problem analytically simply by solving for the conditions placed on the first and second derivatives of f. For anything more complicated than these relatively simple problems, an *iterative algorithm* may be used that takes place over multiple steps. In the k-th iteration of such an algorithm, an estimate $\boldsymbol{x}^{(k)}$ is generated using local or historical information, and the aim is to have the sequence converge as $\{\boldsymbol{x}^{(k)}\} \to \boldsymbol{x}^*$. The domain of f contains the estimates $\boldsymbol{x}^{(k)}$ and is referred to as the *search space*.

A fundamental example is that of Newton's method, which uses derivative information together with an initial estimate $\boldsymbol{x}^{(0)}$ to attempt to converge to a local minimum. In each iteration of the algorithm, a local quadratic approximation is formed from the truncated Taylor series about $\boldsymbol{x}^{(k)}$ as

$$f^{(k)}(\boldsymbol{y}) = f(\boldsymbol{x}^{(k)}) + \boldsymbol{y}^{\mathrm{T}} \nabla f(\boldsymbol{x}^{(k)}) + \frac{1}{2} \boldsymbol{y}^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}^{(k)}) \boldsymbol{y}$$

where $\boldsymbol{y} = 0$ corresponds to the point $\boldsymbol{x}^{(k)}$ by a shift of origin, and this function is minimized instead. By the first-order necessary condition of Proposition B.1, it must hold that a local minimum of the quadratic approximation satisfies $\nabla f^{(k)} = 0$, so solving

$$\begin{aligned} \nabla f(\boldsymbol{x}^{(k)}) + \nabla^2 f(\boldsymbol{x}^{(k)}) \boldsymbol{y} &= 0 \\ \boldsymbol{y} &= -\nabla^2 f(\boldsymbol{x}^{(k)})^{-1} \cdot \nabla f(\boldsymbol{x}^{(k)}) \end{aligned}$$

gives a candidate point $\boldsymbol{y}$ relative to $\boldsymbol{x}$ by the shift of origin. If $\nabla^2 f$ is positive definite

then it is also invertible, and this additionally implies that the point $\boldsymbol{y}$ is a local minimum for the quadratic approximation by the second-order sufficient condition of Proposition B.3. Newton's method is then to use the update formula

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \boldsymbol{y} \\ &= \boldsymbol{x}^{(k)} - \nabla^2 f(\boldsymbol{x}^{(k)})^{-1} \cdot \nabla f(\boldsymbol{x}^{(k)}) \end{aligned} \tag{B.1}$$

in order to generate the next step of the algorithm in the search space.

B.2 Constrained optimization

Constrained optimization expands on the ideas in Section B.1 by placing limitations on the domain of f where a solution is acceptable. These constraints are commonly expressed as a combination of equalities and inequalities that must be satisfied along with the objective function. Beginning with the simpler case of the *equality constrained problem* (ECP), we ask for a solution satisfying

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) = 0. \end{aligned} \tag{ECP}$$

We are again minimizing the objective function $f : \mathbb{R}^n \to \mathbb{R}$, but have now added a set of m equality constraint functions $g_i : \mathbb{R}^n \to \mathbb{R}$ indexed by $i = 1, \ldots, m \leq n$, or equivalently the single vector function $g : \mathbb{R}^n \to \mathbb{R}^m$, that must be satisfied. The solution for the ECP is the point $\boldsymbol{x}^*$ with minimal value $f(\boldsymbol{x}^*)$ among all $\{f(\boldsymbol{x}) : g(\boldsymbol{x}) = 0\}$. Any point that satisfies all of the given constraints in this way is said to be *feasible*, while a point that violates one or more of the constraints is said to be *infeasible*. We will specifically refer to the *unconstrained minimum* of the objective function f to distinguish from the *constrained minimum* or simply *minimum*, which is that point solving the ECP that minimizes f among all feasible solutions.

If the constraint functions under consideration are differentiable in $\mathcal{N}_r$, then their first-order behaviour can be described by writing the Jacobian

$$\boldsymbol{J} = [\nabla g_1, \nabla g_2, , \ldots, \nabla g_m] \tag{B.2}$$

as an $n \times m$ matrix, with columns consisting of the constraint normals. In the special case that the constraints are all linear, this means the equality conditions can be written as

$$g(\boldsymbol{x}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c} = 0$$

for some constant vector $\boldsymbol{c}$. If the constraint normals are linearly independent in a neighbourhood of the constrained optimum, then $\boldsymbol{J}$ has full rank there. This case proves to be important for much of what follows, as do the following two important definitions.

Definition B.1 (Tangent cone)**.** Let $\hat{\boldsymbol{x}}$ be a feasible point and $\{\boldsymbol{x}^{(k)}\} \to \hat{\boldsymbol{x}}$ be any infinite sequence of feasible points approaching $\hat{\boldsymbol{x}}$. Then vector $\boldsymbol{s}$ is a *feasible direction* if there is also a sequence of positive scalars $\delta^{(k)} \to 0$ such that

$$\delta^{(k)}\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}$$

with

$$\lim_{k\to\infty} \boldsymbol{s}^{(k)} = \boldsymbol{s}.$$

The set of all feasible directions so defined is the *tangent cone* at $\hat{\boldsymbol{x}}$.

Definition B.2 (LICQ)**.** The *linear independence constraint qualification* (LICQ) assumes the linear independence of the (active) constraint normals at $\boldsymbol{x}^*$, and is alternately referred to as the regularity or quasi-regularity assumption of point $\boldsymbol{x}^*$.

The notion of an active constraint will be made explicit in Section B.3, but when dealing only with equality constraints it suffices to note that all constraints are active.

It will be helpful to consider the inverse of the matrix $\boldsymbol{J}$ under the assumption of full rank, even in situations where $m < n$, so we rely on the notion of the generalized

Moore-Penrose inverse written as

$$\boldsymbol{J}^{+} = \left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{\mathrm{T}}.$$

This generalized inverse is rank m, and satisfies $\boldsymbol{J}^{+}\boldsymbol{J} = \boldsymbol{I}$. Combining the Jacobian and its generalized inverse gives the projection matrix

$$\begin{aligned}\boldsymbol{P} &= \boldsymbol{J}\boldsymbol{J}^{+} = \boldsymbol{J}^{+\mathrm{T}}\boldsymbol{J}^{\mathrm{T}} \\ &= \boldsymbol{J}\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{\mathrm{T}}\end{aligned} \tag{B.3}$$

which maps a vector into the *constrained subspace* spanned by the constraint normals ∇g_i. To see this, recall that the vector after projection $\boldsymbol{P}\boldsymbol{x}$ is in the span of the columns of $\boldsymbol{J}$ if and only if it is a linear combination of those columns ∇g_i. Writing the projected vector using Eq. (B.3) and collecting terms therefore gives

$$\begin{aligned}\boldsymbol{P}\boldsymbol{x} &= \boldsymbol{J}\left[\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{x}\right] \\ &= \boldsymbol{J}\cdot\boldsymbol{r}\end{aligned}$$

for some vector $\boldsymbol{r} \in \mathbb{R}^m$, and whose elements r_i give the coefficients of the desired linear combination. The complement of this projection is $(\boldsymbol{I} - \boldsymbol{P})$ which maps into the *unconstrained manifold*, an affine manifold (vector subspace with possibly shifted origin). These are complementary in the sense that any point $\boldsymbol{x}$ in the search space can be written as a combination

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{P}\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x} \\ &= \boldsymbol{J}\boldsymbol{r} + \boldsymbol{s}\end{aligned} \tag{B.4}$$

of its constrained and unconstrained components.

As in the unconstrained case, it is possible to characterize a solution to the constrained problem. Given f and g both continuously differentiable, let $\boldsymbol{x}^*$ be a constrained local minimum with respect to f and all g_i for an ECP and consider the set of first-order

feasible variations

$$V = \{\boldsymbol{v} \in \mathbb{R}^n \setminus \mathbf{0} : \boldsymbol{v}^{\mathrm{T}} \nabla g_i(\boldsymbol{x}^*) = 0, \forall\ i\}. \tag{B.5}$$

This set is a recurring concept, and it is helpful to interpret it in different ways. It is the set containing those vectors which remain feasible with respect to a linear approximation of the constraints at $\boldsymbol{x}^*$. It is also the set of non-zero directions $\boldsymbol{v}$ which are orthogonal to the normals of the hyperplanes tangential to the constraint boundaries at $\boldsymbol{x}^*$.

Proposition B.4. *Under the LICQ of Definition B.2, the tangent cone of Definition B.1 is equal to the set of feasible variations V from Eq.* (B.5)*.*

From this proposition, we can also observe that the condition on membership in V is equivalent to requiring $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{v} = 0$. Roughly speaking, if we consider a small step $\boldsymbol{s}$ from $\boldsymbol{x}^*$ that remains feasible, a first-order Taylor expansion must therefore satisfy

$$\begin{aligned} g(\boldsymbol{x}^* + \boldsymbol{s}) &= g(\boldsymbol{x}^*) + \boldsymbol{s}^{\mathrm{T}} \nabla g(\boldsymbol{x}^*) \\ &= 0 \end{aligned}$$

where the additional Taylor terms vanish in the limit with respect to $\boldsymbol{s}$. For this equality to hold, any feasible step $\boldsymbol{s}$ must be in the direction of a feasible variation, and so $\boldsymbol{s}^{\mathrm{T}} \nabla g(\boldsymbol{x}^*) = 0$. Finally, the definition of V is equivalent to those vectors that constitute the entirety of the unconstrained manifold of Eq. (B.4) defined by the Jacobian at $\boldsymbol{x}^*$.

Using the equivalence between incremental feasible steps and the set V of feasible variations together with the fact that $\boldsymbol{x}^*$ is a constrained local minimum by assumption, it must be the case that $\boldsymbol{s}^T \nabla f(\boldsymbol{x}^*) \geq 0$ as well; otherwise, it would be possible to take a feasible step with f decreasing. Stated concisely then, at the constrained minimum $\boldsymbol{x}^*$ there can be no vectors $\boldsymbol{s}$ where both conditions

$$\boldsymbol{s}^{\mathrm{T}} \nabla g(\boldsymbol{x}^*) = 0, \tag{B.6}$$

$$\boldsymbol{s}^T \nabla f(\boldsymbol{x}^*) < 0 \tag{B.7}$$

are satisfied: moving along any vector $\boldsymbol{s}$ from $\boldsymbol{x}^*$ must either result in a non-decreasing change in the objective function or a change in the feasibility, or both. This permits proving a fundamental result for constrained optimization.

Theorem B.5 (First-order necessary condition)**.** *At a constrained minimum $\boldsymbol{x}^*$ with f and g_i continuously differentiable, and with linearly independent $\nabla g_i(\boldsymbol{x}^*)$ (LICQ), the gradient of the objective function is equal to a linear combination of the constraint function gradients evaluated at $\boldsymbol{x}^*$ as*

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) &= -\sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}^*) \\ &= -\boldsymbol{J}\boldsymbol{\lambda}. \end{aligned} \tag{B.8}$$

Proof. If we view the gradient $\nabla f(\boldsymbol{x}^*)$ in its projected form similar to Eq. (B.4), we can write

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) &= \boldsymbol{P}\nabla f(\boldsymbol{x}^*) + (\boldsymbol{I} - \boldsymbol{P})\nabla f(\boldsymbol{x}^*) \\ &= -\boldsymbol{J}\boldsymbol{\lambda} + \boldsymbol{s} \end{aligned} \tag{B.9}$$

as a sum of components in two distinct subspaces. Since we are assuming $\boldsymbol{x}^*$ is optimal, this implies both Eqs. (B.6) and (B.7) are satisfied. We will prove that these equations are satisfied if and only if $\boldsymbol{s} = 0$ in Eq. (B.9), and therefore that Eq. (B.8) is a necessary condition for $\boldsymbol{x}^*$ being optimal.

First, if there are no vectors satisfying both Eqs. (B.6) and (B.7), then $\boldsymbol{s}$ must also be 0. To see this, first note that because it lies in the unconstrained manifold by the decomposition by projection matrix $\boldsymbol{P}$ in Eq. (B.9), the vector $\boldsymbol{s}$ must satisfy $\boldsymbol{s}^{\mathrm{T}}\nabla g(\boldsymbol{x}^*) = \boldsymbol{J}\boldsymbol{s} = 0$ and is orthogonal to any vectors projected into the constrained subspace. Now the claim is proven by contradiction: assume that $\boldsymbol{s}$ is nonzero, then the direction $-\boldsymbol{s}$ immediately satisfies Eq. (B.6) as it lies in the unconstrained manifold. After applying the inner product with the expression for the gradient in

Eq. (B.9), we also have

$$\begin{aligned}(-\boldsymbol{s})^{\mathrm{T}}\nabla f(\boldsymbol{x}^*) &= -\boldsymbol{t}^{\mathrm{T}}\boldsymbol{J}\boldsymbol{\lambda} - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{s} \\ &= -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s} < 0,\end{aligned}$$

which shows that $\boldsymbol{s}$ must also satisfy Eq (B.7) and give a direction in which f decreases, contradicting the assumption that there are no vectors satisfying both conditions.

From the other side, if $\boldsymbol{s} = 0$ then there can be no vectors satisfying the conditions of both Eqs. (B.6) and (B.7). This is trivially true since $\boldsymbol{s} = 0$ implies that the gradient $\nabla f(\boldsymbol{x}^*) = -\boldsymbol{J}\boldsymbol{\lambda}$ is a linear combination of the constraint normals, thus any $\boldsymbol{u}$ satisfying $\boldsymbol{u}^{\mathrm{T}}(-\boldsymbol{J}\boldsymbol{\lambda}) = 0$ immediately violates Eq. (B.7). ■

From this proof we see that under certain assumptions, at a constrained minimum $\boldsymbol{x}^*$ it will always be possible to express the gradient of the objective function as a linear combination of the gradients of the constraints. The coefficients in the vector $\boldsymbol{\lambda}$ determine the linear combination and their existence is a *necessary* condition for a point to be a constrained optimum. These coefficients are the *Lagrange multipliers* and they play an important role in many approaches to constrained optimization.

B.3 Extending to inequalities

If the constraints are defined with inequalities instead of equalities, then we have the inequality constrained problem (ICP) given by

$$\begin{aligned}\min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0.\end{aligned} \tag{ICP}$$

The major difference is of course how the constraints are treated: feasible points are no longer only at the intersection of the constraint boundaries, and the number of constraints m may be larger than the search space dimension n without violating the LICQ. The notion of a feasible point in this case is extended to those satisfying the

constraint inequalities, and as with the ECP an infeasible point will have a positive constraint value $g_i(\boldsymbol{x}) > 0$ for some i. The Jacobian $\boldsymbol{J}$ and projection matrix $\boldsymbol{P}$ now also refer to the constraint boundaries, which can be thought of as the set of points that satisfy a particular constraint g_i as an equality.

Inequality constraints also necessitate introducing the concept of an *active* constraint. We say that constraint g_i is active at a point $\boldsymbol{x}$ if $g_i(\boldsymbol{x}) \geq 0$, and *inactive* otherwise. The *active set* is denoted by $\mathcal{A}$ and corresponds to the set of constraints (or equivalently, of constraint indices) that are active at $\boldsymbol{x}^*$, and the size of this set is limited under the LICQ by the dimension of the search space $|\mathcal{A}| \leq n$. In the most general setting, the types of constraints may be mixed and we have

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i \in \mathcal{I} \\ & g_j(\boldsymbol{x}) = 0, \quad j \in \mathcal{E}. \end{aligned} \tag{GCP}$$

This gives separate index sets $\mathcal{I}, \mathcal{E}$ for inequality and equality constraints, respectively. As it generalizes and encompasses the ECP and ICP cases, we refer to it simply as the general constrained optimization problem (GCP). This definition also overlaps significantly with that of a nonlinear programming problem (NLP) sometimes used in the literature for numerical optimization.

Since the constrained optimum must be feasible, the set of active constraints at $\boldsymbol{x}^*$ are those that are satisfied as equalities

$$\mathcal{A} = \{i : g_i(\boldsymbol{x}^*) = 0,\ i \in \mathcal{I} \cup \mathcal{E}\}$$

which necessarily includes all equality constraints. Since $g_i(\boldsymbol{x}) \leq 0$ indicates feasibility for $i \in \mathcal{I}$, the definition of the set of feasible variations V of Eq. (B.5) is modified accordingly as

$$V = \{\boldsymbol{v} \in \mathbb{R}^n \setminus \boldsymbol{0} : \boldsymbol{v}^{\mathrm{T}} \nabla g_i(\boldsymbol{x}^*) \leq 0\ \forall i \in \mathcal{I},\ \boldsymbol{v}^{\mathrm{T}} \nabla g_j(\boldsymbol{x}^*) = 0\ \forall j \in \mathcal{E}\} \tag{B.10}$$

which are those directions which remain feasible with respect to the linearized constraints at $\boldsymbol{x}^*$. Similarly, Definition B.2 of the LICQ is expanded to require the linear independence of the constraint normals for constraints in $\mathcal{A}$ rather than only for equality constraints. Generalizing this way to include inequality constraints leads to re-stating the necessary conditions as the Karush-Kuhn-Tucker (KKT) conditions:

Theorem B.6 (KKT Necessary Conditions)**.** *If $\boldsymbol{x}^*$ is a local minimum of $f(\boldsymbol{x})$ that satisfies the equality constraints $g_j(\boldsymbol{x}) = 0$ for $j \in \mathcal{E}$ and inequality constraints $g_i(\boldsymbol{x}) \leq 0$ for $i \in \mathcal{I}$, and if additionally the constraints in $\mathcal{A}$ have linearly independent normals (LICQ), then there exists an optimal Lagrange multiplier vector $\boldsymbol{\lambda}^*$ where*

$$\nabla f(\boldsymbol{x}^*) + \sum_{i \in \mathcal{A}} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) = 0, \tag{B.11}$$

$$\lambda_i^* \geq 0, \qquad i \in \mathcal{I}, \tag{B.12}$$

$$\lambda_i^* g_i(\boldsymbol{x}^*) = 0, \qquad i \in \mathcal{I} \cup \mathcal{E}. \tag{B.13}$$

Proof. The condition of Eq. (B.11) is essentially the same as in Theorem B.5 which stated the same necessary condition for equality constraints, while Eqs. (B.12) and (B.13) are a result of now including inequalities. The necessity of requiring non-negative Lagrange multipliers in Eq. (B.12) can be seen by considering the opposite: if $\lambda_p < 0$, then as $|\mathcal{A} \cap \mathcal{I}| \leq n$ and the normals are linearly independent by assumption, it is possible to determine a vector $\boldsymbol{s}$ that is orthogonal to the constraint normals $\nabla g_i(\boldsymbol{x}^*)$ for all inequality constraints that are active but not associated with λ_p, so $i \in \mathcal{A},\ i \neq p$, yet for which $\boldsymbol{s}^{\mathrm{T}} \nabla g_p(\boldsymbol{x}^*) < 0$ ensuring $\boldsymbol{s} \in V$ according to Eq. (B.10). Then by Eq. (B.11) we can write

$$\boldsymbol{s}^{\mathrm{T}} \nabla f(\boldsymbol{x}^*) = -(\lambda_p) \left(\boldsymbol{s}^{\mathrm{T}} \nabla g_p(\boldsymbol{x}^*) \right) < 0.$$

Note that this inequality holds since both bracketed terms are themselves negative. This gives a feasible direction in which f decreases, violating the assumption that $\boldsymbol{x}^*$ is a local minimum. We can therefore conclude that Lagrange multipliers must be non-negative for inequality constraints.

The condition of Eq. (B.13) is referred to as the *complementary slackness condition* as it forces no more than one of $g_i(\boldsymbol{x}^*) < 0$ and $\lambda_i^* > 0$ to be true. This is equivalent to requiring that both $g_i(\boldsymbol{x}^*)$ and λ_i^* cannot be non-zero, or that inactive constraints have Lagrange multipliers equal to zero. If *strict complementarity* holds, then exactly one of $g_i(\boldsymbol{x}^*) < 0$ and $\lambda_i^* > 0$ is true, otherwise constraints with $g_i(\boldsymbol{x}^*) = \lambda_i^* = 0$ may exist and are termed *weakly active*. ■

B.4 The Lagrangian function

Following the result of Theorem B.6 outlining the KKT first-order necessary conditions, we define the *Lagrangian function* as

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= f(\boldsymbol{x}) + \sum_i \lambda_i g_i(\boldsymbol{x}) \\ &= f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathrm{T}} g(\boldsymbol{x}). \end{aligned} \tag{B.14}$$

Doing so gives a very helpful interpretation of KKT conditions by expressing them in terms of the first-order derivatives of L. The condition of Eq. (B.11) becomes

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) + \boldsymbol{\lambda}^{*\mathrm{T}} \nabla g(\boldsymbol{x}^*) &= 0 \\ &= \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \end{aligned} \tag{B.15}$$

and the requirement that constraints in the active set are satisfied as equalities becomes

$$\begin{aligned} g(\boldsymbol{x}^*) &= 0 \\ &= \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*). \end{aligned} \tag{B.16}$$

Under the assumption of LICQ, the Jacobian $\boldsymbol{J} = \nabla g(\boldsymbol{x}^*)$ is of full rank and the Lagrange multipliers given by $\boldsymbol{\lambda}$ are unique at the constrained optimum, so we refer to $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ as the optimal pair. Thus, the necessary conditions for the existence of an optimal KKT pair $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ are equivalent to requiring that the Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda})$ has a stationary point at $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$. This correspondence underlies the fundamental connection between constrained optimization and unconstrained minimization of a

Lagrangian function.

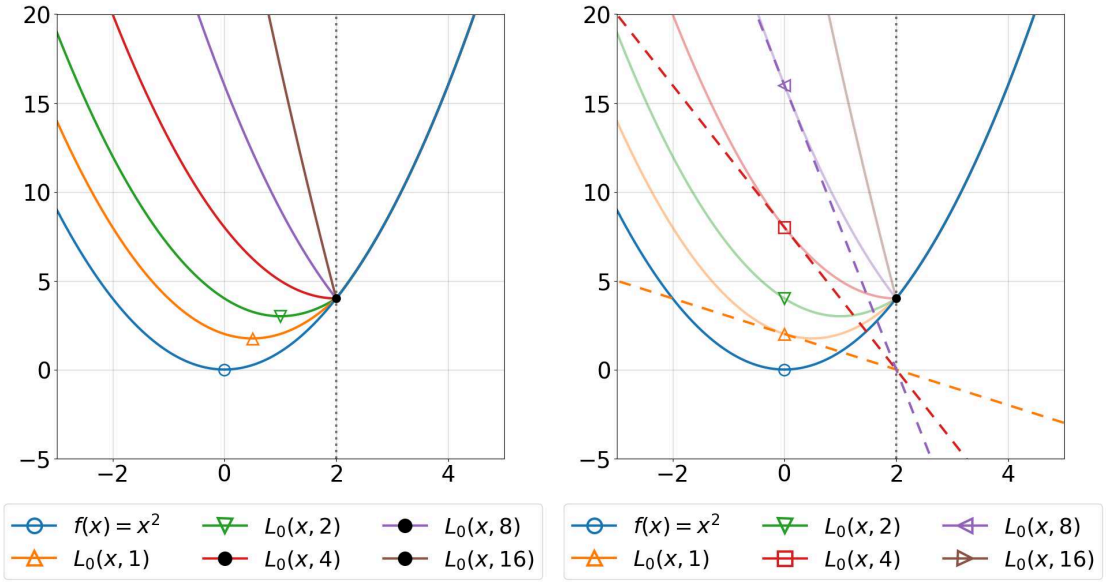

Figure B.1: Visualizations in $n = 1$ of objective function $f(x) = x^2$ with inequality constraint $x \geq 2$ and the Lagrangians $L(x, \lambda)$ resulting from $\lambda = 2^k$ for $k = 0, \ldots, 4$ after enforcing Eq. (B.13). The optimal multiplier is $\lambda^* = 4$. At left, the minimal points are marked for each curve $L_0(x, \lambda)$. At right, the intersection is marked between each curve $L(x, \lambda)$ and the line $\lambda(2 - x)$. Figure 1.1 gives the analogous case for an equality constraint.

A visual example of the correspondence between Lagrangian functions and their optimal points is given in Figure B.1, analogous to Figure 1.1, for the simple objective function $f(x) = x^2$ (blue lines) and single constraint function $g(x) = 2 - x \leq 0$. In the left-most plot, the curves resulting from

$$L(x, \lambda) = f(x) + \lambda g(x)$$

using various choices of Lagrange multiplier λ are shown along with their associated minimums. As the constraint is an inequality, the curves for the resulting Lagrangian functions are truncated to visualize the effect of Eq. (B.13) given by Theorem B.6, resulting in $L(x, \lambda) = f(x)$ whenever $g(x) \leq 0$. The optimal choice of Lagrange multiplier is $\lambda^* = 4$ for this problem, and so the curve for Lagrangian $L(x, 4)$ (red lines) shares its unconstrained minimum with the solution of the constrained problem

at $x = 2$. In the right-most plot, lines

$$\ell(x) = \lambda(2 - x)$$

are additionally shown for selected values of λ, representing the second half of the Lagrangian functions defined by Eq (B.14) and geometrically shifting the curve of the objective function so that the resulting curve shares its minimum with the solution of the constrained problem.

The Lagrangian equations of Eqs. (B.15) and (B.16) together give a system of $n + |\mathcal{A}|$ equations and unknowns, which is $n + m$ if all constraints are active. In practice, this system may even be solved analytically if the problem equations are known, giving precise Lagrange multipliers and minimum $\boldsymbol{x}^*$. Often, the multipliers must instead be approximated alongside the candidate solution. Using the Lagrangian function, it becomes possible to give a concise description of *second-order* conditions on an optimal solution $\boldsymbol{x}^*$.

Proposition B.7 (Second-order Necessary Conditions)**.** *Given a KKT pair* $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ *satisfying the conditions of Theorem B.6, if* f *and* g *are also twice continuously differentiable, then*

$$\boldsymbol{y}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \boldsymbol{y} \geq 0 \tag{B.17}$$

for all $\boldsymbol{y}$ *in the set*

$$V' = \left\{ \boldsymbol{y} : \boldsymbol{y}^T \nabla g_i(\boldsymbol{x}^*) = 0, \ \forall i \in \mathcal{A} \right\}.$$

This states that for $\boldsymbol{x}^*$ to be a constrained optimum, it is necessary for the Hessian of the Lagrangian to be positive semi-definite at $\boldsymbol{x}^*$ with respect to the set $V' \subseteq V$ containing directions $\boldsymbol{y}$ that satisfy as equalities those constraints that are in the active set. If f and the g_i are additionally convex, then these necessary conditions become sufficient conditions, and the lone optimal KKT pair corresponds to the local minimum for the problem. However, in practice it is possible that a solution satisfying both the first- and second-order necessary conditions will not also be a local minimum. In order to guarantee that our solution is also a local minimum for

L, and thus a constrained minimum for the ICP, we can use the following proposition:

Proposition B.8 (Second-order Sufficient Condition)**.** *Given a KKT pair* $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ *and with the same conditions of Proposition B.7, if additionally*

$$\boldsymbol{y}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \boldsymbol{y} > 0 \tag{B.18}$$

for all $\boldsymbol{y}$ *in* V'*, then* $\boldsymbol{x}^*$ *is a local minimizer.*

This states that if the Hessian of the Lagrangian is positive-*definite* with respect to the set of feasible variations for $\boldsymbol{x}^*$, then it is guaranteed to be a local constrained minimum. Note the similarities between this sufficient condition for a constrained optimum of ICP phrased in terms of the Lagrangian, and the second-order sufficient conditions given in Proposition B.3 for unconstrained optimization. While any point for which the Hessian of the Lagrangian is positive-definite will also be positive-definite with respect to V', the reverse need not be true: a point satisfying the second-order sufficient condition could still only be a saddle point of the Lagrangian function L. Meeting this second-order sufficient condition through unconstrained optimization of a Lagrangian is a primary motivator behind the augmented Lagrangian approach or *method of multipliers*. There are several ways to approach its construction, but the chief result is to construct a Lagrangian function that is *augmented* with a penalty term that ensures positive curvature in a neighbourhood of the optimum. In this way, a local minimum found through unconstrained minimization of the Lagrangian will correspond to the constrained minimum of the ICP, due to the second-order sufficiency condition.

B.5 Dual formulation for the augmented Lagrangian

The concept of duality comes from a more general theory for numerical optimization [28]. For the augmented Lagrangian, it involves describing a related *dual*, the solutions for which give information about (or even correspond directly with) solutions to the *primal* problem. Note that this is closely related to the *primal functional* described

in Section 3.1 but may be treated distinctly, so when referring to the latter case I explicitly use the term of primal functional. Duality results are given by each of Rockafellar [99, 100], Fletcher [42, 43], and Bertsekas [26, 27] for both convex and non-convex problems using Lagrangian functions. This section synthesizes key points of their results below. Given the definition of $L_{\boldsymbol{\omega}}$ in either of Eq. (3.9) or Eq. (3.12), the dual is here defined as

$$\begin{aligned} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) &= L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ &= \min_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha}) \end{aligned} \tag{B.19}$$

with respect to $\boldsymbol{\alpha}$, where $x(\boldsymbol{\alpha})$ is again the associated KKT point $\boldsymbol{x}$ (local optimum) for a given value of $\boldsymbol{\alpha}$, and $L_{\boldsymbol{\omega}}$ is the augmented Lagrangian which is solved by the KKT pair $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*)$. An important feature of the dual is that the value of $\boldsymbol{\alpha}^*$ maximizes the function $\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})$. This can be seen by noting first that

$$\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) = L_{\boldsymbol{\omega}}(x(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \leq L_{\boldsymbol{\omega}}(\boldsymbol{x}^*, \boldsymbol{\alpha}) \tag{B.20}$$

where the inequality follows by the definition of $\boldsymbol{x}(\boldsymbol{\alpha})$. That is, $x(\boldsymbol{\alpha})$ minimizes $L_{\boldsymbol{\omega}}$ across all $\boldsymbol{x}$ for this choice of $\boldsymbol{\alpha}$, which includes the vector $\boldsymbol{x}^*$ that minimizes $L_{\boldsymbol{\omega}}$ across all $\boldsymbol{x}$ for $\boldsymbol{\alpha}^*$. Additionally, the inequality

$$[\Psi(\boldsymbol{x})]_i \leq \frac{-\alpha_i^2}{2\omega_i} \tag{B.21}$$

can be shown to hold across all constraints. To see this, we refer to Eq. (3.9) and recall that membership in the index set $i \in \mathcal{Z}$ corresponds with constraints where

$$[\Psi(\boldsymbol{x})]_i = \alpha_i g_i(\boldsymbol{x}) + \frac{1}{2}\omega_i g_i(\boldsymbol{x})^2$$

while $i \in \mathcal{P}$ corresponds with the complementary set of constraints where

$$[\Psi(\boldsymbol{x})]_i = \frac{-\alpha_i^2}{2\omega_i}.$$

Now if $i \in \mathcal{P}$, the inequality of Eq. (B.21) is immediately satsified by the definition above, while on the other hand if $i \in \mathcal{Z}$ then we have $g_i(\boldsymbol{x}^*) \geq \frac{\alpha_i}{\omega_i}$ from the corresponding condition of Eq. (3.9), and this also satisfies Eq. (B.21). We can therefore write

$$L_{\boldsymbol{\omega}}(\boldsymbol{x}^*, \boldsymbol{\alpha}) \leq L_{\boldsymbol{\omega}}(\boldsymbol{x}^*, \boldsymbol{\alpha}^*) = \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}^*). \tag{B.22}$$

By combining the inequalities of Eqs. (B.20) and (B.22), we can therefore conclude that $\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) \leq \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}^*) \; \forall \; \boldsymbol{\alpha}$.

The same conclusion can be reached by considering derivatives of the dual function. To begin with, assume that all constraints are indexed by $\mathcal{Z}$ as given in Eq. (3.9). Then the first-order gradient is found by expanding the dual function and taking the derivative with respect to $\boldsymbol{\alpha}$ as

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) &= \nabla_{\boldsymbol{\alpha}} L_{\boldsymbol{\omega}}(x(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\
&= \nabla_{\boldsymbol{\alpha}} \left(f(\boldsymbol{x}(\boldsymbol{\alpha})) + \boldsymbol{\alpha}^{\mathrm{T}} g(\boldsymbol{x}(\boldsymbol{\alpha})) + \frac{1}{2} g(\boldsymbol{x}(\boldsymbol{\alpha}))^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}(\boldsymbol{\alpha})) \right)
\end{aligned}$$

and by applying the chain rule then collecting terms we have

$$\begin{aligned}
&= \nabla_{\boldsymbol{\alpha}} \boldsymbol{x}(\boldsymbol{\alpha}) \cdot \nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\boldsymbol{\alpha})) + g(\boldsymbol{x}(\boldsymbol{\alpha})) + \boldsymbol{\alpha}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{\alpha}} \boldsymbol{x}(\boldsymbol{\alpha}) \cdot \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\boldsymbol{\alpha})) \\
&\qquad + \nabla_{\boldsymbol{\alpha}} \boldsymbol{x}(\boldsymbol{\alpha}) \cdot \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\boldsymbol{\alpha}))^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}(\boldsymbol{\alpha})) \\
&= \nabla_{\boldsymbol{\alpha}} \boldsymbol{x}(\boldsymbol{\alpha}) \cdot \Big(\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\boldsymbol{\alpha})) + \boldsymbol{\alpha}^{\mathrm{T}} \cdot \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\boldsymbol{\alpha})) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(\boldsymbol{\alpha}))^{\mathrm{T}} \boldsymbol{\Omega} g(\boldsymbol{x}(\boldsymbol{\alpha})) \Big) \\
&\qquad + g(\boldsymbol{x}(\boldsymbol{\alpha})) \\
&= \nabla_{\boldsymbol{\alpha}} \boldsymbol{x}(\boldsymbol{\alpha}) \cdot \Big(\nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \Big) + g(\boldsymbol{x}(\boldsymbol{\alpha}))
\end{aligned} \tag{B.23}$$

where the ∇ operator applied to a vector throughout refers to the associated Jacobian matrix consisting of columns of gradients. The same result arises from treating $L_{\boldsymbol{\omega}}$ as a function of two variables and applying the "multivariable" chain rule. Let $[\partial \boldsymbol{x} / \partial \boldsymbol{\alpha}]$ refer to the matrix of partial derivatives with the entry of the i-th row and j-th column being $\partial x_i / \partial \alpha_j$, then

$$\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})^{\mathrm{T}} = \left[\frac{\partial L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial \boldsymbol{x}} \right] \left[\frac{\partial \boldsymbol{x}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \right] + \left[\frac{\partial L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \right] \left[\frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\alpha}} \right]$$

which also reduces to Eq. (B.23).

A simplification for these expressions for $\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})$ arises from using the first-order conditions of Theorem. B.6, since

$$\nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha})) = 0$$

and so Eq. (B.23) reduces to

$$\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) = g(\boldsymbol{x}(\boldsymbol{\alpha})) \tag{B.24}$$

and thus each element is given by the partial derivative

$$[\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})]_i = \frac{\partial L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial \alpha_i}$$

for constraints indexed by $i \in \mathcal{Z}$. By a similar argument, the partial derivative

$$\frac{\partial L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial \alpha_i} = \frac{-\alpha_i}{\omega_i}$$

applies to constraints indexed by $i \in \mathcal{P}$, and thus

$$\frac{\partial \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})}{\partial \alpha_i} = \max \left[g_i(\boldsymbol{x}(\boldsymbol{\alpha})), \frac{-\alpha_i}{\omega_i} \right] \tag{B.25}$$

gives the elements of the entire gradient, aligning with the conditions of Eq. (3.9).

Calculating the Hessian is slightly more involved. As we did with the gradient, assume first that the index set $\mathcal{P}$ is empty. Then begin by considering the partial derivatives of $\nabla_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\omega}}$ from Eq. (B.24) after application of the chain rule, given by

$$\frac{\partial g(\boldsymbol{x}(\boldsymbol{\alpha}))}{\partial \boldsymbol{\alpha}} = \frac{\partial g}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} = \boldsymbol{J}^{\mathrm{T}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} \tag{B.26}$$

with $\boldsymbol{J}$ the Jacobian of the constraints as in Eq. (B.2). For notational brevity, the parameters of g are dropped in the above expression, and we will continue to do so for functions where the parameters are understood. To solve for $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}}$, we can introduce known quantities by applying $\partial / \partial \boldsymbol{\alpha}$ to $\nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0$ with the multivariable

chain rule in order to get

$$\frac{\partial \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}}{\partial \boldsymbol{\alpha}} = \frac{\partial \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} + \frac{\partial \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}}{\partial \boldsymbol{\alpha}} = 0. \tag{B.27}$$

In particular, the partial derivative

$$\frac{\partial \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}}{\partial \boldsymbol{x}} = \nabla_{\boldsymbol{x}}^2 L_{\boldsymbol{\omega}} \tag{B.28}$$

is the Hessian of the augmented Lagrangian with respect to $\boldsymbol{x}$, and the partial derivative

$$\begin{aligned} \frac{\partial \nabla_{\boldsymbol{x}} L_{\boldsymbol{\omega}}}{\partial \boldsymbol{\alpha}} &= \frac{\partial}{\partial \boldsymbol{\alpha}} \left(\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) + \boldsymbol{\alpha}^{\mathrm{T}} \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) + g(\boldsymbol{x}) \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) \right) \\ &= \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) = \boldsymbol{J} \end{aligned} \tag{B.29}$$

is the Jacobian of g. Taking the values of Eqs. (B.28) and (B.29) and substituting into Eq. (B.27), we have

$$\frac{\partial \nabla_{\boldsymbol{x}} L}{\partial \boldsymbol{\alpha}} = (\nabla_{\boldsymbol{x}}^2 L_{\boldsymbol{\omega}}) \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} + \boldsymbol{J} = 0$$

which re-arranges to give

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} = -(\nabla_{\boldsymbol{x}}^2 L_{\boldsymbol{\omega}})^{-1} \cdot \boldsymbol{J}.$$

Using this identity together with Eq. (B.26), we can therefore write the Hessian of the dual as

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}}^2 \psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha}) &= \boldsymbol{J}^{\mathrm{T}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} \\ &= -\boldsymbol{J}^{\mathrm{T}} \cdot (\nabla_{\boldsymbol{x}}^2 L_{\boldsymbol{\omega}})^{-1} \cdot \boldsymbol{J} \end{aligned} \tag{B.30}$$

To also include the case where the index set $\mathcal{P}$ is not empty, observe that the various second-order partial derivatives of $\Psi(\boldsymbol{x})$ in Eq. (3.9) for $i \in \mathcal{P}$ are given by

$$\frac{\partial^2 \Psi}{\partial \alpha_i \alpha_j} = 0$$

everywhere except when $i = j$, where they are equal to $\frac{-1}{\omega_i}$. Without loss of generality, the constraint indices may be re-arranged so that the Hessian of the dual can be written using block matrices as

$$\nabla^2\psi(\boldsymbol{\alpha}) = \begin{bmatrix} -\boldsymbol{J}^{\mathrm{T}} \cdot (\nabla^2_{\boldsymbol{x}} L_{\boldsymbol{\omega}})^{-1} \cdot \boldsymbol{J} & 0 \\ 0 & -\boldsymbol{\Omega}^{-1} \end{bmatrix} \tag{B.31}$$

with $-\boldsymbol{J}^{\mathrm{T}} \cdot (\nabla^2 L_{\boldsymbol{\omega}})^{-1} \cdot \boldsymbol{J}$ corresponding to $i \in \mathcal{Z}$ and $-\boldsymbol{\Omega}^{-1}$ to $i \in \mathcal{P}$. Since the Hessian of the augmented Lagrangian is positive definite at the optimum, the Hessian of the dual is negative, and so is maximized by $\boldsymbol{\alpha}$ at the stationary point $\nabla\psi = g(\boldsymbol{x}(\boldsymbol{\alpha})) = 0$.

Using the above, and in particular Eq. (B.24), it can be seen that the update used in Eq. (3.6) for the method of multipliers is in fact a form of gradient ascent with stepsize given by $\boldsymbol{\omega}$. Other step sizes are possible, and Bertsekas [26] even derives an optimal value that is expressed in terms of minimum and maximum eigenvalues of the Hessian of the dual. The understanding of approximations for Lagrange multipliers as being maximizing steps in the search space of a dual function also suggests alternative approaches for constructing a sequence of multiplier approximations intended to converge to the optimum, such as Newton's method.

B.5.1 Newton's method for Lagrange multipliers

Newton's method for calculating Lagrange multipliers applies the same approach as described in Eq. (B.1) by minimizing the negative of the dual function $-\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})$ in order to generate a sequence $\{\boldsymbol{\alpha}^{(k)}\}$ that approaches $\boldsymbol{\alpha}^*$. Explicitly, the Newton step calculates

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} - \left(\nabla^2\psi_{\boldsymbol{\omega}}\right)^{-1} \cdot (\nabla\psi) \tag{B.32}$$

from a given estimate $\boldsymbol{\alpha}^{(k)}$ in order to minimize $-\psi_{\boldsymbol{\omega}}(\boldsymbol{\alpha})$. Combining the first- and second-order derivatives from Eqs. (B.24) and (B.30) gives the Newton step as

$$\begin{aligned}\boldsymbol{\alpha}^{(k+1)} &= \boldsymbol{\alpha}^{(k)} + \left[\boldsymbol{J}^{\mathrm{T}}\left(\nabla^2_{\boldsymbol{x}} L_{\boldsymbol{\omega}}(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)}), \boldsymbol{\alpha}^{(k)})\right)^{-1} \cdot \boldsymbol{J}\right]^{-1} \cdot g(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)})) \\ &= \boldsymbol{\alpha}^{(k)} + \left(\boldsymbol{J}^{\mathrm{T}} \cdot \nabla^2_{\boldsymbol{x}} L^{-1}_{\boldsymbol{\omega}} \cdot \boldsymbol{J}\right)^{-1} \cdot g(\boldsymbol{x}(\boldsymbol{\alpha}^{(k)}))\end{aligned} \tag{B.33}$$

The sequence arrived at for $\{\boldsymbol{\alpha}^{(k)}\}$ as generated by this approach can then be used as a sequence of Lagrange multipliers for $L_{\boldsymbol{\omega}}(\boldsymbol{x}, \boldsymbol{\alpha})$. Also of interest is to note [43] that $\left(\boldsymbol{J}^{\mathrm{T}} \cdot \nabla^2_{\boldsymbol{x}} L^{-1}_{\boldsymbol{\omega}} \cdot \boldsymbol{J}\right)^{-1} \approx \boldsymbol{\Omega}$ for large values in $\boldsymbol{\Omega}$, which reduces Eq. (B.33) to the same update rule as in Eq. (3.6).

Appendix C

Additional figures

Additional figures are collected here for the experimental results discussed in Chapter 5. Where given, regular ECDF plots are grouped together by problem and dimension, and arranged into pairs of rows representing the proportion of successful f-targets plotted against the count of f-evaluations (top) and successful g-targets plotted against the count of g-evaluations (bottom).

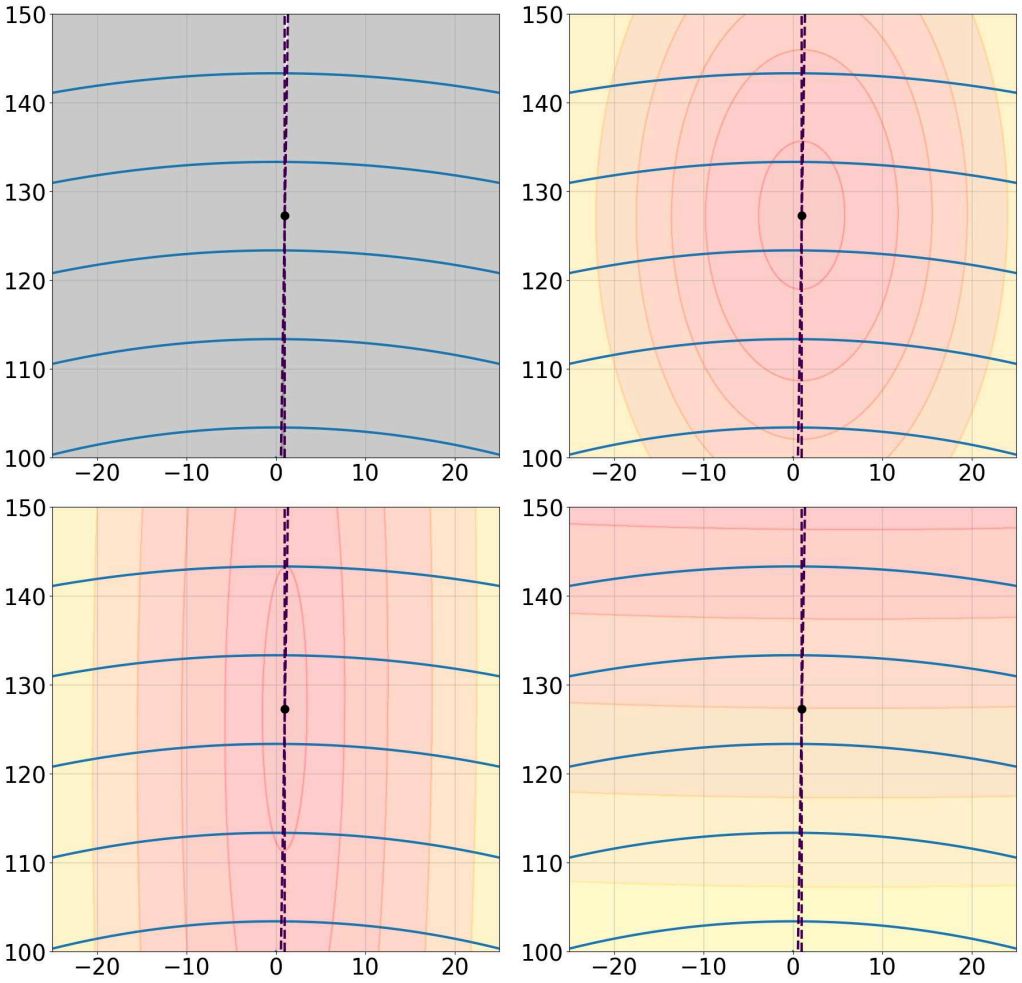

Figure C.1: Visualization in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with NFR inequality constraints. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*,\ \omega = 1$. Bottom left: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*,\ \omega = 20$. Bottom right: contour regions for $L_\omega(\boldsymbol{x}, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = 20\boldsymbol{\alpha}^*,\ \omega = 1$. The constrained optimum is marked throughout at $\boldsymbol{x}^* \approx [1, 127.321]$. Similar to Figure 5.4 but with equal axis scaling.

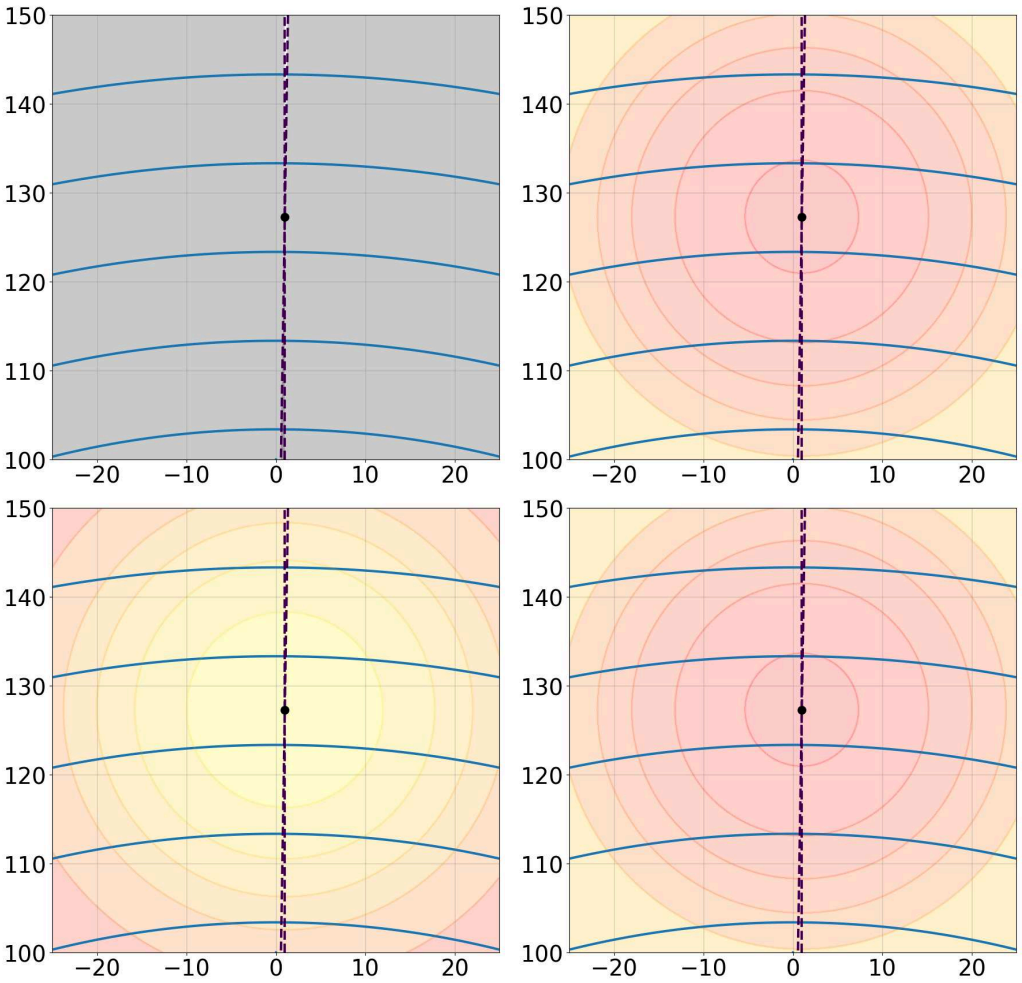

Figure C.2: Visualization in $n = 2$ of contour lines for the objective function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ with NFR inequality constraints. Top left: objective and constraint functions given with the infeasible region shaded. Top right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2$. Bottom left: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{-2}$. Bottom right: contour regions for $\phi(\boldsymbol{x})$ with $\omega = 2 \cdot 10^{2}$. The constrained optimum is marked throughout at $\boldsymbol{x}^* \approx [1, 127.321]$. Similar to Figure 5.5 but with equal axis scaling.

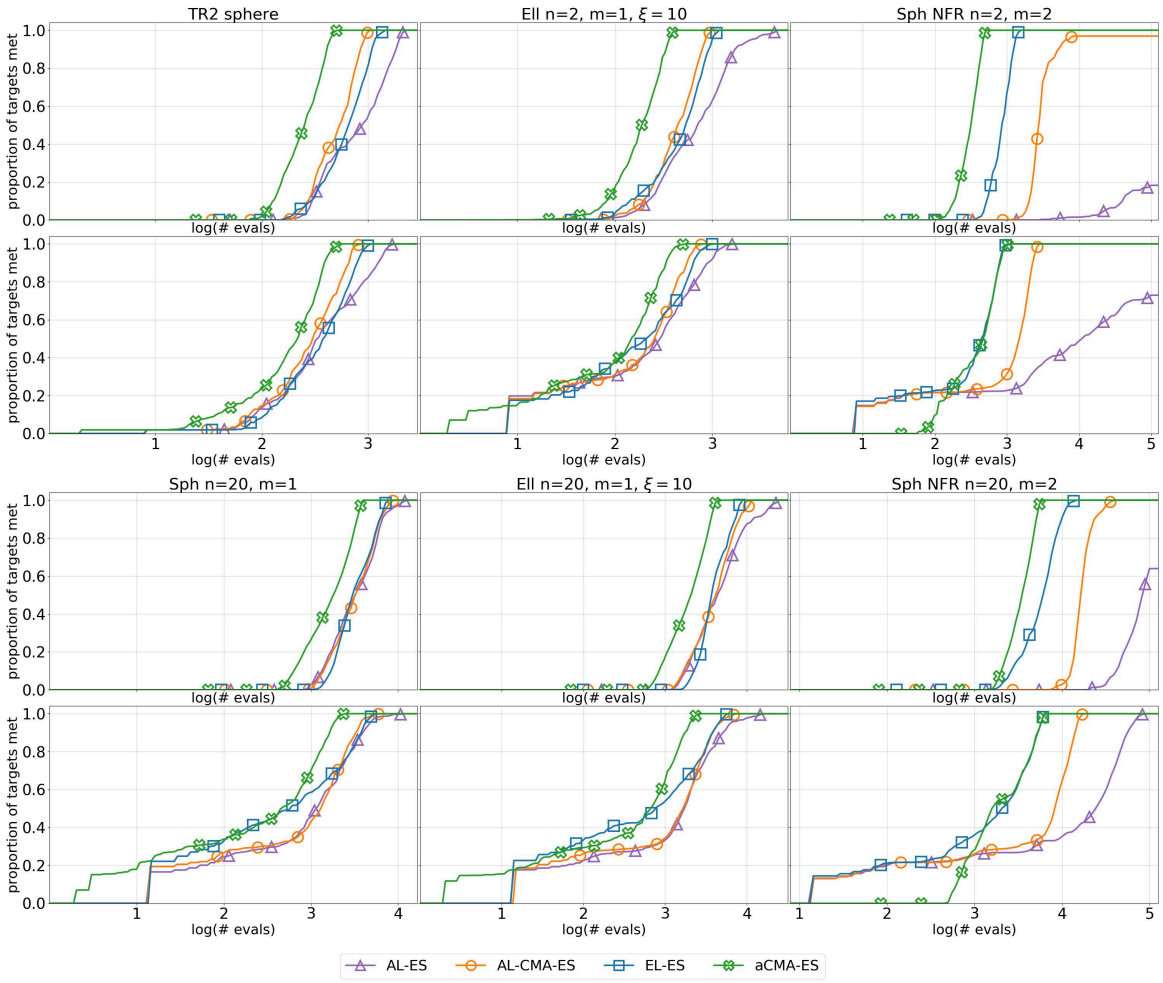

Figure C.3: ECDF plots paired vertically by problem (indicated by in-column labels) showing f-evals vs. f-targets (top plot of pair) and g-evals vs. g-targets (bottom plot of pair). The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

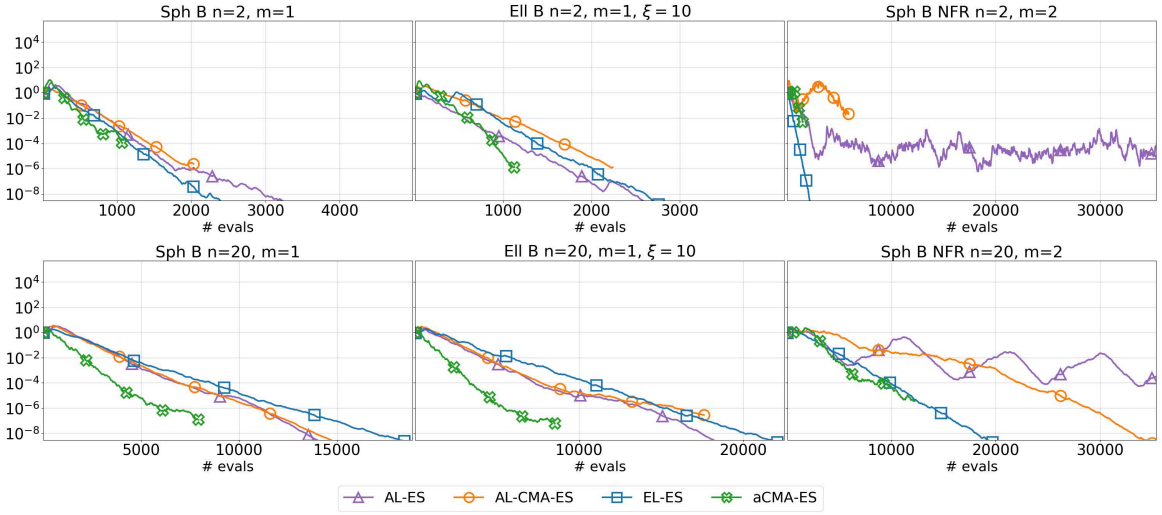

Figure C.4: Convergence plots showing step size σ with respect to $(f + g)$-evals for median runs from each of four algorithms on large B variants.

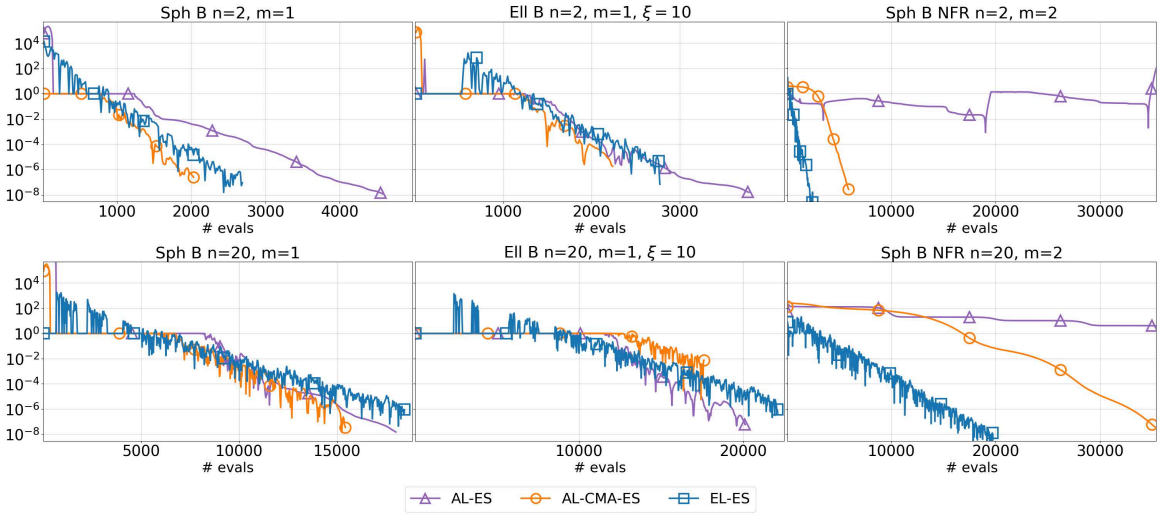

Figure C.5: Convergence plots showing distance $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$ from the optimal Lagrange multiplier vector with respect to $(f + g)$-evals for median runs from the three Lagrangian methods on large B variants.

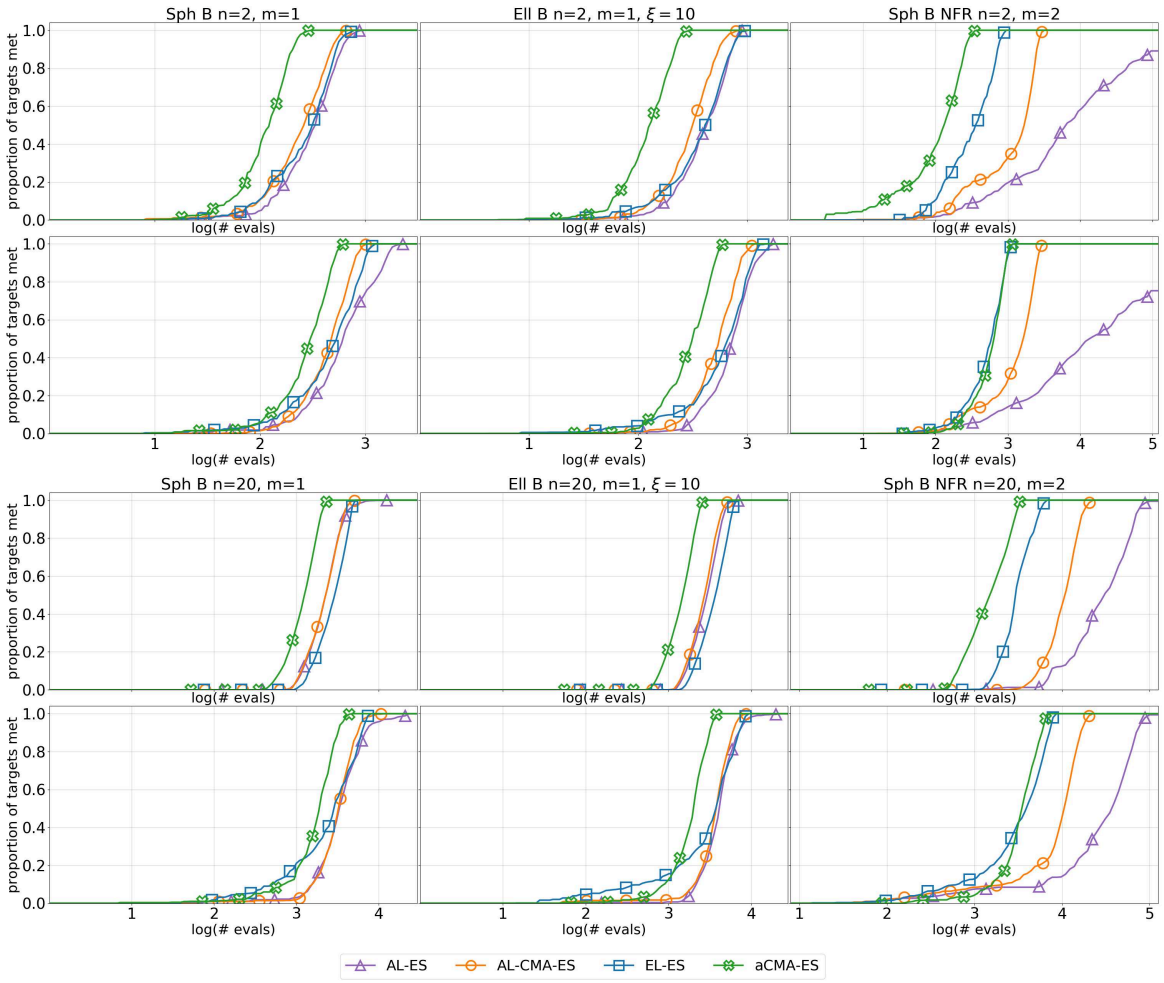

Figure C.6: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom) on large B problem variants. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

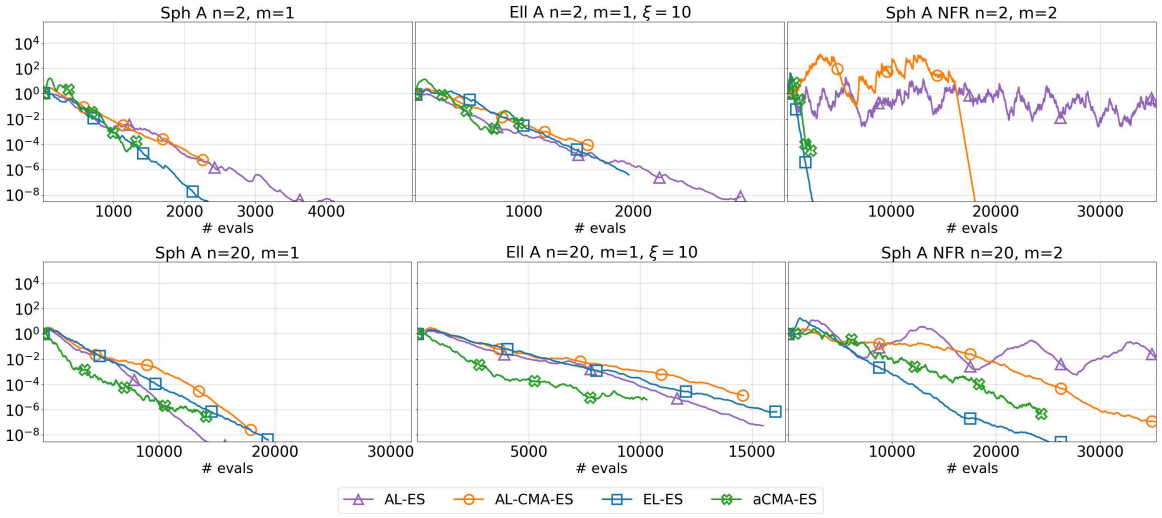

Figure C.7: Convergence plots showing step size σ with respect to $(f + g)$-evals for median runs from each of four algorithms.

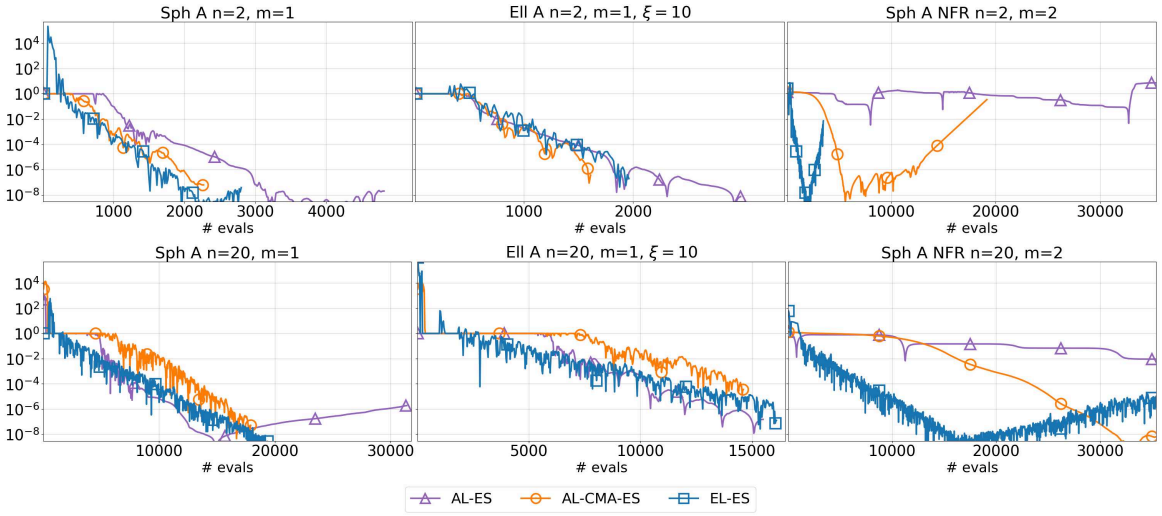

Figure C.8: Convergence plots showing distance $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|/\|\boldsymbol{\alpha}^*\|$ from the optimal Lagrange multiplier vector with respect to $(f + g)$-evals for median runs from the three Lagrangian methods. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

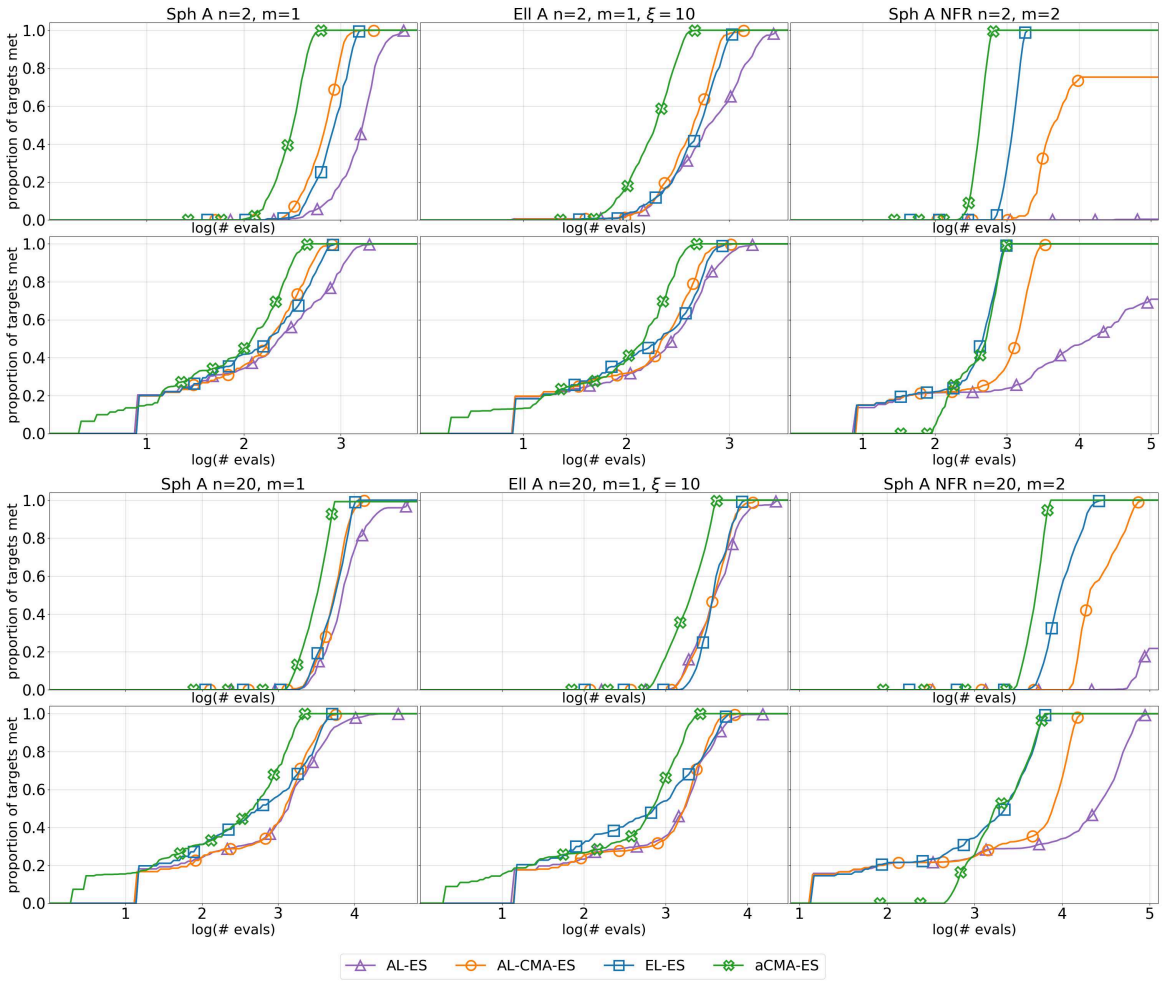

Figure C.9: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom) on large A problem variants. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

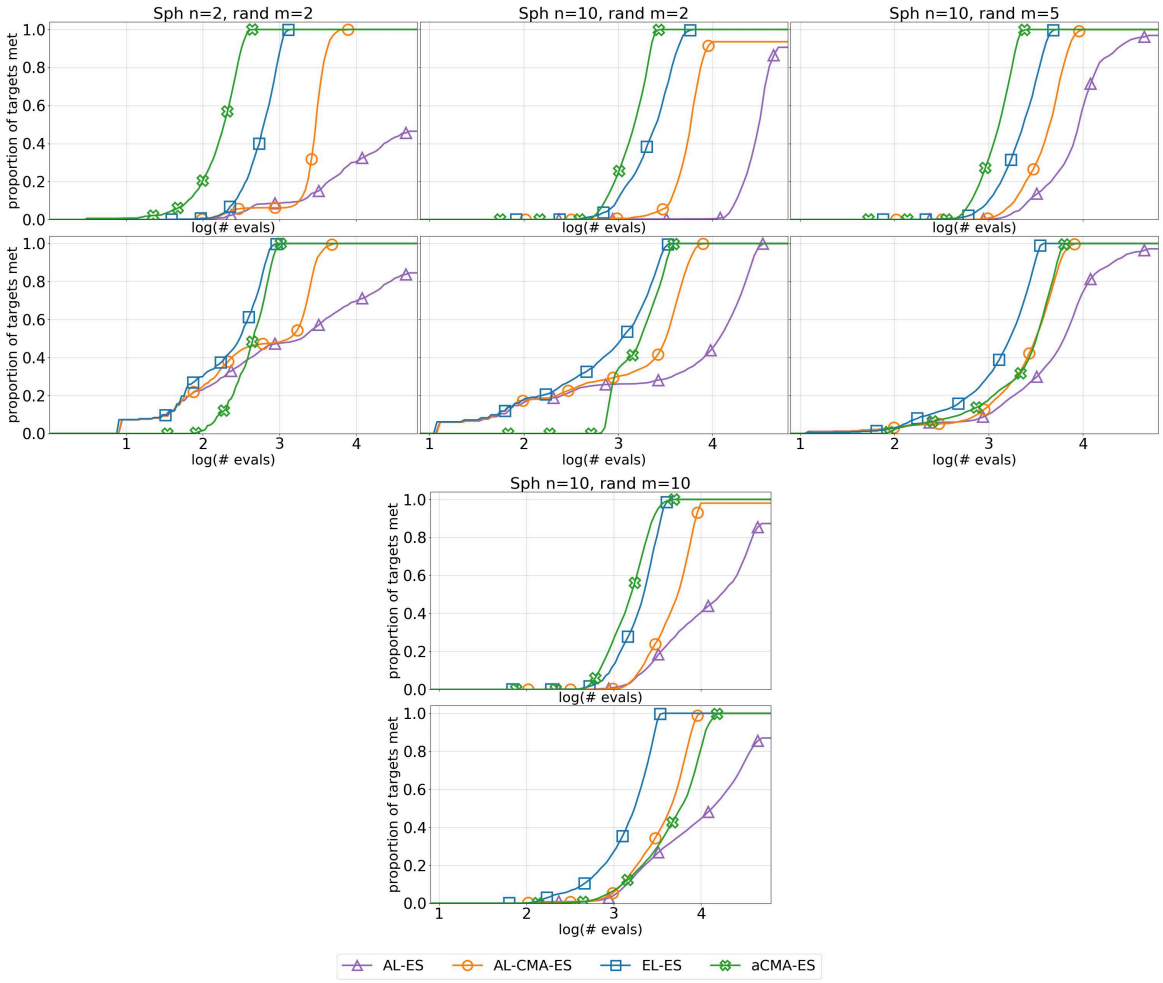

Figure C.10: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom) for randomly generated constraints on $n = 10$ spheres. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

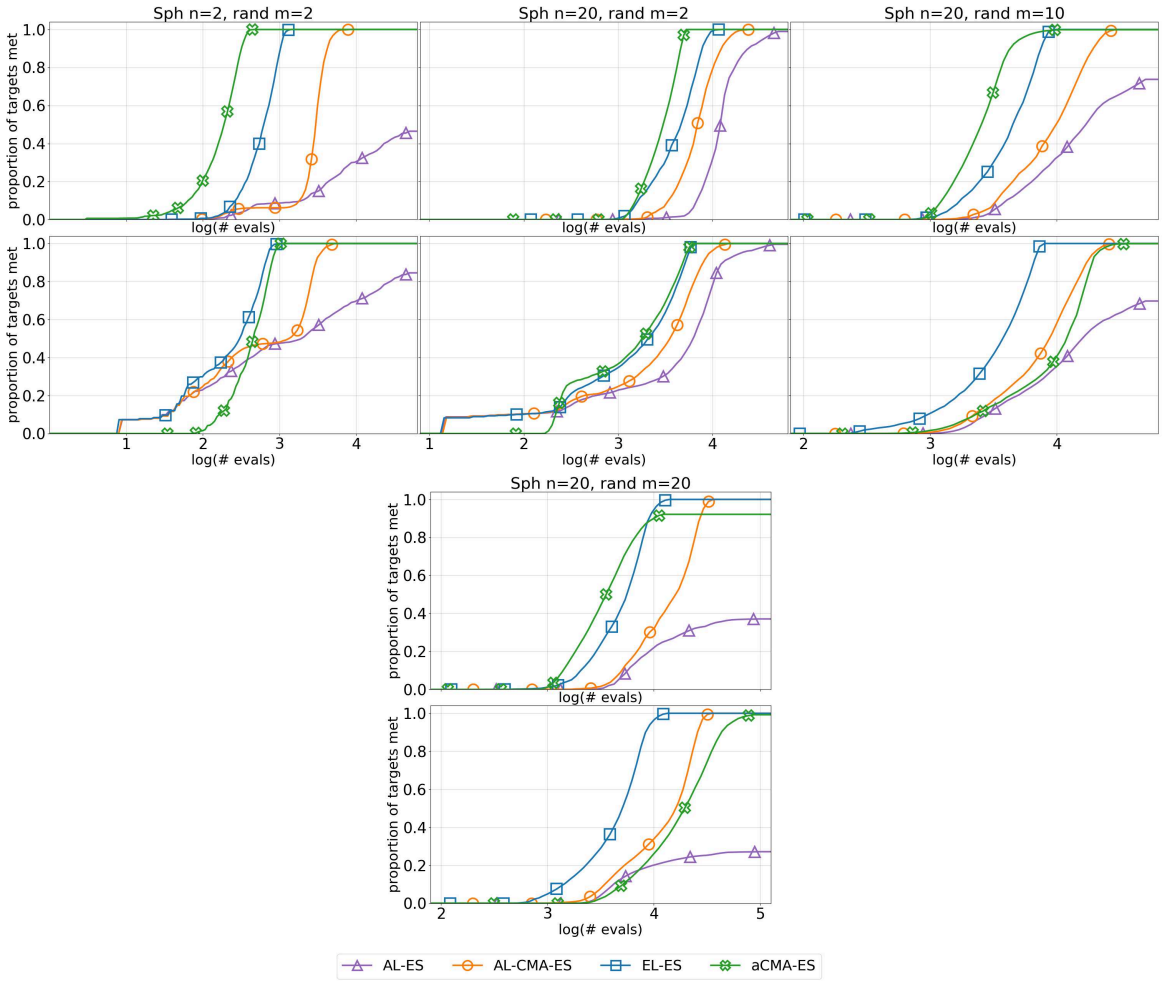

Figure C.11: Pairs of ECDF plots showing f-evals vs. f-targets (top) and g-evals vs. g-targets (bottom) for randomly generated constraints on $n = 20$ spheres. The axes are shared across plots for the same problem and aligned to allow comparisons between plots for f- and g-targets.

Bibliography

[1] A Abudhahir and S Baskar. An evolutionary optimized nonlinear function to improve the linearity of transducer characteristics. *Measurement Science and Technology*, 19(4):045103, 2008.

[2] D. V. Arnold. Analysis of a repair mechanism for the $(1, \lambda)$-ES applied to a simple constrained problem. In *Genetic and Evolutionary Computation Conference — GECCO 2011*, pages 853–860. ACM Press, 2011.

[3] D. V. Arnold. On the behaviour of the $(1, \lambda)$-ES for a simple constrained problem. In H.-G. Beyer and W. B. Langdon, editors, *Proceedings of the 11th International Conference on Foundations of Genetic Algorithms*, FOGA'11, pages 15–24. ACM Press, 2011.

[4] D. V. Arnold. Resampling versus repair in evolution strategies applied to a constrained linear problem. *Evolutionary Computation*, 21(3):389–411, 2013.

[5] D. V. Arnold and N. Hansen. Active covariance matrix adaptation for the $(1 + 1)$-CMA-ES. In *Genetic and Evolutionary Computation Conference — GECCO 2010*, pages 385–392. ACM Press, 2010.

[6] D. V. Arnold and N. Hansen. A $(1 + 1)$-CMA-ES for constrained optimisation. In *Genetic and Evolutionary Computation Conference — GECCO 2012*, pages 297–304. ACM Press, 2012.

[7] Dirk V. Arnold. Optimal weighted recombination. In *Proceedings of the 8th International Conference on Foundations of Genetic Algorithms*, FOGA'05, pages 215–237. Springer-Verlag, 2005.

[8] Dirk V Arnold. Weighted multirecombination evolution strategies. *Theoretical Computer Science*, 361(1):18–37, 2006.

[9] Dirk V Arnold. On the behaviour of the $(1, \lambda)$-σSA-ES for a constrained linear problem. In *Parallel Problem Solving from Nature-PPSN XII*, pages 82–91. Springer, 2012.

[10] Dirk V Arnold. On the behaviour of the $(1, \lambda)$-ES for conically constrained linear problems. *Evolutionary Computation*, 22(3):503–523, 2014.

[11] Dirk V Arnold. An active-set evolution strategy for optimization with known constraints. In *International Conference on Parallel Problem Solving from Nature*, pages 192–202. Springer, 2016.

[12] Dirk V Arnold. Reconsidering constraint release for active-set evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 665–672, 2017.

[13] Dirk V Arnold and Daniel Brauer. On the behaviour of the (1+1)-ES for a simple constrained problem. In *Parallel Problem Solving from Nature–PPSN X*, pages 1–10. Springer, 2008.

[14] Dirk V Arnold and Jeremy Porter. Towards an augmented Lagrangian constraint handling approach for the $(1+1)$-ES. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 249–256, 2015.

[15] Asma Atamna. *Analysis of Randomized Adaptive Algorithms for Black-Box Continuous Constrained Optimization*. PhD thesis, Université Paris-Saclay, 2017.

[16] Asma Atamna, Anne Auger, and Nikolaus Hansen. Analysis of linear convergence of a $(1+1)$-ES with augmented Lagrangian constraint handling. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 213–220, 2016.

[17] Asma Atamna, Anne Auger, and Nikolaus Hansen. Augmented Lagrangian constraint handling for CMA-ES - case of a single linear constraint. In *International Conference on Parallel Problem Solving from Nature*, pages 181–191. Springer, 2016.

[18] Asma Atamna, Anne Auger, and Nikolaus Hansen. Linearly convergent evolution strategies via augmented Lagrangian constraint handling. In *Proceedings of the 14th International Conference on Foundations of Genetic Algorithms*, FOGA'17, pages 149–161, 2017.

[19] Asma Atamna, Anne Auger, and Nikolaus Hansen. On invariance and linear convergence of evolution strategies with augmented Lagrangian constraint handling. *Theoretical Computer Science*, 832:68–97, 2020.

[20] Anne Auger, Nikolaus Hansen, JM Perez Zerpa, Raymond Ros, and Marc Schoenauer. Experimental comparisons of derivative free optimization algorithms. In *International Symposium on Experimental Algorithms*, pages 3–15. Springer, 2009.

[21] Samineh Bagheri, Wolfgang Konen, and Thomas Back. Equality constraint handling for surrogate-assisted constrained optimization. In *2016 IEEE Congress on Evolutionary Computation*, pages 1924–1931. IEEE, 2016.

[22] Samineh Bagheri, Wolfgang Konen, Michael Emmerich, and Thomas Bäck. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing*, 61:377–393, 2017.

[23] Ardeshir Bahreininejad. Improving the performance of water cycle algorithm using augmented Lagrangian method. *Advances in Engineering Software*, 132:55–64, 2019.

[24] Umesh Balande and Deepti Shrimankar. An oracle penalty and modified augmented Lagrangian methods with firefly algorithm for constrained optimization problems. *Operational Research*, 20(2):985–1010, 2020.

[25] Helio JC Barbosa, Afonso CC Lemonge, and Heder S Bernardino. A critical review of adaptive penalty techniques in evolutionary computation. In Rituparna Datta and Kalyanmoy Deb, editors, *Evolutionary Constrained Optimization*, pages 1–27. Springer, 2015.

[26] Dimitri P Bertsekas. Combined primal-dual and penalty methods for constrained minimization. *SIAM Journal on Control*, 13(3):521–544, 1975.

[27] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. New York: Academic Press, 1982.

[28] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific Belmont, 2nd edition, 1999.

[29] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[30] Johannes Daniel Buys. *Dual algorithms for constrained optimization problems*. PhD thesis, Leiden University, 1972.

[31] Rudy Chocat, Loïc Brevault, Mathieu Balesdent, and Sébastien Defoort. Modified covariance matrix adaptation–evolution strategy algorithm for constrained optimization under uncertainty, application to rocket design. *International Journal for Simulation and Multidisciplinary Design Optimization*, 6:A1, 2015.

[32] Carlos A Coello Coello. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the 23rd Annual Conference on Genetic and Evolutionary Computation*, pages 692–714, 2021.

[33] Guillaume Collange, Nathalie Delattre, Nikolaus Hansen, Isabelle Quinquis, and Marc Schoenauer. Multidisciplinary optimization in the design of future space launchers. In Piotr Breitkopf and Rajan Filomeno Coelho, editors, *Multidisciplinary Design Optimization in Computational Mechanics*, chapter 12, pages 459–468. John Wiley & Sons, Ltd, 2013.

[34] Lino Costa, Isabel Santo, Roman Denysiuk, and Edite Fernandes. Hybridization of a genetic algorithm with a pattern search augmented Lagrangian method. In *2nd International Conference on Engineering Optimization (EngOpt 2010)*, 2010.

[35] Kalyanmoy Deb and Soumil Srivastava. A genetic algorithm based augmented Lagrangian method for constrained optimization. *Computational Optimization and Applications*, 53(3):869–902, 2012.

[36] Paul Dufossé and Asma Atamna. Benchmarking several strategies to update the penalty parameters in AL-CMA-ES on the bbob-constrained testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, page 1691–1699. Association for Computing Machinery, 2022.

[37] Paul Dufossé, Cyrille Enderli, Laurent Savy, and Nikolaus Hansen. Phased-array antenna pattern optimization with evolution strategies. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6. IEEE, 2020.

[38] Paul Dufossé and Nikolaus Hansen. Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 519–527, 2021.

[39] Roger Fletcher. A class of methods for nonlinear programming with termination and convergence properties. In J. Abadie, editor, *Integer and Nonlinear Programming*, pages 157–175. North-Holland, 1970.

[40] Roger Fletcher. A general quadratic programming algorithm. *IMA Journal of Applied Mathematics*, 7(1):76–91, 1971.

[41] Roger Fletcher. An exact penalty function for nonlinear programming with inequalities. *Mathematical Programming*, 5(1):129–150, 1973.

[42] Roger Fletcher. An ideal penalty function for constrained optimization. *Journal of the Institute of Mathematics and its Applications*, 15(3):319–342, 1975.

[43] Roger Fletcher. *Practical Methods of Optimization*, volume 2. John Wiley & Sons, 1981.

[44] Roger Fletcher. *Practical Methods of Optimization*, volume 1. John Wiley & Sons, 1981.

[45] Roger Fletcher and Shirley A Lill. A class of methods for nonlinear programming: II computational experience. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 67–92. Academic Press, 1970.

[46] Christodoulos A Floudas and Panos M Pardalos. *A collection of test problems for constrained global optimization algorithms*. Springer Verlag, 1990.

[47] Xihe Gao, Jeremy Porter, Stephen Brooks, and Dirk V Arnold. Evolutionary optimization of tone mapped image quality index. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 176–188. Springer, 2017.

[48] Torkel Glad and Elijah Polak. A multiplier method with automatic limitation of penalty growth. *Mathematical Programming*, 17(1):140–155, 1979.

[49] PC Haarhoff and JD Buys. A new method for the optimization of a nonlinear function subject to nonlinear constraints. *The Computer Journal*, 13(2):178–184, 1970.

[50] S Ben Hamida and Marc Schoenauer. An adaptive algorithm for constrained optimization problems. In *Parallel Problem Solving from Nature PPSN VI*, pages 529–538. Springer, 2000.

[51] Nikolaus Hansen. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[52] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *CoRR*, abs/1604.00772, 2016.

[53] Nikolaus Hansen, Dirk V Arnold, and Anne Auger. Evolution strategies. In Janusz Kacprzyk and Witold Pedrycz, editors, *Handbook of Computational Intelligence*. Springer, 2015.

[54] Nikolaus Hansen and Anne Auger. Principled design of continuous stochastic search: From theory to practice. In Yossi Borenstein and Alberto Moraglio, editors, *Theory and principled methods for the design of metaheuristics*, pages 145–180. Springer, 2014.

[55] Nikolaus Hansen, Anne Auger, Dimo Brockhoff, Dejan Tušar, and Tea Tušar. COCO: Performance Assessment. *arXiv e-prints*, page arXiv:1605.03560, May 2016.

[56] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference companion on Genetic and Evolutionary Computation*, pages 1689–1696, 2010.

[57] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021.

[58] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[59] Michael Hellwig and Hans-Georg Beyer. A linear constrained optimization benchmark for probabilistic search algorithms: the Rotated Klee-Minty problem. In *International Conference on Theory and Practice of Natural Computing*, pages 139–151. Springer, 2018.

[60] Michael Hellwig and Hans-Georg Beyer. A matrix adaptation evolution strategy for constrained real-parameter optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2018.

[61] Michael Hellwig and Hans-Georg Beyer. Analysis of a meta-ES on a conically constrained problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 673–681, 2019.

[62] Michael Hellwig and Hans-Georg Beyer. Benchmarking evolutionary algorithms for single objective real-valued constrained optimization–a critical review. *Swarm and Evolutionary Computation*, 44:927–944, 2019.

[63] Michael Hellwig, Patrick Spettel, and Hans-Georg Beyer. Comparison of contemporary evolutionary algorithms on the Rotated Klee-Minty problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1879–1887, 2019.

[64] Magnus R Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[65] David Mautner Himmelblau. *Applied Nonlinear Programming*. New York: McGraw-Hill, 1972.

[66] Willi Hock and Klaus Schittkowski. *Test examples for nonlinear programming codes*. Springer Verlag, 1981.

[67] Christian Igel, Thorsten Suttorp, and Nikolaus Hansen. A computational efficient covariance matrix update and a (1+ 1)-CMA for evolution strategies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 453–460. ACM, 2006.

[68] G. A. Jastrebski and D. V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *IEEE World Congress on Computational Intelligence — WCCI 2006*, pages 9719–9726. IEEE Press, 2006.

[69] Jeffrey A Joines and Christopher R Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In *International Conference on Evolutionary Computation*, pages 579–584. IEEE, 1994.

[70] Stefan Kern, Sibylle D Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing*, 3(1):77–112, 2004.

[71] Minty GJ Klee V. How good is the simplex algorithm? In *Inequalities - III : proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, September 1-9, 1969*, pages 159–175, 1972.

[72] O. Kramer and H.-P. Schwefel. On three new approaches to handle constraints within evolution strategies. *Natural Computing*, 5(4):363–385, 2006.

[73] Oliver Kramer. A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*, 2010.

[74] Oswin Krause and Tobias Glasmachers. A CMA-ES with multiplicative covariance matrix updates. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 281–288, 2015.

[75] Sébastien Le Digabel and Stefan M Wild. A taxonomy of constraints in simulation-based optimization. *arXiv preprint arXiv:1505.07881*, 2015.

[76] Genghui Li and Qingfu Zhang. Multiple penalties and multiple local surrogates for expensive constrained optimization. *IEEE Transactions on Evolutionary Computation*, 25(4):769–778, 2021.

[77] JJ Liang, Thomas Philip Runarsson, Efren Mezura-Montes, Maurice Clerc, PN Suganthan, CA Coello Coello, and Kalyanmoy Deb. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, 2006.

[78] Wen Long, Ximing Liang, Shaohong Cai, Jianjun Jiao, and Wenzhuan Zhang. A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. *Neural Computing and Applications*, 28(1):421–438, 2017.

[79] Wen Long, Ximing Liang, Shaohong Cai, Jianjun Jiao, and Wenzhuan Zhang. An improved artificial bee colony with modified augmented Lagrangian for constrained optimization. *Soft Computing*, 22(14):4789–4810, 2018.

[80] Wen Long, Ximing Liang, Yafei Huang, and Yixiong Chen. A hybrid differential evolution augmented Lagrangian method for constrained numerical and engineering optimization. *Computer-Aided Design*, 45(12):1562–1574, 2013.

[81] Asghar Mahdavi and Mohammad Ebrahim Shiri. An augmented Lagrangian ant colony based method for constrained optimization. *Computational Optimization and Applications*, 60(1):263–276, 2015.

[82] R. Mallipeddi and P. N. Suganthan. Problem definitions and evaluation criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization. Technical report, Nanyang Technological University, Singapore, 2010.

[83] Silja Meyer-Nieberg and Hans-Georg Beyer. The dynamical systems approach – progress measures and convergence properties. In *Handbook of Natural Computing*, pages 741–814. Springer, 2012.

[84] E. Mezura-Montes and C. A. Coello Coello. Constraint-handling in nature-inspired numerical optimization: Past, present, and future. *Swarm and Evolutionary Computation*, 1(4):173–194, 2011.

[85] Efrén Mezura-Montes and Carlos A Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, 2005.

[86] Zbigniew Michalewicz and Naguib Attia. Evolutionary optimization of constrained problems. In *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 98–108. World Scientific Publishing, 1994.

[87] A Miele, EE Cragg, and AV Levy. Use of the augmented penalty function in mathematical programming problems, part 2. *Journal of Optimization Theory and Applications*, 8(2):131–153, 1971.

[88] Jorge J Moré and Stefan M Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[89] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaptation based on non-local use of selection information. In Y. Davidor et al., editors, *Parallel Problem Solving from Nature — PPSN III*, pages 189–198. Springer Verlag, 1994.

[90] Radka Polakova. L-SHADE with competing strategies applied to constrained optimization. In *2017 IEEE Congress on Evolutionary Computation*, pages 1683–1689. IEEE, 2017.

[91] Jeremy Porter and Dirk V Arnold. An evolutionary spline fitting algorithm for identifying filamentous cyanobacteria. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 40–45, 2013.

[92] Jeremy Porter and Dirk V Arnold. Analyzing the behaviour of multi-recombinative evolution strategies applied to a conically constrained problem. In Rituparna Datta and Kalyanmoy Deb, editors, *Evolutionary Constrained Optimization*, pages 181–204. Springer, 2015.

[93] Michael JD Powell. A method for nonlinear constraints in minimization problems. In *Optimization*, pages 283–298. Academic Press, 1969.

[94] I. Rechenberg. *Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Friedrich Frommann Verlag, 1973.

[95] Rommel G Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243, 2014.

[96] Rommel G Regis. Surrogate-assisted particle swarm with local search for expensive constrained optimization. In *International Conference on Bioinspired Methods and Their Applications*, pages 246–257. Springer, 2018.

[97] Ana Maria AC Rocha, Tiago FMC Martins, and Edite MGP Fernandes. An augmented Lagrangian fish swarm based method for global optimization. *Journal of Computational and Applied Mathematics*, 235(16):4611–4620, 2011.

[98] R Tyrrell Rockafellar. New applications of duality in nonlinear programming. In *Proceedings of the Fourth Conference on Probability Theory (Braşov, 1971)*, pages 73–81, 1970.

[99] R Tyrrell Rockafellar. A dual approach to solving nonlinear programming problems by unconstrained optimization. *Mathematical Programming*, 5(1):354–373, 1973.

[100] R Tyrrell Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12(2):268–285, 1974.

[101] R.T. Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, 12(6):555–562, 1973.

[102] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *Computer Journal*, 3(3):175–184, 1960.

[103] Günter Rudolph. Stochastic convergence. In *Handbook of Natural Computing*, pages 847–869. Springer, 2012.

[104] Thomas P. Runarsson and Xin Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, 2000.

[105] Naoki Sakamoto and Youhei Akimoto. Adaptive ranking based constraint handling for explicitly constrained black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 700–708, 2019.

[106] Naoki Sakamoto and Youhei Akimoto. Adaptive Ranking-based Constraint Handling for Explicitly Constrained Black-Box Optimization. *Evolutionary Computation*, pages 1–32, 04 2022.

[107] H-P Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser, Basel, 1977.

[108] H.-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, 1981.

[109] H. P. Schwefel. *Evolution and Optimum Seeking*. Wiley, 1995.

[110] Ofer M Shir and Amir Yehudayoff. On the covariance-hessian relation in evolution strategies. *Theoretical Computer Science*, 801:157–174, 2020.

[111] Aijuan Song, Guohua Wu, R. Mallipeddi, and P.N. Suganthan. Comparison of results in 2019 on CEC 2017 competition on constrained real parameter optimization. `https://github.com/P-N-Suganthan/CEC2017`. last accessed on 2022-04-25.

[112] Patrick Spettel, Zehao Ba, and Dirk V. Arnold. Active Sets for Explicitly Constrained Evolutionary Optimization. *Evolutionary Computation*, pages 1–23, 04 2022.

[113] Patrick Spettel and Hans-Georg Beyer. Analysis of the $(\mu/\mu_{\mathrm{i}},\ \lambda)$-$\sigma$-self-adaptation evolution strategy with repair by projection applied to a conically constrained problem. *CoRR*, abs/1812.06300, 2018.

[114] Patrick Spettel and Hans-Georg Beyer. Analysis of the $(1,\ \lambda)$-σ-self-adaptation evolution strategy with repair by projection applied to a conically constrained problem. *Theoretical Computer Science*, 785:30–45, 2019.

[115] Patrick Spettel and Hans-Georg Beyer. A multi-recombinative active matrix adaptation evolution strategy for constrained optimization. *Soft Computing*, 23(16):6847–6869, 2019.

[116] Patrick Spettel and Hans-Georg Beyer. Analysis of the $(\mu/\mu,\ \lambda)$-CSA-ES with repair by projection applied to a conically constrained problem. *Evolutionary Computation*, 28(3):463–488, 2020.

[117] Patrick Spettel, Hans-Georg Beyer, and Michael Hellwig. A covariance matrix self-adaptation evolution strategy for optimization under linear constraints. *IEEE Transactions on Evolutionary Computation*, 23(3):514–524, 2018.

[118] Patrick Spettel, Hans-Georg Beyer, and Michael Hellwig. Steady state analysis of a multi-recombinative meta-ES on a conically constrained problem with comparison to σSA and CSA. In *Proceedings of the 15th International Conference on Foundations of Genetic Algorithms*, FOGA'19, pages 43–57, 2019.

[119] Soumil Srivastava and Kalyanmoy Deb. A genetic algorithm based augmented Lagrangian method for computationally fast constrained optimization. In *Swarm, Evolutionary, and Memetic Computing*, pages 330–337. Springer, 2010.

[120] Thorsten Suttorp, Nikolaus Hansen, and Christian Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75(2):167–197, 2009.

[121] Min-Jea Tahk and Byung-Chan Sun. Coevolutionary augmented Lagrangian methods for constrained optimization. *IEEE Transactions on Evolutionary Computation*, 4(2):114–124, 2000.

[122] T. Takahama and S. Sakai. Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites. In *IEEE World Congress on Computational Intelligence – WCCI 2006*, pages 308–315. IEEE Press, 2006.

[123] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In *IEEE Congress on Evolutionary Computation*, pages 1–9. IEEE, 2010.

[124] Biruk Tessema and Gary G Yen. An adaptive penalty formulation for constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(3):565–578, 2009.

[125] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. A comparative study of large-scale variants of CMA-ES. In *International conference on parallel problem solving from nature*, pages 3–15. Springer, 2018.

[126] Yong Wang, Jia-Peng Li, Xihui Xue, and Bing-chuan Wang. Utilizing the correlation between constraints and objective function for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 24(1):29–43, 2020.

[127] Yong Wang, Da-Qing Yin, Shengxiang Yang, and Guangyong Sun. Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE Transactions on Cybernetics*, 49(5):1642–1656, 2019.

[128] Stephen J Wright and Jorge Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[129] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2017.

[130] Tao Xu, Jun He, and Changjing Shang. Helper and equivalent objective different evolution for constrained optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 9–10, 2019.

[131] Tao Xu, Jun He, and Changjing Shang. Helper and equivalent objectives: Efficient approach for constrained optimization. *IEEE Transactions on Cybernetics*, 52(1):240–251, 2020.

[132] Min Zhang, Wenjian Luo, and Xufa Wang. Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 178(15):3043–3074, 2008.