MULTI-COMMODITY NETWORK FLOW BASED APPROACH TO THE DYNAMIC CELL FORMATION PROBLEM: AN EXPLORATION OF DESIGN ROBUSTNESS

by

Salah Mehdi Elaskari

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

at

Dalhousie University
Halifax, Nova Scotia
July 2014

DEDICATION

This work is dedicated to my wife and children.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis deals with robust cell formation and the design of central backup cellular manufacturing systems under uncertainty. Robustness in design is important when designing cellular manufacturing systems because these systems have to perform efficiently over long periods of time and under uncertainty in many, if not most, design parameters. This research has three main objectives: (a) first, the design of cellular manufacturing systems in a dynamic environment (multi-period cell formulation problem or the MPCFP); (b) second, the introduction and analysis of a new cellular manufacturing system called the central backup cellular manufacturing system (CBCMS); and (c) third, the formulation of the cell design problem as a robust optimization problem.

We first develop an efficient mathematical model and solution strategy for the MPCFP that arises when designing dynamic cellular manufacturing systems. We illustrate the formulation through the use of several examples from the literature and provide a computational benchmark for future research. As part of the second objective, we propose the central backup cellular manufacturing system, which may be viewed as a generalized system that combines group technology (GT) design and fractal cell design.

The CBCMS is presented as the design continuum between GT and fractal cell layout organizations for manufacturing systems. The MPCFP model is used as a springboard to explore the CBCMS with the introduction of special constraints, to optimally design GT, fractal, and CBCMS manufacturing systems. Several scenarios are presented to understand the design continuum with the use of an example from the literature.

Finally, we look at a robust extension to the MPCFP model and discuss how cell formation decisions depend on the nature of uncertainty and the objectives of the cell designer. In the robust extension, product demands in each period is expressed through a set of finite scenarios with a given probability of occurrence. Error variables are used in the formulation to represent shortfall in product demand and shortage in machine capacity. Using an example from the literature, we show how the optimal solution to the robust extension to the MPCFP depends on the penalties that a decision maker assigns to these error variables.

LIST OF ABBREVIATIONS USED

BSH	Boundary Search Heuristic
CBCMS	Central Backup Cellular Manufacturing System
CFP	Cell Formation Problem
CMS	Cellular Manufacturing System
DCFP	Dynamic Cell Formation Problem
DFLP	Dynamic Facility Layout Problem
DLP	Dynamic Layout Problem
DPLP	Dynamic Plant Layout Problem
FLP	Facility or Block Layout Problem
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
GNP	Gross National Product
GT	Group Technology
LP	Linear Programming
MCNFM	Multi-Commodity Network Flow Model
MIP	Mixed Integer Programming
MPA	Manufacturing System Performance Analyzer
MPCFP	Multi-Period Cell Formation Problem
PF/CF	Part Family/Cell Formation
QAP	Quadratic Assignment Problem
RMPCFP	Robust Multi-Period Cell Formation Problem
RO	Robust Optimization
SA	Simulated Annealing
SME	Small to Mid-sized Enterprise
WIP	Work in Process

ACKNOWLEDGEMENTS

First of all, I wish to express my sincere gratitude and thanks to my supervisor, Dr. Uday Venkatadri, for his invaluable advice and support that helped me complete my dissertation. Without his knowledge and experience, it would have been much more difficult to finish the research work. I truly appreciate his time, effort, and encouragement. More importantly, I want to thank him for his sincere support and guidance during the long journey of my program. Under his supervision, I become a better researcher and a better instructor.

I would also like to acknowledge the other committee members (in no particular order): Dr. Carl-Louis Sandblom, Dr. J. Pemberton Cyrus, and Dr. Lei Liu for their time, useful comments and helpful advice in improving the quality of work during the course of my program. As well, I would like to thank Dr. Eldon Gunn for his advice and helpful suggestions. My sincere thanks go to the faculty, staff and many graduate students in the Department of Industrial Engineering. Their support and friendship made life more enjoyable.

I am grateful to have been supported by the postgraduate scholarship from the Libyan Ministry of Higher Education and Scientific Research administrated by the Libyan-North American Scholarship Program through the Canadian Bureau for International Education (CBIE). Also, I would like to express my thanks and appreciation to the Natural Sciences and Engineering Research Council of Canada (NSERC) and DSTN: DSME Trenton Ltd. for their funding through the Collaborative Research and Development grants program. Thanks also go to Dalhousie University and its financial and moral support.

Finally, considerable love and appreciation go out to my wife, Fatma Edyoub, who was with me all these years. I thank her for being an understanding and wonderful person. I thank my children for their sacrifice and patience. I am also indebted to my parents, brothers, and sisters who provided encouragement and support throughout my studies.

Chapter 1: Introduction

1.1 Facility Layout Design

The design components of a facility include the facility system, its layout, and its material handling system (Tompkins, 2003). While these components are inter-related, they are generally treated as separate entities. The facility system includes infrastructure components such as buildings, parking lots, and road access. The goal of facility systems design is to ensure that the energy, light, gas, heat, ventilation, air conditioning, water, sewage, communication, and safety needs of the infrastructure are met. The layout consists of production areas, production-support areas, and personnel areas within the facility. The production area may consist of equipment, machinery and tools. The production support area may include inspection stations, quality labs, maintenance shops, and storages. The personnel area may include operator work space, offices, lockers, and lunch-rooms. The handling system may incorporate material handling equipment and mechanisms needed to support production and storage facility interactions. Since all of these entities are closely related to each other, the design components should be investigated and analyzed both independently and together within the context of facility design during the process of designing or maintaining a facility.

In today's business environment, facilities planning is a strategic activity. Manufacturing companies collaborate with partners and align their activities with suppliers and customers to form a supply chain in an effort to remain competitive in the global marketplace. It is important to recognize that contemporary facilities planning considers the facility as a dynamic entity and that the key requirement for a successful facilities plan is its adaptability and its ability to become suitable for new use (Tompkins, 2003). Today's customers are more aware of what products are available and what quality and price is suitable for them. On the other hand, trade barriers are very limited nowadays, resulting in many products and components being sourced overseas. Due to these factors, most manufacturing companies must now be flexible, responsive, efficient and agile to fulfill market requirements. Responding to abnormalities quickly and changing plans to meet delivery dates within a competitive budget envelop is therefore a major challenge as well as a goal.

While most of the literature pertaining to facilities layout planning is geared towards industrial and manufacturing plants, these techniques and principles can be used to solve the layout problems for hospitals, hotels, educational facilities, airports, transportation hubs and commercial centers, as well as public utilities such as banks, post offices and similar types of buildings. (Tompkins, 2003) indicates that, since 1955, approximately 8% of the gross national product (GNP) of the United States has been spent annually on new facilities. Table 1-1 shows a breakdown of the GNP percentage for the most important industry groupings in the United States.

Table 1-1: GNP annual expenditures on new facilities in the U.S. since 1955

Industry	GNP (%)
Manufacturing	3.2
Mining	0.2
Railroad	0.2
Air and other transportation	0.3
Public utilities	1.6
Communication	1.0
Commercial and others	1.5
All industries	**8.0**

1.2 Factors in Facility Layout Design

There has been a rapid increase in local and international competition in almost all industries. Moreover, because facility design is a long-term infrastructure planning decision, a number of factors related to facility layout design have to be considered:

- Product life cycles are becoming shorter even as marketplace demand is driving product life cycles. Some companies initiate products or modify current products in order to take advantage of temporary and long-term market changes.

- Product mix variability, rapid changeover, variety in product design, and adjusting the product mix level to meet the requirements of the market should be prioritized to satisfy customers.
- Due to demand volume variability, customer orders may vary significantly from period to period.
- Increasing demand for customization and shorter delivery time results in manufacturers being forced to be responsive in meeting customer requests.
- Due to rapid technological advances, manufacturers must continuously modify and/or introduce new equipment or processes to revamp their products as a way of taking advantage of the latest technology.

Other factors may come into play due to organizational needs, ecological impacts, and legal requirements. Organizational needs may include addressing company growth or even the need to downsize or outsource because of fierce competition. From the point of view of ecological impact, many companies today are undergoing a greening attempt to decrease energy and fuel consumption, and reduce their carbon footprint. Legal requirements may imply new or modified products as a result of laws passed during a legislative season.

The solutions that are most in demand for solving facility layout problems in today's business environment are generally those involving comprehensive analysis and simplification. Using sophisticated technology and applying advanced methodology that at some point improves productivity tends to increase complexity and often makes matters difficult to control. It is worth noting that some of the most successful manufacturers not only use highly sophisticated technology and the most advanced techniques, but they also combine them with excellent management, organization development, and system analysis and optimization of manufacturing systems. Most research work on facility design layout has been designed based on dominant product flow that focuses on reducing the cost of material handling rather than decreasing the overall cost of manufacturing system activities. Studies on manufacturing indicate that 30-75% of a product's cost can be attributed to material handling expenses (Sule, 2009). (Tompkins, 2003) mentions that material handling activities account for 20-50% of a manufacturing company's total operating budget. Therefore, while minimizing material handling is important, the overall cost should also be considered.

1.3 Addressing the Re-Layout Problem in Changing Environments

As a result of product life cycles, product mix, demand variability, level of customization, level of automation, and other mentioned factors, a layout that has been designed based on dominant product flow may be subject to change over time. However, regardless of how long a layout has been in use, incremental improvement and re-layout will take place in time in order to minimize production costs and remain competitive.

The redesign of an existing facility can be the result of any number of circumstances, such as the introduction of completely different products from different businesses, the introduction of new products in the same line of business, or the installation of new equipment or processes. It can also be the result of realizing an increase or decrease in throughput volume due to the evolution of the product life cycle, resulting in product mix variability. This may lead to modifications in material handling equipment and mechanisms needed to support production and facility interaction. Therefore, changes to material handling equipment and processes, re-layout of existing equipment, and re-arrangement of production support areas and personnel areas are options that manufacturing facilities may need to consider over time.

Re-layout may include minor redesign with the need for department or cell integration, expansion, dissolution, or major redesign (i.e., creating a completely new facility). (Heragu, 2008) stated that designers will encounter layout design problems when a system needs to be expanded, consolidated, or modified in another way. Based on these assumptions, we introduce a classification for the re-layout design stages. Figure 1-1 classifies the re-layout stages into four main categories:

- Modify the existing facility system. This may include re-layout of some machines in the facility and/or modifications in material handling systems.
- Consolidate components of the existing facility system.
- Expand or downsize the existing facility system.

- Design a completely new facility.

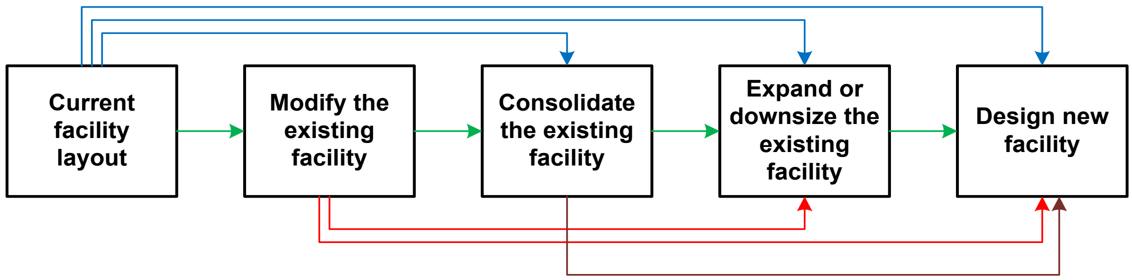


Figure 1-1: Re-layout stages

Once a layout decision is made, it cannot be changed in the short-term. The layout remains in place for a considerable period of time and it is therefore crucial that options be evaluated as carefully as possible before making a decision about alterations.

Over the past two decades, researchers and manufacturing companies have explored the dynamic nature of marketing and production environments. The level of competition in the international marketplace is higher than ever before, and it is unlikely to decline in the future. Only manufacturers that are able to adopt suitable facility layout working practices with other related issues will be able to cope, compete, and thrive. Manufacturing companies should be sufficiently flexible to deal with changes to product mix, demand, and delivery time. To achieve the change that will help a company and its products remain competitive, a systematic and comprehensive approach to re-layout design is required. Therefore, manufacturing facilities need periodic reviews of the layout design, material handling system design, and day-to-day operations. Manufacturing companies must be continually striving for excellence in these aspects.

This research aims to address important issues in layout design, focusing on how to establish a comprehensive solution to the layout design of manufacturing facilities. The research goal in this thesis is to develop models and strategies for layout and re-layout design for dynamic environments in the cellular manufacturing domain. We believe this approach will help manufacturers develop appropriate strategies to adjust to recent and future anticipated changes in manufacturing environments and satisfy market demand. In this research, we will look at some of the underlying issues and link detailed designs of facility layouts to today's business environment. While optimizing facility layout is a vital decision, this cannot be done without research work in this area to understand the dynamics of plant layout in changing environments.

The layout problem in a manufacturing system involves determining the location of machines, manufacturing cells, departments, and other production and personnel support areas within a facility. Manufacturing cells or department arrangements in any facility must be designed to minimize material handling flow between departments or cells, minimize unnecessary personnel movement within the workplace, decrease congestion to allow easy flow of materials and personnel, make use of the existing space effectively, simplify communication and supervision within the workplace, and provide secure surroundings for workers and property. These determinations will lead to decreased manufacturing and material handling costs and increased productivity and system performance.

Some researchers argue that established manufacturing companies need to change the layout of departments every two or three years (Nicol & Hollier, 1983). The occurrence of layout changes has increased in recent years to a great extent because product mixes are changing more frequently now than ever before. Also, demands tend to fluctuate from time to time based on business environments and customer needs. Because we do not have the luxury to continually change facility layout design decisions, it is extremely important that, when making these decisions, we take into consideration their long-term impact.

1.4 General Types of Layout

In the process of designing a facility, the material flow pattern has to be determined first, after which the facilities designers can determine the type of layout to be implemented. The four general types of layouts are: process layout, product layout, fixed position layout, and group technology (GT) layout. These layout types are mainly used in manufacturing systems, as will be explained later in this section. There are also other new generations layouts mentioned in the literature, such as fractal, holonic, and distributed layouts (R. G. Askin, Ciarallo, & Lundgren, 1999), (Balakrishnan & Cheng, 2007), (Balakrishnan &

Hung Cheng, 2009), (Benjaafar, Heragu, & Irani, 2002). Fractal layout forms part of the focus of interest in this research, as it has been developed as an alternative for manufacturing job shops (process layout). In this research, we look at both GT and fractal layouts in order to explore the possibility of designing alternative layouts for manufacturing systems.

1.4.1 Process Layout

In a process layout, machines are arranged based on the processes they accomplish. Thus, similar machines or workstations are grouped and placed together in one department. For instance, lathes are located in one department, welding machines are located together in another department, and so on.

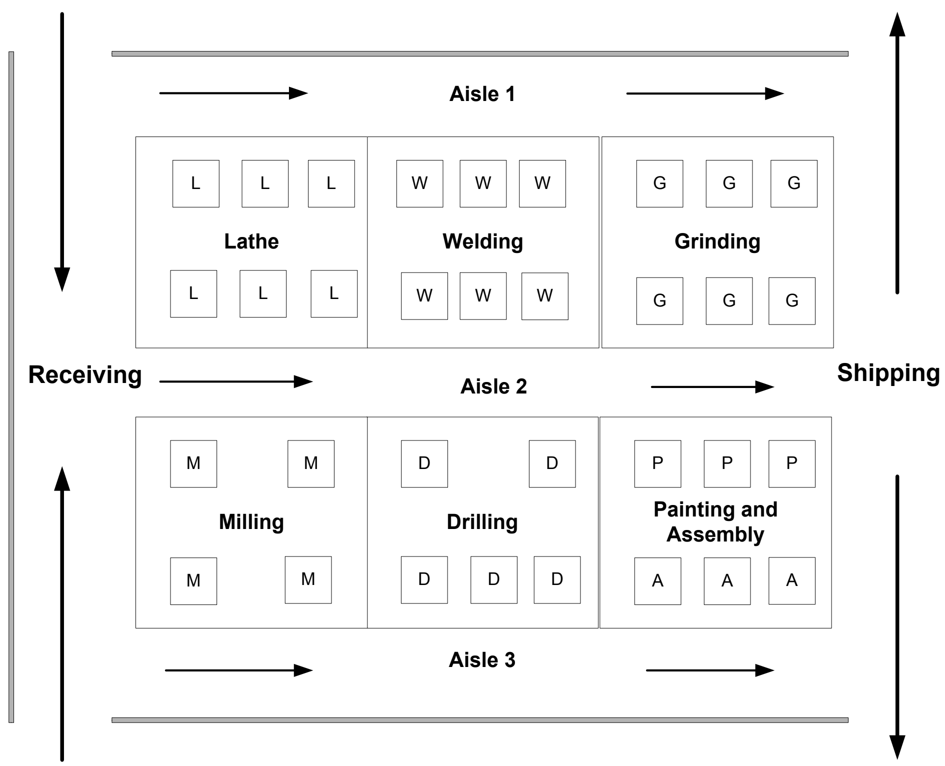


Figure 1-2: A typical process layout (functional layout)

Figure 1-2 shows a typical process layout. For example, a group of six lathe machines, indicated as L, are shown in the lathe department. Other departments, such as welding, grinding, milling and drilling, have a group of similar machines (indicated as W, G, M and D, respectively). Painting and assembly stations (indicated as P and A, respectively) are needed at the final production stage before packaging and shipping the manufactured products to their destinations.

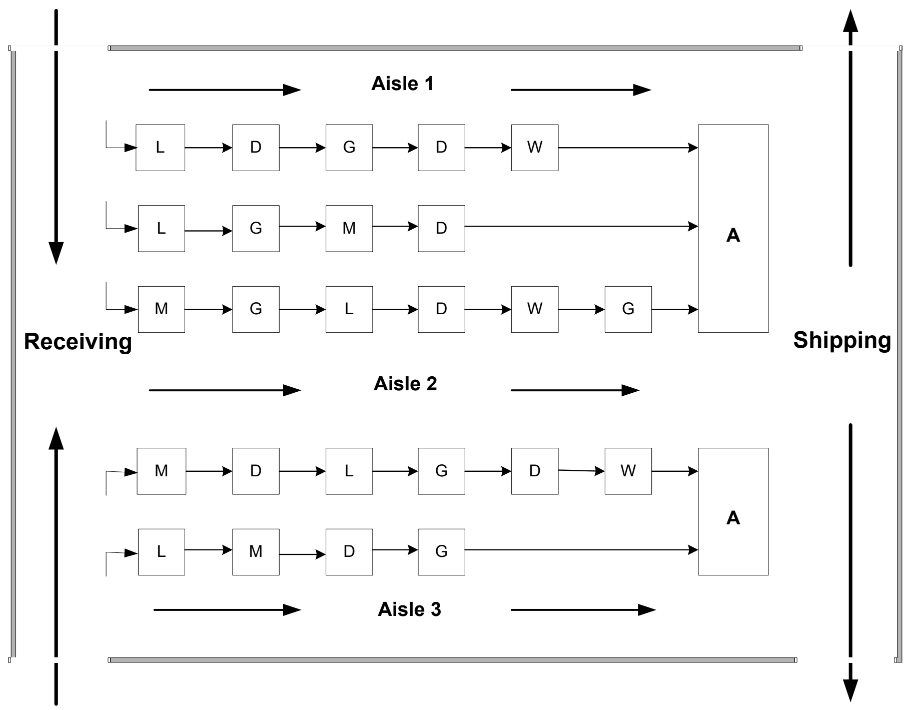
The process layout is useful for companies that manufacture a variety of products in small quantities, where each product is usually different from the others. Although the process layout offers flexibility and allows personnel to become experts in a particular process or function, it has some major disadvantages, such as increased material handling costs and traffic congestion, long product cycle times and queues, complexity in planning and control, and decreased productivity (Heragu, 2008).

1.4.2 Product Layout

The product layout is also known as production-line layout or flow-shop layout. The product layout involves the arrangement of machines and workstations according to the sequence of operations for the parts being manufactured. Materials usually flow from the first machine or workstation to the subsequent operations on the line while the finished products are ready at the end of the production-line. This type of layout is typically adopted for mass production, and the product layout generally results in large quantities of a single or few items (i.e., high-volume and low variety products).

As illustrated in Figure 1-3, the material is fed continuously from the first machine or workstation directly to the next adjacent machine or workstation in each production-line. It then goes through the subsequent operations until the final product is assembled prior to shipping. Advantages of product layout include simple product routing, reduced work in process (WIP) inventories, reduced material handling cost and time, reduced manufacturing time, and better production control. Product layout has multiple limitations, but the most important are low flexibility for changing the layout and possible interruption of manufacturing activities.

Once a specific product layout is adopted, the cost of changing the layout is substantial; therefore, product layout is not appropriate for companies that plan to manufacture high varieties of products or make frequent product changes. Furthermore, machine breakdown, shortage of materials, and absence of skilled workers may lead to stoppages in manufacturing activities.



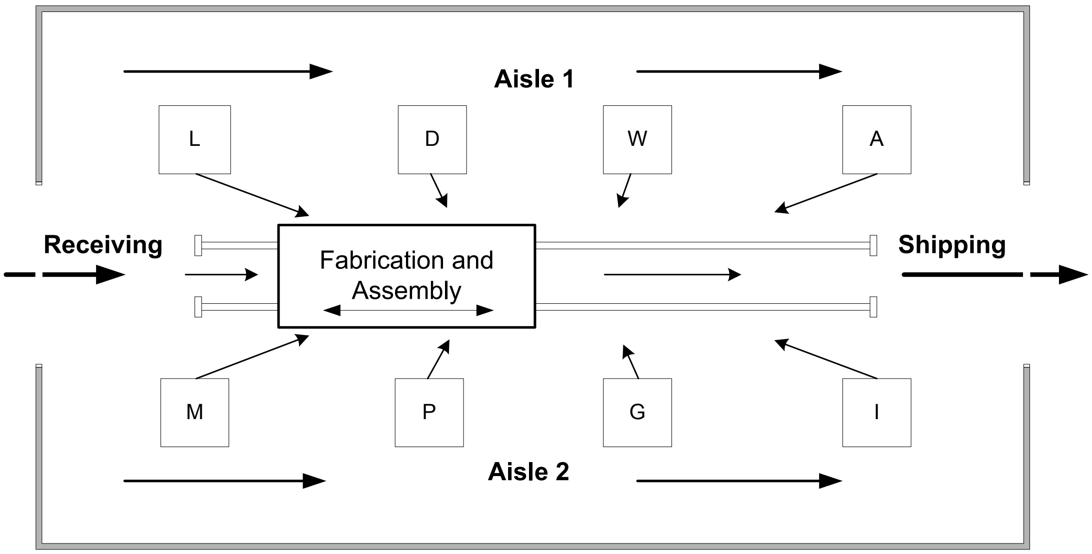
Note: L = lathe, M = mill, D = drill, W = weld, G = grinder, A = assembly

Figure 1-3: Product layout

1.4.3 Fixed Position Layout

In a fixed position layout, the concept is slightly different from other layouts. Namely, unlike other layouts, the product does not move from one machine or workstation to another. Instead, machines, workers and other related materials are taken to the location of bulky products, which remain stationary. This type of layout is typically adopted when the product is considered massive and cannot be easily transported through the manufacturing facility, such as ship building and aircraft assembly.

The advantages of using this layout are that the product is not frequently moved from location to location, thus reducing transportation costs and the chance of damaging the bulky product. The disadvantages of this type of layout are the relatively high costs of moving equipment to and from the working area and the low equipment utilization (Heragu, 2008).



Note: C = cutting, L = lathe, M = mill, D = drill, W = weld, G = grinder, P = paint, I = inspection

Figure 1-4: Fixed position layout in aircraft assembly

Fixed position layouts are common in shipbuilding. Also, aircraft manufacturing companies have adapted the fixed position layout to allow the aircraft to move slowly during assembly, as shown in Figure 1-4 (this layout is actually a hybrid fixed position and product layout). The benefit of this development is to facilitate several disparate teams working on the product to be scheduled to minimize conflicts while respecting precedence constraints.

1.4.4 Group Technology-Based Layout and Cellular Manufacturing

Group technology (GT) espouses the general philosophy that since similar products are generally manufactured using similar manufacturing processes, they can be grouped and manufactured together (R. G. Askin & Standridge, 1993). GT breaks down the product set into composite product families, within each of which, products are manufactured using similar processes and procedures (Irani, 1999).

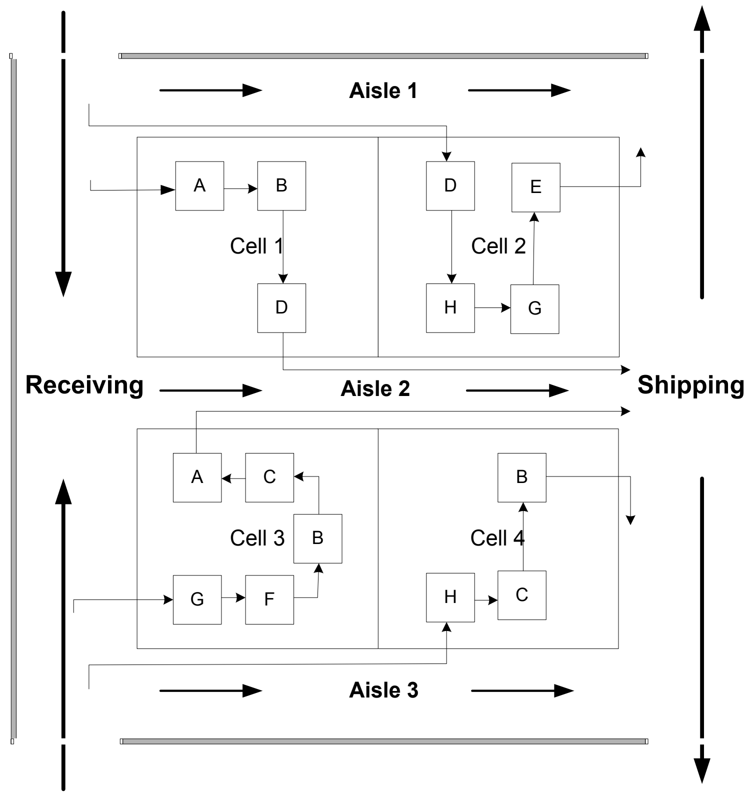


Figure 1-5: Cellular manufacturing layout based on (GT)

The machine grouping, as illustrated in Figure 1-5, may have duplicated machines that are used specially for reducing intercellular material handling or when the capacity of one machine cannot handle the entire product or part family. Each cell in GT layout may have different and similar machines depending on the assigned product or part family. Similar machines in a cell are not necessarily identical in their processing capabilities. For example, two lathes may differ in their setup or processing capabilities. GT-based layout attempts to lower WIP inventory, reduce material handling and traffic congestion, simplify employee training, and reduce production lead times, and has proven to be very successful when implemented properly. Earlier studies by (Pullen, 1976) and (Houtzeel & Brown, 1984) have shown the following improvements:

- Throughput time reduction (5-90%)
- Work-in-process inventory reduction (8-80%)
- Material handling reduction (10-83%)
- Job satisfaction improvement (15-50%)
- Fixturing requirements reduction (10-85%)
- Setup time reduction (2-95%)
- Space reduction (1-85%)
- Quality improvement (5-90%)
- Finished goods inventory reduction (10-75%)

Cellular manufacturing (CM) is defined as an application of GT that involves grouping machines based on the parts being manufactured. The main objective of CM is to identify machine cells and parts families simultaneously, and to allocate parts families to machine cells in a way that minimizes the intercellular movement of parts. To implement the cellular manufacturing systems (CMS) concept successfully, analysts must develop the layout of machines within the cells to minimize inter-and-intracellular material-handling cost (Heragu, 2008). CM has been applied successfully in many manufacturing environments and can achieve significant benefits (Black, 1983).

In CMS, the primary purpose of the cell is to reduce the material handling cost, WIP inventory, setup time, and labor cost. Consequently, it is also designed to process a wide range of parts, make material-flow more efficient, improve quality, improve space utilization, and make communication simpler. CM is often used for a family of products, has equipment that is correctly and specifically sized for the entire cell (usually arranged in a 'C' or 'U' shape so the incoming materials and outgoing finished products are easily monitored), and has cross-trained people for flexibility. CM involves the use of manufacturing cells that can be formed in a variety of ways, the most popular of which involves the grouping of machines, employees, materials, tooling, and material handling and storage equipment to produce families of parts. CM is often associated with just-in-time, total quality management, and lean manufacturing concepts and techniques (Tompkins, 2003).

As presented above, the advantage of GT is essentially to lower production costs and improve the quality of manufactured products. The implementation of cellular manufacturing could, however, have some disadvantages compared to other traditional layouts such as process and product layouts. The disadvantages are as follows:

1. Cell implementation usually leads to an increase in investment, as certain machines may need to be replicated to construct independent cells. Consequently, in implementing CMS, companies weigh the operational benefits such as reduced WIP inventory and throughput time against the costs of increased investment (Vakharia, 1986).

2. GT is less flexible than a process layout. Usually, a product is completely processed in a single cell. However, some products or parts may be processed in different cells due to the non-availability of machines. GT is generally not flexible enough to deal with major changes to product or product demand (Irani, 1999).

3. Due to the imbalance of utilization of equipment and operators, as mentioned above, some machine types must be replicated among several cells. This may result in a decline in machine utilization.

1.4.5 Fractal Layout

Fractal layout has been developed as an alternative for manufacturing job shops that allocate the total number of machines for most processes equally across several fractal cells. In many ways, a fractal cell layout can also be considered a CMS where the fractal cells, as shown in Figure 1-6, are similar units capable of manufacturing all products. Fractal cell layouts minimize the flow while providing flexibility. Each fractal acts as an independent unit, resulting in a highly decentralized system. Although fractal cells have the flexibility to handle high product variety, the investment and maintenance aspects of fractal layouts can be very expensive compared to other layouts for the same production (Venkatadri, Rardin, & Montreuil, 1997).

The other types of facilities that were indicated earlier in this chapter are holonic and distributed layouts. In holonic layouts, machines are placed seemingly at random through the plant. Each machine is a holon (meaning, an independent part of the whole) which, together with the other machines, forms the whole facility. Parts are assigned throughout the facility based on machine availability and processing capability (R. G. Askin et al., 1999). Distributed layouts are those where machine replicates are strategically distributed across physical space (Irani, Cavalier, & Cohen, 1993).

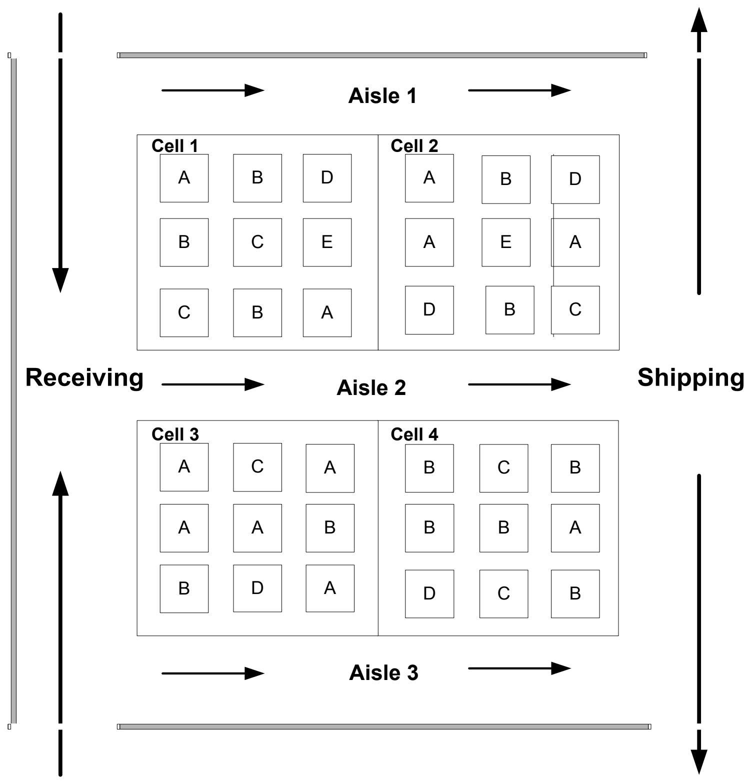


Figure 1-6: Fractal layout

In this research, we are interested in robust cell formation for cellular manufacturing systems in general, including GT and fractal cell layouts. The reason for including fractal cells and not holonic or holographic is because the latter two do not have clearly defined cells.

According to our definition, robust cell formation should be:

1. Adaptable. The system should be able to adapt to changes in demand variability, product mix, etc., mentioned earlier in this chapter.

2. Dynamic. Design decisions for future time periods should be explicitly considered during the design stage in the first period. In this research, we look forward to exploring how robust cell formation for cellular manufacturing systems can be designed under certain and uncertain situations.

1.5 Designing a Cellular Manufacturing System for Dynamic Environments

A new manufacturing facility requires a sizeable initial investment. This is true for both large companies and small to mid-sized enterprises (SMEs), which have to invest a substantial amount of money to acquire facilities and purchase processing and material handling equipment. Such a large investment is not without risk. Moreover, since changing or moving equipment can be time-consuming and expensive, the design of a manufacturing facility requires careful analysis. The impacts of design decisions are far-reaching because the facility is expected to last for a long period of time.

Manufacturing facilities have various types of process requirements that imply different types of equipment and workstation. For example, these processes include, but are not limited to: forming and shaping processes (i.e., metal-rolling, metal-forging, metal extrusion, drawing, sheet-metal forming), machining processes (i.e., turning and hole-making, milling, abrasive machining and finishing operations) and joining processes (i.e., welding, brazing and soldering). The production processes are performed on machines or workstations such as heavy-duty presses, flexible machining centers, and cutting, bending and welding equipment. The material handling infrastructure, which may include heavy-duty overhead cranes, conveyer belts, automated guided vehicle systems and rail transfers, makes revisions to the facility layout design extremely expensive.

Any large facility is designed in terms of smaller units (cells or departments). Estimating how many machines or workstations are required in these units, calculating the space requirements for cells or departments, locating cells or departments within the facility, locating machines or workstations within cells, and designing a material handling plan are some of the tasks involved in designing the overall facility. Therefore, it is important to develop methodologies to generate and evaluate alternative facility designs to minimize the costs of investment, production, and material handling over the lifetime of the facility.

In Chapter 1, we mentioned how discrete part manufacturing facilities are product-oriented (assembly lines, flow lines, etc.), process-oriented (e.g. job shop layouts), or cell-based (e.g. cellular manufacturing). Other organizations are also mentioned in the literature, such as fractal cells and holographic cells. The fractal cell is essentially a cellular organization, while the holographic layout is a highly distributed layout.

Layout design is a challenging task for any organizations because there are many trade-offs that should be considered. For example, by using a larger number of machines or workstations, the material handling cost may be reduced but the investment cost goes up. Also, there is a trade-off between flexibility, production, and investment costs. For example, by purchasing flexible machines, we may reduce the cost of material handling, but this will increase production and investment costs because flexible machines may be more expensive to purchase and operate.

As well, there are several considerations that need to be taken into account, such as short- and long-term demand variability for the products, range of products to be produced, uncertainty in processing (e.g., varying yields), dealing with machine downtimes due to preventive maintenance or failure (predictable and unpredictable). In addition, when designing a facility, one needs to take into account how production planning will take place within its four walls in order to ensure that the flow is smooth, that bottlenecks are alleviated, and that congestion and interference (e.g., between overhead cranes) are reduced.

1.6 Research Background

This thesis deals with both the design of layouts operating under stochastic environments and dynamic layout design.

As mentioned in the previous chapters, there are many different types of layouts, such as traditional (product, process, group technology) and new generation (fractal, holographic, holonic, and distributed). (Venkatadri et al., 1997), (Montreuil, Venkatadri, & Rardin, 1999), and (Benjaafar et al., 2002) expand on these, arguing that layouts need to be flexible, modular, and reconfigurable.

There are two key considerations in designing a layout: whether the environment is uncertain (stochastic layout design), and whether the layout will likely need to change in the future (dynamic layout design). In this thesis, we are interested in both aspects as they relate to the cell formation problem.

Numerous papers in the literature deal specifically with designing layouts for stochastic environments, as seen in Chapter 2. The methodology and settings used in these papers are varied, but in the two-phase stochastic programming setting, the first phase decision is the layout, while the recourse variables are to change the routings, add or remove machines, or re-layout. Changing the routing is often the cheapest recourse, with products simply being rerouted elsewhere. For example, in GT, when demand changes and a cell no longer has the capacity for a process (machine), a fraction of the product demand may be rerouted to another cell that has the remaining capacity on the process (machine). Adding or removing machines is the next most expensive option, but unless machines are replaced in an existing facility, the addition of machines is possible when space is available. In certain cases, a machine or a group of machines (cell) may be introduced by space expansion in a direction (for example, a wall may be extended). Complete re-layout is very expensive but sometimes inevitable.

Therefore, in light of these likely expenses and disruptions, the strategy used by most plant organizations to deal with changes over time or even random changes is to tolerate suboptimal layouts as long as rerouting options are available. When this becomes a regular occurrence, the organization may consider minor expansions or adjustments (swapping machines, renovating cells, etc.) or may decide to re-layout or relocate.

The CFP in stochastic and dynamic settings has been studied mostly in GT and cellular manufacturing contexts. The reference model in the literature for cellular manufacturing systems design is (Wicks & Reasor, 1999). They considered the dynamics of the production environment by incorporating a multi-period forecast of product mix and demand into their model. The authors presented a mixed-integer non-linear program with quadratic and cubic terms for the design of cellular manufacturing systems under fluctuation in the demand for products and product mix. The objective function in their formulation was to minimize the total cost of material handling and machine relocation over a forecast period. One extreme solution for the cellular manufacturing design problem is to purchase as few machines as possible, resulting in a high cost of material handling. The other extreme solution is to duplicate machines indiscriminately to reduce inter-cell traffic, but this strategy results in higher acquisition costs. The objective is to find an intermediate design within these extremes in the spectrum.

The model suffers from the following three limitations. First of all, it is difficult to solve because it has non-linear and integer variables. Secondly, the model assumes that each part has only one machine type sequence, which is very restrictive. With the choice of technologies in modern manufacturing, it may be possible to use a 5-axis CNC milling center to machine a part as one possible sequence. On the other hand, the part may also be machined using a routing through conventional machines such as lathes, drilling machines, and milling machines. It is important to be able to model the inherent trade-offs resulting from alternate routings. i.e., the trade-offs between the costs of high technology equipment with simpler routings versus lower technology equipment with complex routings. Thirdly, the model implicitly assumes growth in demand. The authors had a positive variable to represent the number of machines of a type acquired in a period. However, with negative demand scenarios, it may be necessary to discard machines. Even with the restriction on one machine type sequence, finding an exact optimal solution to the cell design problem using the Wicks and Reasor approach is difficult. The authors propose GA as the preferred solution mechanism for the problem.

A promising avenue for further research which we will explore through this research is how RO can be incorporated into cellular manufacturing systems design using the approach in (Mulvey & Vanderbei, 1995).

1.7 Research Objectives and Thesis Organization

This research has three objectives:

1. Developing an efficient integer linear programming formulation for the dynamic cell formation problem that extends the literature, such as the models in (Wicks & Reasor, 1999), (Vila Gonçalves Filho & José Tiberti, 2006a), and (Tunnukij & Hicks, 2009).

2. Exploring the central backup cellular manufacturing system (CBCMS), which is a hybrid manufacturing design that has features of both GT and fractal layouts.

3. Extending the model developed in objective (1) to deal with uncertainties in demand using the robust optimization framework.

The remainder of the thesis is organized as follows:

In Chapter 2, we present a review of the literature pertinent to our research. This includes cell formation problems (CFP) in cellular manufacturing environments, fractal layout organization, routing flexibility, layout optimization, dynamic facility layout problem, dynamic cellular manufacturing under uncertainty, and robust optimization.

In Chapter 3, an efficient multi-commodity network flow based formulation for the multi-period cell formation problem is presented, along with some case studies and computational results.

The concept of the central backup cellular manufacturing system (CBCMS) is introduced in Chapter 4 to understand and explore the design continuum from GT to fractal cell layouts. We extend the MPCFP model presented in Chapter 3 to design GT, fractal, or

CBCMS layouts using side-constraints. We then analyze how a designer can generate these different types of layouts based on efficiency and flexibility requirements.

In Chapter 5, a robust optimization formulation of the MPCFP in Chapter 3 is developed to deal with uncertainties in demand during cell design. We present computational results obtained by applying the robust model to an example taken from the literature.

Finally, the thesis concludes in Chapter 6 with a summary of thesis contributions and recommendations for future research in this area.

Chapter 2: Literature Review

The literature review relevant to this research is classified into topics that have relevance to robust cell formation and the design of efficient and flexible cellular manufacturing systems.

These topics are:

1. The cell formation problem (CFP) in cellular manufacturing environments.
2. Fractal layout organization.
3. Routing flexibility.
4. Layout optimization.
5. Dynamic facility layout problem and dynamic cellular manufacturing under uncertainty.
6. Robust optimization.

2.1 The Cell Formation Problem in Cellular Manufacturing Environments

Over the last three decades, a number of researchers have studied the CFP in cellular manufacturing design. Several mathematical models and solution algorithms have been developed for the CFP and employ different methods for solving it. These include: matrix arrangement, similarity coefficient analysis, graph theory, mathematical programming, heuristics, and meta-heuristics. Meta-heuristic methods include approaches such as simulated annealing (SA), genetic algorithm (GA), and TABU search.

(Papaioannou & Wilson, 2010) presented a number of solution methods that have been used for the CFP in the last decade. Matrix arrangement methods deal with the arrangement of rows and columns of a part-machine matrix to form a block diagonal matrix. Parts families and machine groups can be formed from the blocks, with each block representing a manufacturing cell (King & Nakornchai, 1982) (Chan & Milner, 1982).

Similarity coefficient methods classify parts families and machine groups based on the similarities between parts or machines. Similarities are based on the machines, tools, and fixtures required by the parts. Different measures of similarities have been developed to form parts families and machine groups (Seifoddini & Wolfe, 1986). (Yin & Yasuda, 2005) present a taxonomy and review of similarity coefficient methods published in the literature and point to a three-step procedure attributed to (Romesburg, 1987) as the commonality between similarity coefficient methods:

1. Form the machine-part incidence matrix.
2. Construct the similarity matrix.
3. Use a clustering algorithm to process the similarity matrix to obtain a diagram called a tree (dendogram), from which groups are obtained.

Another paper by (Yin & Yasuda, 2005) presents a comparative investigation to evaluate the performance of various similarity coefficient methods applied to CFP. The authors classify the similarity coefficients into two categories: efficient (three similarity coefficients) and inefficient (four similarity coefficients) for solving the CFP. They also found that the Jaccard similarity coefficient is the most stable similarity coefficient.

In graph theoretical methods, the machine-part matrix is represented by a graph. The aim of this method is to obtain sub-graphs from the machine-part graph to identify parts families and machine groups. For example, (Ribeiro, 2009) computes the dissimilarities between parts and organizes the production system into parts-families and group-machines. A graph is generated and a coloring algorithm is used to obtain a number of cells equal to the desired number of cells.

Mathematical programming may also be used to solve the CFP. In these methods, parts families and machine groups can be formed simultaneously based on the solution to a mixed integer programming (MIP). The simplest formulation of the problem is based on clustering, where cells are formed from the machine-part incidence matrix in order to minimize the number of exceptional elements (Boctor, 1991) (Elbenani & Ferland, 2012). (Wang, 2003) proposed two linear assignment models to solve the machine-cell and parts family formation for the design of CMS. The performance of the developed linear assignment algorithms has been demonstrated to be very effective and efficient, especially for dealing with large-sized problems.

Beyond the standard clustering or assignment approach, designing cells requires making a trade-off between duplicating machines in one or more cells and increasing material movement between cells. The textbook by (R. G. Askin & Standridge, 1993) presents one such model. (R. Askin, Selim, & Vakharia, 1997) proposed an interactive cell formation method that can be used to design flexible cells. In their work, the authors illustrated routing flexibility (i.e., the ability for the cellular system to process parts within multiple cells) and demand flexibility (i.e., the ability of the cellular system to respond quickly to changes in parts demand and parts mix). A mathematical model for assigning operation types to machine types was presented, in which the objective function minimizes the total annual operating cost of the operation-machine assignments along with the total annualized procurement cost of machines. (Selim, Askin, & Vakharia, 1998) introduced a mathematical formulation that includes two additional dimensions of the CFP. The first dimension is grouping workers, and the second dimension deals with tooling. The dynamic cell formation and worker assignment problem is simultaneously proposed by (Aryanezhad, Deljoo, & Al-e-hashem, S. M. J. Mirzapour, 2009).

Heuristic methods use rules that guide the search process. Under this classification, there are heuristics and meta-heuristics, such as SA, GA, and TABU search. Heuristics represent decision procedures and rules of thumb that expert users employ to solve a problem. (Heragu & Gupta, 1994) developed a heuristic method for forming parts families and machine groups. They addressed several constraints, such as machine capacity, technological requirements, cell size, and number of cells within the proposed layout. This algorithm allows the user to change the values of the parameters and arrive at a new solution quickly.

(Kochikar & Narendran, 1998) developed a heuristic algorithm for solving CFP for flexible manufacturing systems (FMS). The authors assumed deterministic demand in terms of mix and volume over the planning horizon and introduced heuristic procedures based on maximizing flexibility to assign parts families to machine cells in order to retain as much flexibility as possible for parts scheduling. The heuristic uses a grouping criterion that reflects the multi-faceted nature of flexibility and is, in effect, a composite of routing, machine, and parts transfer flexibility. Manufacturing cell flexibility is defined by the authors as a composite of three flexibility measures: productivity, processivity, and transferability. The evaluation shows that the heuristic has a tendency to create a large number of small cells.

(Liu, Yin, Yasuda, & Lian, 2010) proposed a mathematical model to deal with the CFP that incorporates production factors such as production volume, batch size, alternative process routing, cell size, unit cost of inter- and intra-cell movements, and path coefficient of material flow. A three-stage heuristic algorithm was developed to solve the NP-hard problem. In this algorithm, the first step is to form the temporary machine group plan according to the alternative process routing of each part. The second step is to select the appropriate process routing of each part with respect to the overall material movement cost. The last step is to configure the regular manufacturing cells based on appropriate process routing. The computational results for several problems showed that the approach provides only locally optimal solutions.

Among the studies in the literature that use meta-heuristics to solve the CFP is one by (Vila Gonçalves Filho & José Tiberti, 2006b). Their paper presents a group GA for the cell layout design problem and includes several new features such as the chromosome codification scheme, the correction mechanism, and crossover and mutation operators. These work directly with a group of machines, as opposed to individual machines. The algorithm is based on group encoding instead of simple machine encoding generally used by most GA implementations encountered in the literature. It is necessary to run the algorithm several times to ensure the best solution. However, when grouping machines, the number of parts has no effect on the size of the space solution, making the algorithm attractive for problems where the number of parts is large.

(Mahdavi, Paydar, Solimanpur, & Heidarzade, 2009) proposed a mathematical model for cell formation in CMS based on the cell utilization concept. The aim of the model is to minimize simultaneously the number of voids and exceptional elements in cells to achieve higher cell utilization. The authors presented an algorithm based on GA to solve the mathematical model.

(Tunnukij & Hicks, 2009) presented the enhanced grouping GA that has been developed to solve the CFP without predetermining the number of manufacturing cells or the number of machines and parts within each cell. The enhanced grouping GA uses a greedy heuristic and employs a rank-based roulette-elitist strategy, which is a new mechanism for creating successive generations. The quality of the solutions obtained from enhanced grouping GA was compared with other methods using the grouping efficacy measure. The results show that the enhanced grouping GA is effective.

(Cao & Chen, 2005) developed an optimization model integrating cell formation and parts allocation to generate a robust system configuration to meet probabilistic production demands in a number of probabilistic composite scenarios. The model considers the trade-off between the system cost and the expected material handling cost. The TABU search process was developed to find the optimal or near-optimal solution to the NP-hard problem.

Another issue important to the CFP is that of stochasticity in demand. The CFP in stochastic and dynamic settings has been studied mostly in GT and cellular manufacturing contexts. (Harhalakis, Ioannou, Minis, & Nagi, 1994). It focuses on cell formation under random product demand and presents an approach to obtain robust shop decompositions. This approach aims to come up with a cellular design that has satisfactory performance over a certain range of demand variations. The statistical characteristics of external demand and the capacity of the system resources are both considered. The design objective is to minimize the expected material handling costs, while constraints are imposed by resource capacities and cell size limits. A two-stage design approach is presented. In the first stage, the feasible production volumes, given the distribution of independent demand, are determined, and are used to compute the design criterion for the candidate shop configurations. This design stage provides a link between forecast market demand and feasible production volumes upon which the cell formation process should be based. In the second stage, near-optimal cell formation is determined using a grouping method.

It should be noted here that we are interested in the dynamic CFP over a multi-period planning horizon with varying product mix and demand. The dynamic evolution of a CMS is captured by extending the CFP model to multiple periods (MPCFP). (Tavakkoli-Moghaddam, Aryanezhad, Safaei, & Azaron, 2005) presented a traditional meta-heuristic method for solving the MPCFP. In this paper, a nonlinear integer model of the MPCFP was presented and then solved by GA, SA, and TABU search. (Ghotboddini, Rabbani, & Rahimian, 2011) used a decomposition algorithm to solve the dynamic CFP and also considers worker assignment over a multi-period planning horizon.

Dynamic CFP in a group layout setting has been studied by several researchers. (Deljoo, Mirzapour Al-e-hashem, Deljoo, & Aryanezhad, 2010) expanded previous dynamic cell formation models presented in the literature and used GA as the solution methodology.

(Jayakumar & Raju, 2010) presented a case study problem to illustrate the applicability of their proposed model in a dynamic production environment for multi-period planning. The model has the advantage of forming machine cells and parts families simultaneously and addresses the dynamic nature of the production environment by considering a multi-period forecast of product mix and varying demand during the formation of part families and cells. The objective of the model is to minimize the sum of machine constant cost, the operating cost, and the inter- and intra-cell material handling cost for the given periods.

The reference model in the literature for the MPCFP design is from (Wicks & Reasor, 1999). They considered the dynamics of the production environment by incorporating a multi-period forecast of product mix and demand into their model. The authors presented a mixed integer non-linear program with quadratic and cubic terms for the design of CMS under fluctuations in the demand for products and product mix. The objective function in their formulation is to minimize the total cost of material handling and machine relocation over a forecast period. Finding an optimal solution to the cell design problem using the formulation in (Wicks & Reasor, 1999) formulation is difficult, however, so GA is proposed as the preferred solution mechanism.

2.2 Fractal Layout Organization

Since this thesis proposes the CBCMS layout, which is a hybrid layout within the fractal and GT cell layouts, it is important to take into account the relevant research in fractal layout. The fractal layout has been developed as an alternative for manufacturing job shops (Montreuil et al., 1999) and essentially allocates machines for most processes equally across several fractal cells. While fractal cells have flexibility to handle parts variety, fractal layout setup costs can be prohibitive compared to other layouts for the same production. Several researchers proposed different methodologies and models for fractal manufacturing. Some of these methodologies and models are presented in the following paragraphs.

(Tharumarajah, Wells, & Nemes, 1996) discussed ionic, fractal, and holonic manufacturing. Also discussed in the paper are the underlying principles on which these concepts are based, along with a comparison of their design and operational features. The authors provided concrete examples of shop-floor applications as envisaged by these concepts. (Montreuil et al., 1999) proposed the fractal cell layout for job shop environments, and compared the performance of the fractal layout with those obtained using the function layout, group layout, and holographic layouts. They concluded that the fractal layout presents an interesting compromise with low handling and low investment costs while offering flexibility. However, setup costs increase in this type of layout (Balakrishnan & Cheng, 2007).

(Venkatadri et al., 1997) introduced a multi-commodity network flow model (MCNFM), a special form of linear program to minimize travel distance. The solution approach to the fractal layout problem confronts the issue of flow assignment when more than one replicate of a particular type is present in a cell or on the floor. The authors presented an integrated design methodology for the design of a fractal cell shop, in which a decomposition approach is used to perform assignment and layout tasks. Computational results are shown in the paper. Experiments are conducted to find out whether the fractal layout provides flow efficiencies comparable to the group layout and capacity requirements close to the function layout. Seven cases are analyzed for four basic job shop designs: functional layout, pure group layout, holographic layout, and fractal layout. In all cases, the function and holographic layouts use the least number of workstations, whereas the functional layout has the worst flow performance. In two out of seven cases, the group layout performs the best in terms of the flow distance measure (total across the layout of flow multiplied by distance). The fractal layout performed well with respect to flow distance, outperforming all other designs in five out of seven cases. Also, it used marginally more workstations than the functional layout, with significant reduction in flow distance.

(R. G. Askin et al., 1999) investigated fractal and holonic approaches to locating machines. They developed heuristic procedures for assigning machines to locations for holonic layout and also proposed routing algorithms for routing jobs between machine centers in fractal and in holonic systems and assigning jobs in fractal systems. A simulation study compares these approaches to process layouts on the basis of cycle time, material handling, and uniformity of work load across machines. The holonic and fractal approaches are found to be potentially useful layout concepts.

(Saad & Lassila, 2004) introduced various fractal cell configuration methods for different system design objectives and constraints. A mathematical model and a TABU search-based fractal layout design algorithm were developed to optimize product distribution to the cells and the arrangement of machines and cells on the shop floor. Three heuristic algorithms were used in the simulation program, and a TABU search algorithm was used for all configuration methods to optimize the internal machine layout within cells. Two simpler heuristic methods iteratively search a neighborhood for optimal permutations of external cell layout. The proposed fractal cell configuration methods are applied to two case studies using the developed TABU search-based program for fractal layout design. The results show that, in fractal layouts, a trade-off is required between machine quantity and travel distance. Also, it is possible to reduce travelling distance by increasing the degree of optimization on machine layout and product distribution for a specific product demand and mix.

(Shin, Mun, & Jung, 2009) proposed a self-evaluation framework of a manufacturing system that facilitates continuous and quick adaptation. The framework adopts fractal organization for its principle control architecture. In a self-evolutionary manufacturing system, each production resources regulates its own goal by responding to changes in its environment.

2.3 Routing Flexibility

(Sethi & Sethi, 1990) defined routing flexibility in a manufacturing system as being the ability to manufacture a part using alternate routes through the system. (R. Askin et al., 1997) defined routing flexibility as the ability of the cellular system to process parts within multiple cells. Routing flexibility is important in any CMS. Having this flexibility allows production schedulers to use alternate routings in real-time when machines are overloaded or cannot be used due to maintenance. Many researchers have studied routing flexibility in the context of cellular manufacturing systems.

(R. Askin et al., 1997) proposed a cell formation method that considers routing flexibility and responds to changes of demand and part mix. The method balances cost and flexibility through four phases. Phase I deals with cost reduction by minimizing the fixed machine costs and direct processing costs. Phase II takes advantage of manufacturing similarities and the operation flexibility of machines to assign parts-operation to machines. As a result, lower material handling costs are achieved. Phase III involves grouping machines and creating cells by balancing material handling and system flexibility, and Phase IV increases the flexibility of the design by evaluating and improving cell configuration.

(Nagi, Harhalakis, & Proth, 1990) considered projected production and distribute demand among alternate routings in order to obtain better cell formation. The developed linear programming (LP) formulation addresses routing selection and cell formation simultaneously. The objective is to minimize parts traffic while satisfying the demand of parts and machine capacity constraints. (Harhalakis et al., 1994) consider the CFP with product demand variation. Demand statistics and resource capacities are considered in the cell formation stage to minimize the expected inter-cell material handling cost.

While some researchers in the literature considers cell formation and parts routing selection as two independent problems, (Sankaran & KasiIingam, 1990) integrated them. The authors developed a 0-1 integer programming formulation to select parts routings and to form cells based on the total system costs of processing and annual machine operations. The cells are formed based on the total system costs while providing routing flexibility in the system. (Ramabhatta & Nagi, 1998) present an integrated formulation of the CFP with planning issues in the form of a 0-1 mixed integer linear programming model based on (Sankaran & KasiIingam, 1990). The formulation considers the issues of alternative routing, resource capacity, and operation sequences that impact inter-cell material handling. To solve the typical industrial-size problems, a branch-and-bound algorithm is developed to provide provable optimal or near-optimal solutions compared to those obtained by the heuristic of (Sankaran & KasiIingam, 1990). In another paper, (Caux, Bruniaux, & Pierreval, 2000) presented a method to solve the CFP with alternative routings and machine capacity constraints. The method simultaneously solves the CFP and the parts-routing assignment problem. They proposed a branch-and-bound algorithm for selecting optimal routings, which in turn minimizes the inter-cell transfer while respecting machine capacity constraints. The use of the branch-and-bound algorithm is limited for large-sized problems or unconstrained problems due to computational time.

The approach in (Adil, Rajamani, & Strong, 1996) was to select an alternate plan for each part as well as simultaneously group parts and machines. The authors developed a non-linear integer programming model to solve the CFP. The model identifies parts families and groupings of machines simultaneously while considering alternative routings. The model is then transferred into a linear integer programming model to solve optimally the CFP for small instances. However, for large problems, a simulated annealing algorithm is developed to obtain an efficient solution. (Jayaswal & Adil, 2004) considered factors such as operation sequence, machine replications, and alternate process routings simultaneously when solving the CFP, developing a model and solution methodology. The objective is to minimize the sum of costs of inter-cell material handling, machine investment and machine operation with respect to all factors mentioned above. To solve the model, a solution algorithm comprising simulated annealing and local search heuristics is developed. Computational results show that the algorithm generates a good quality solution and is efficiently capable of solving large problems with 100 parts and 50 machines.

(Nsakanda, Diaby, & Price, 2010) presented a formulation of the capacitated parts-routing problem in manufacturing systems with routing and processing flexibility. The problem is formulated as a network flow-based LP model, which minimizes total material handling, production, and outsourcing costs with respect to machine capacity limits and parts demands. The authors develop a price-directed decomposition-based approach based on Dantzig-Wolfe to exploit the network sub-structure of the model, and thereby do not require an enumeration of the possible routes for all parts. The computational results involving industrial-sized large scale problems are conducted to show cost gains from system flexibility.

(Sofianopoulou, 1999) developed a nonlinear integer programming model to solve the CFP for manufacturing systems where multiple replicates of similar machines exist and alternative process plans for parts types are available. The processing sequence for each part type is taken into account in order to determine the exact amount of inter-cell moves. The objectives are to assign machines and parts families to cells and to determine parts routing in order to minimize the total amount of inter-cell moves. A heuristic algorithm based on two-dimensional simulated annealing is employed to solve the model, and the computational results are shown on medium-sized problems. In another paper, (Spiliopoulos & Sofianopoulou, 2007) presented a bounding scheme that allows all combinations of alternative routing to be examined when solving only a few cell formation problems, thereby limiting the solution space, which is searched heuristically. This leads to increased reliability of the solutions obtained. The computational results indicate that this approach is viable for problems where the average inter-cell move for the CFP sub-problem solved is not excessive.

2.4 Layout Optimization

In this section, we provide a brief literature review of spatial cell design and then highlight the importance of the spatial cell model in designing the arrangement of cells and the machines within the cells. The facility or block layout problem (FLP) is used to find an optimal two-dimensional block layout for departments, assuming that the flow between any pair of departments is adequately represented by a rectilinear flow between their centroids (Gunn & Venkatadri, 2008). The solution of the FLP is a block layout that provides information about the dimensions of the cells and their location within the facility. In brief, the three main constraints on the FLP are: a) non-overlapping constraints; b) building perimeter restrictions; and c) cell area constraints. In addition to constraints on cell shape, another factor is a cell's location within a building, input/output stations, etc. (Saraswat, 2006).

(Kusiak & Heragu, 1987) and (Meller & Gau, 1996) have studied the FLP extensively, (Tompkins, 2003) indicated that an efficient layout configuration can result in a substantial reduction in the initial investment and operational costs. In this research, we do not deal directly with the FLP. However, for researchers seeking a comprehensive solution to the layout problem, the use of sequence pair representation to represent the relative position of the cells and machines within the cells in a layout (spatial cell design) seems promising. The sequence pair concept was first introduced by (Murata, Fujiyoshi, Nakatake, & Kajitani, 1996) for large-scale VLSI chip design. (Meller, Chen, & Sherali, 2007) proposed a formulation for FLP based on the sequence pair approach. The core issue of the FLP is to find a sequence pair, which is a set of two permutations that define the relative positioning of the department, when taken together (Gunn & Venkatadri, 2008).

The continuous space machine layout design problem consists of linear variables for department size and location and zero-one variables to represent the relative locations of machines with respect to each other (left, right, above, below). The spatial layout design problem is NP-Hard and very difficult to solve optimally for instances larger than 13 machines in sequence. If the FLP has various objectives, then the problem is a multi-criteria problem, and thus a multi-objective design is needed to solve the FLP. However, most solution methods for the FLP consider the use of single objective (mainly flow distance) as an alternative for the actual cost.

2.5 Dynamic Facility Layout Problem and Dynamic Cellular Manufacturing under Uncertainty

Dynamic and uncertain factors stemming from shorter product life cycles, product mix variability, demand volume variability, shorter delivery time and a high level of customization are factors to consider when designing cellular or other types of facility layouts. Manufacturers must be able to respond to such changes and uncertainties with reasonable investment and operating costs. A number of researchers have proposed different methods to deal with the layout design problem for such environments. The literature may be generally classified into papers on dynamic facility design (which deals with how a layout should be designed over time) and stochastic facility design (which deals with how to design a layout to take uncertainty into account).

(Rosenblatt, 1986) first proposed the dynamic facility layout problem (DFLP) and suggested dynamic programming to solve the problem. (Montreuil & Venkatadri, 1991) introduced a linear programming formulation for the DFLP with unequal area and variable shape departments. Pair-wise interchanges heuristics for the DFLP was proposed by (T. Urban, 1993), whereas (Conway & Venkataramanan, 1994) used GA search for the same problem and (Kaku & Mazzola, 1997) used a TABU search heuristic for the dynamic layout problem (DLP). (Balakrishnan, Jacobs, & Venkataramanan, 1992) model the constrained dynamic plant layout problem (DPLP) as a singly constrained shortest path problem and compare it to dynamic programming. Other examples of dynamic layout models in process or group layout settings are (Balakrishnan & Cheng, 2009) and (McKendall Jr. & Hakobyan, 2010). Some review papers have recognized that a methodology for the combined stochastic and dynamic layout design is needed (Balakrishnan & Cheng, 2007) and (Kulturel-Konak, 2007), but, to our knowledge, there have been no such implementations.

Several studies have been conducted to address flexible plant layout. (Yang & Peters, 1998) proposed a flexible machine layout design using a heuristic procedure based on a construction-type layout algorithm. The developed procedure solves a robust machine layout design problem over a multi-period planning horizon. The design procedure considers demand uncertainty and is not restricted to equal size machines. (Benjaafar &

Sheikhzadeh, 2000) presented an approach for the design of plant layouts in stochastic environments. (Chen, 1998) developed an integer programming model to minimize the overall manufacturing cost in a dynamic environment. In this paper, the presented model and the solution method are designed to generate manufacturing cells that may be sustained for a planning horizon of multiple time periods.

(T. L. Urban, 1998) developed a methodology utilizing incomplete dynamic programming to find the optimal solution to the DFLP with fixed rearrangement costs at exceptionally reduced solution times. The solution requires the calculation of T (T+1)/2 solutions to the quadratic assignment problem (QAP). They present a heuristic solution methodology for larger problems. A strong lower bound is developed for the general problem that dominates all existing bounds, and it is shown how the bound can be used as an initial test for optimality before the dynamic program is solved. An improved upper bound is established utilizing incomplete dynamic programming that dominates any previously established bound.

(Kochhar & Heragu, 1999) presented a framework for the design of a dynamic facility that can respond effectively to changes in product design, product mix, and production demand. They demonstrated a technique for a multiple-floor, dynamic facility layout that considers different aspects such as automated material handling systems, fixed versus variable path material handling devices, the size of storage, and acceptable inventory levels. Although the decisions concerning these aspects are made after the block layout has been generated, these aspects affect the block plan that represents an efficient layout. A genetic algorithm-based heuristic is used for solving the design problem. Using the layout generated by the algorithm, the facility designer can then apply "if/then" analysis and non-quantifiable criteria to assess the suitability of each alternative. Their approach is to provide the facility designer with a tool that does not necessarily generate the best design, but tries to produce a number of reasonably good alternatives.

(Balakrishnan & Cheng, 2000) adapted an existing GA for solving the DLP, suggesting a procedure that calls a GA with nested loops. The inner loop uses a steady state replacement approach, and replaces the most "unfit" individual in each generation. The outer loop replaces a large number of “unlucky” individuals in a generation. The proposed GA differs from the existing implementation in three ways: first, they adopted a different crossover operator to increase the search space; second, they used mutation to increase population diversity; and third, they used a new generational replacement strategy to help increase population diversity. The conducted study shows that the proposed GA is quite effective.

(Balakrishnan, Chun, & Conway, 2000) investigated the design of a facility layout based on a multi-period planning horizon. In today’s market-based, dynamic environment, layout rearrangement may be required during the planning horizon to maintain layout effectiveness. The authors proposed a few algorithms to solve dynamic plan layout problem by using dynamic programming and pair-wise exchange based on a previous method introduced by (T. Urban, 1993). Furthermore, they suggested some computationally efficient improvements to Urban’s pair-wise exchange procedure, with their tests showing that these improvements are worth implementing. It is reported also that these could be used to solve large problems.

(Erel, Ghosh, & Simon, 2003) presented a new heuristic based on the idea of viable layouts to solve the DLP. The authors focused on the solution of the basic DLP by following approaches developed by (Rosenblatt, 1986) and (Balakrishnan et al., 1992). Their objective was to obtain the optimal sequence of layouts. Given all possible layouts, the DLP can be viewed as a shortest path problem on a multi-stage, directed, acyclic network with costs on both nodes and arcs. Each stage corresponds to a time period in the planning horizon, with the nodes at any stage representing all possible layouts and the arcs between the nodes in two consecutive stages signifying the moves from one layout in one period to possibly another in the next. The node cost is the flow cost of the associated layout in the given time period and the arc cost is the relocation cost between two successive layouts. The proposed scheme includes two main phases. In the first phase, a viable set of layouts is identified, and in the second phase, the shortest path problem is solved over this set. The proposed scheme is reasonably flexible in that the user can control certain parameters to acquire a desirable balance between solution speed and accuracy. Their computational results show that this scheme is competitive with the other available solution methods.

(Balakrishnan & Cheng, 1998) investigated the design of facility layouts based on multi-period planning horizons in which the material handling flows between the different departments in the layout may change. The authors considered intermediate settings when the costs of layout rearrangement are neither negligible nor prohibitive. The dynamic layout approach maintains the balance between the material handling and layout rearrangement costs. They adopted the dynamic facility LP model based on (Balakrishnan et al., 1992), in which the objective was to minimize the sum of the layout rearrangement costs and the material handling over the planning horizon.

(Dunker, Radons, & Westkamper, 2005) presented an approach that can handle unequal department sizes that may change form one period to the next. They introduced a mathematical model for the DFLP for departments of unequal size (i.e., departments of different sizes that can change from period to period). The workshop floor area and departments were divided into rectangles, the shape of which is determined by its side lengths, allowing it to change from period to period. This enabled the authors to model expansion, shrinking or replacement by a new department. The position is described by the coordinates of the center point and the orientation. There are two possible orientations: the long side is parallel either to the x axis or to the y axis. The cost of a layout for a given period will depend on the distance between certain points attached to the departments or to the workshop floor. To solve the model, the authors presented an algorithm combining dynamic programming and genetic search for solving the DFLP.

(Balakrishnan & Hung Cheng, 2009) investigated the DFLP under a rolling horizon and uncertainty. One of the objectives was to examine whether algorithms that work well under the fixed horizon situation will perform as well under rolling horizons. They first solved the fixed period problems using these algorithms to determine whether any algorithms stand out from the rest. Then they solved rolling horizon problem using these algorithms. This allowed the authors to test whether there are algorithms that work consistently well under fixed and rolling horizons. They investigated the performance of algorithms under fixed and rolling horizons, under different shifting costs and flow variability, as well as under forecast uncertainty. The results show that algorithms that dominated under fixed horizons may not work well under rolling horizons. Moreover, the general consensus is that it is difficult to identify an algorithm that performs well under all situations.

(McKendall Jr. & Hakobyan, 2010) introduced a heuristic for DFLP with unequal-area departments. The DFLP is the problem of finding positions of departments on the plant floor for multiple periods (where material flows between departments change during the planning horizon), such that departments do not overlap and the sum of the material handling and rearrangement cost is minimized. The authors proposed a boundary search technique that places departments along the boundaries of already placed departments. This is used to construct a solution for the DFLP for different cases. For example, the departments may have unequal-areas, in which departments area are fixed for each period but may vary from one period to another. Furthermore, a department may have free orientations, or the layout for each period may usse the continuous representation of the plant floor, etc. A mixed integer linear programming (MILP) mathematical model is presented for the DFLP with unequal-area departments. The authors presented a solution technique for the proposed heuristic that can be represented as a vector of department period pairs. The boundary search heuristic (BSH) consists of a selection and placement procedure. In BSH, which is a binary search algorithm embedded within BSH, there is a search for feasible area along the boundary of already placed departments. Once the initial layout plan is obtained using BSH, a TABU search heuristic, called TS/BSH, is used to improve the layout plan.

(Drolet, Abdulnour, & Rheault, 1996) presented the virtual cellular organization and the dynamic cellular organization. Both concepts permit gains in terms of performance and flexibility with the use of computer and information technology. As the authors mentioned, the virtual cellular manufacturing and dynamic cellular manufacturing could be a profitable and interesting alternative to classical cellular manufacturing.

(Balakrishnan & Cheng, 2005) combined cellular manufacturing research and dynamic layout research (dynamic layout research deals with designing layouts when product demands change over time) to model manufacturing cell formation under dynamic demand. Techniques range from the simple to the sophisticated and flexible, with the simple usually manipulating (part-machine matrices). The sophisticated techniques handle many constraints in forming cells, such as: maximum cell size, different demands for different products, different number of cells, and the presence of set-up costs. Traditionally, cells are created by grouping parts which are produced into families based on the operation required by the parts. Cells consist of common machines which are physically grouped together and dedicated to producing the parts families. The authors suggested an alternate framework, proposing to change the cellular configuration periodically when the cost-benefit analysis favors such a move. In this way, the cellular layout will be better suited to the demand in each period and thus be more effective and agile during the planning horizon. Also, they proposed examining multiple layouts when considering cell redesign in order to incorporate different qualitative and quantitative considerations. According to their results, as cell rearrangement cost get higher, the job shop may be preferred to cellular manufacturing.

(Balakrishnan & Cheng, 2007) conducted a literature review to categorize research that has been done to address cell reconfiguration and uncertainty issues in CMS. First, they described a deterministic model for CMS reconfiguration due to planned product changes. Second, they discussed the issue of uncertainty in demand or product mix, along with various issues in CMS such as robust design, part reallocation, fractal layout, virtual manufacturing cells, hybrid cells, modular layout, routing flexibility, and multi-objective system selection. They also presented a mathematical programming formulation for multi-period planning with cell reconfiguration. In this model, they assumed that a reasonable forecast of new product introductions and parts mix or volume change can be made so that a multi-period plan is possible. The design of the cell should be effective for varying parts mix and uncertain volume, since it is often difficult to predict how successful a particular model will be. Additionally, there may be resource uncertainties such as machine breakdowns.

The process of solving the CFP is less complex if the product mix and demand are deterministic (i.e., the product mix and demand are known in each period). (Seifoddini, 1990) presented a probabilistic model to solve the dynamic cell formation problem, while (Suer, Huang, & Maddisetty, 2010) proposed a resource sharing concept to deal with unstable demand. The authors developed a layered CMS to form dedicated, shared and remainder cells to deal with probabilistic demand. Simulations are used to compare the performance of the layered cellular manufacturing with the classical CMS.

(Braglia, Zanoni, & Zavanella, 2005b) investigated analytical issues that affect the design of a flexible layout, discussing the problem of layout design in an uncertain environment when market variations may determine significant fluctuations in the flows between resources. This fact may enhance the need for layout reconfiguration, as material handling costs may increase due to the change in traffic flows. However, the plant reconfiguration may be avoided by adopting a robust layout, which can support mix and/or volume fluctuations. After introducing the stochastic layout problem, they develop a mathematical model, and a case of normally distributed flows is discussed by simulation experiments, addressing the layout robustness concern. According to their results, the layout degradation mechanisms are interpreted, proposing an index for the a priori evaluation of the layout robustness. Two theorems are presented. The first one show that, in a dynamic environment, the layout that minimizes the total expected costs may be found by studying only the matrix of average flows between the resources. The second theorem shows that, under reasonable hypotheses, the cost function of each possible layout fits a normal distribution whose mean and variance are analytically measurable. They also discuss the analytical calculation of the foreseeable costs related.

(Norman & Smith, 2006) presented a formulation of the facilities block layout problem which considers uncertainty in material handling costs on a continuous scale by use of expected values and standard deviations of product forecasts. A genetic algorithm meta-heuristic with a flexible bay construct of the departments and total facility area is used for solving the design problem. The design can be optimized directly for robustness over a range of uncertainty that is pre-specified by the user. The proposed formulation presents a computationally tractable and intuitively appealing alternative to earlier stochastic formulations that are based on discrete scenario probabilities.

(Drira, Pierreval, & Hajri-Gabouj, 2007) conducted a survey related to facility layout problems and classified the existing literature using criteria such as: the manufacturing system features, static vs. dynamic layout problems, formulation of layout problems, and solution approaches. They discussed the layout problems, which strongly depend on the specific features of manufacturing systems such as production variety and volume, material handling, multi-floor layout aspects, pick-up and drop-off locations, and facility shape and dimensions. They reported the importance of being more flexible when designing a facility layout. They also presented several mathematical formulations of the layout problems and describe how they can be solved. The solution strategy for static and dynamic layout can include graph theory or neural network. Models may further be classified as either single objective or with multiple objectives, and formulations may use a continual or discrete space representation. The authors also reported several types of optimization approaches, including exact methods such as branch-and-bound and approximate methods which include heuristic and meta-heuristics. Meta-heuristics are used to solve layout problems with larger size and more realistic constraints, while evolutionary methods are used to solve complex problems. Other approaches include hybrid methods, in which, for instance, meta-heuristics may be combined with heuristics or simulation. It is also possible to combine optimization and heuristic methods.

(Krishnan, Cheraghi, & Nayak, 2008) proposed a facility layout design model to determine a compromise layout that can minimize the maximum loss in material handling cost both for single and multiple periods. The authors described the development of three models for designing facility layouts under uncertainty. The first model is a single-period multi-scenario model in which the objective is to minimize maximum losses (min-max approach) of a facility layout problem that has multiple possible demand scenarios by proposing a new compromise layout. The second model is a multi-period, multi-scenario model in which the objective is to minimize maximum loss due to material handling costs for multiple periods while taking into account the transition cost (i.e., costs related to changes in layout from period to period). The previous models assume equal probability of occurrence for each scenario and treat scenarios with very low probability on par with scenarios with high probabilities. The third model was formulated to minimize the total expected loss. It considers the associated probability of occurrence of each scenario and helps to generate a compromise layout that can minimize the total expected loss from all scenarios rather than reduce the maximum losses of specific scenarios. The developed model addresses the minimization of the total expected loss as well. The resulting mathematical models are solved to generate improved layouts using a two-pass GA to determine the layout. The proposed models are solved for single-period and multi-period case studies, and were compared to an existing model in the literature that used total expected flow. Results indicate that the proposed models generate compromise layouts that are efficient in reducing the risks associated with facility layout design when dealing with multiple production scenarios.

(Lahmar & Benjaafar, 2005) presented a procedure for the design of distributed layouts in settings with multiple periods where product demand and product mix may vary from period to period and where a re-layout may be undertaken at the beginning of each period. They presented a multi-period model for jointly determining layout and flow allocation within a dynamic distributed layout problem. The objective is to design a layout for each period that balances re-layout costs between periods with material flow efficiency in each period, i.e., to minimize the sum of material flow costs and rearrangement costs over a planning horizon consisting of numbers of periods. There are two limiting cases to the dynamic distributed layout problem. The first is where rearrangement costs are insignificant, allowing us to solve a series of independent single-layout problems. The second is where rearrangement costs are prohibitively high, allowing us to combine all the flows from all the periods and then solving a single-period layout problem.

The solution procedure shows that the exact solution procedure for the distributed layout can be obtained based on using a branch-and-bound algorithm. (Lahmar & Benjaafar, 2005) offered a decomposition-based heuristic to perform well relative to lower bounds. The heuristic approach is based on an iterative procedure in which they decompose the problem into two subproblems: a facility layout subproblem and a flow allocation subproblem. A solution is obtained by iteratively solving for a facility layout problem with fixed flows followed by a flow allocation problem with a fixed layout. Their solution algorithm, along with a data generating procedure, was implemented in a program application written in C and interfaced with the optimization solver Cplex. The authors examined the value of distributed layouts for varying assumptions about system parameters to draw several managerial insights. The authors showed that distributed layouts are most valuable when demand variability is high or product variety is low. Finally, they showed that department duplication (e.g., through the disaggregation of existing functional departments) exhibits strong diminishing returns, with most of the benefits of a fully distributed layout realized with relatively few duplicates of each department type.

(Benjaafar et al., 2002) reported that existing layout configurations do not meet the need of multiproduct enterprises and that there is a need for a new generation factory layouts that are flexible, modular, and easy to reconfigure. The tendency in research is toward layout design for dynamic and uncertain environments. The authors explored alternative layout configurations and alternative performance metrics for designing flexible factories, listing three approaches to layout design that address the three distinct needs of a flexible factory. The first two approaches may be thought of as novel layout configurations, namely, distributed and modular layouts. In the third approach, they use operational performance as a design criterion to generate agile layouts. They discussed several research challenges related to distributed layouts, modular layouts, and agile layouts. Also, they mentioned three main trends considered very important to industry. The first is the move toward lighter and more portable equipment, the second is the increased modularization of products and increased postponement in product differentiation, and the third is the shift to more flexible and scalable machines. While they noted that the above trends would not be applicable to all industries, designing layouts that are robust and able to sustain a wide range of products is critical.

(Meng, Heragu, & Zijm, 2004) discussed the reconfigurable layout problem, which differs from traditional, robust and dynamic layout generally in two aspects: first, it considers deterministic material handling and relocation costs in addition to assuming that production data are available only for the current and upcoming production period. Second, it considers stochastic performance measure such as WIP inventory level and production lead time. They proposed a process to solve the reconfigurable layout problem. An open queuing network-based analysis model developed previously, called the manufacturing system performance analyzer (MPA), is used. MPA is based on the parametric decomposition method to analytically evaluate the key performance measures of a queuing network. MPA is used to estimate the stochastic performance measures of a layout. These are combined with deterministic performance measures such as material handling cost to determine the layout for the next period.

2.6 Robust Optimization

Robust Optimization (RO) is a modeling methodology combined with computational tools to optimize problems in which some data elements are uncertain (Ben-Tal & Nemirovski, 2002). RO is gaining popularity in the optimization literature and is promising to be an important framework for solving practical optimization problems under uncertainty.

(Mulvey & Vanderbei, 1995) developed an alternative approach for robust optimization that integrates a goal programming formulation with a scenario-based description of problem data. In their abstract, the authors define a robust optimization model as follows: “a solution is robust if it remains close to optimal for all scenarios of the input data, and model robust if it remains almost feasible for all data scenarios.” In this approach, a series of solutions that are progressively less sensitive to the realizations of the model data from a scenario set are generated. The main idea is to penalize constraint deviations in the objective function of a mathematical program so that the trade-off between feasibility and optimality can be understood by changing the weight of the penalty coefficients. The authors illustrated RO concepts using several examples. The RO approach can generate robust solutions for several real-world application such as power capacity expansion, matrix balancing and image reconstruction, air-force airline scheduling, scenario immunization for financial planning, and minimum weight structural design.

Post-optimality studies such as sensitivity analyses are used to observe the impact of data disturbance on the model's recommendation. These kinds of studies are insufficient in themselves because they only observe the impact of data uncertainty on the model solution. Although data in some models are usually known with some probabilistic distribution, data in the real-world of operation research applications are often incomplete. This is one of the difficulties that operations researchers encounter being faced with the problem of noisy, incomplete or erroneous data. (Ben-Tal, El Ghaoui, & Nemirovski, 2009) in their book stated that some of data entries (future demand, returns, etc.) do not exist when the problem is solved and hence are replaced with their forecasts. These data entries are thus subject to prediction errors. Obtaining a satisfactory forecast for demand is therefore critical. Nonetheless, small uncertainties in data may affect the feasibility and optimality of a problem. In some cases, a small percentage of data uncertainties (as low as 0.1%) can make the optimal solution infeasible and thus practically meaningless.

(Ben-Tal & Nemirovski, 2000) conducted a study of 90 LP problems from the well-known NETLIB library collection based on ((Ben-Tal & Nemirovski, 1997); (Ben-Tal & Nemirovski, 1999); (El Ghaoui & Lebret, 1997) ; (El Ghaoui, Oustry, & Lebret, 1998)). This study shows how the feasibility of optimal solutions to LP problems can be affected by quite small perturbations of data. Robust solutions guard against uncertainty, and a decision-maker is, generally speaking, ready to make a small sacrifice in optimality to have a solution that is both feasible and reasonably good across several scenarios. Other papers include (Ben-Tal & Nemirovski, 2002) and (Bertsimas & Sim, 2004). (Bertsimas & Sim, 2004) proposed a robust approach for solving linear optimization problems with uncertain data. Their approach seeks to make RO less conservative by trying to find ways to decrease the price of robustness. The price of robustness can be flexibly adjusted by placing probability bounds on constraint violations. Therefore, their RO methodology provides: a) solutions with probabilistic guarantees that constraints will be satisfied, and b) the ability to vary protection bounds of constraint violation, depending on the constraint.

(Bertsimas, Pachamanova, & Sim, 2004) proposed a framework for the robust modeling of LP problems using uncertainty sets described by an arbitrary norm. Similarly, (Bertsimas & Brown, 2009) proposed a framework of coherent risk measures as a starting point for uncertainty set construction for robust linear optimization problems. A few years later, (Bertsimas, Brown, & Caramanis, 2011) conducted a survey for the primary research, both theoretical and applied, in the field of RO. In their paper, they focused on the following issues: a) the computational attractiveness of RO approaches, b) the power of modeling, and c) the broad applicability of RO methodology. They also present some results linking RO to adaptable models for multi-stage decision-making problems.

(Beyer & Sendhoff, 2007) published a review paper on the field of RO. Starting from Taguchi's robust design methodology, the authors considered the robust regularization approach based on worst-case scenarios usually used to find robust solutions to linear and quadratic constrained optimization problems. Several papers also describe various techniques of RO that have been tailored for specific application areas. Some examples are (Bertsimas et al., 2004) (portfolio optimization) and (Bertsimas, Nohadani, & Teo, 2010) (radiation therapy).

More recently, (Bredström, Flisberg, & Rönnqvist, 2013) proposed a method to solve production and distribution planning problems with uncertain parameters in a rolling horizon planning. The method is based on the decomposition principle, where they first solve an upper-level problem for the first time period in which the parameters are known. The lower-level problem uses the upper-level solution and computes a worst-case scenario for a given period with uncertain parameters.

(Braglia, Zanoni, & Zavanella, 2005a) developed layout stability indices based on flows (resulting from product demand and routing sheets). With these indices, it is possible to calculate the risk of deterioration in layout performance due to market dynamics. While the methodology here is not that of RO, the stability indices are important to understand.

(Cao & Chen, 2005) developed what they call a robust optimization model to form cells in manufacturing system to meet production demands expressed in a number of probabilistic scenarios. The proposed model integrates cell formation and part allocation to generate a robust system configuration to minimize machine cost and inter-cell material handling cost. The model considers the trade-off between the system cost and expected material handling cost. TABU search was developed to find the optimal or near-optimal solution to the NP-hard problem. While the word robust appears in the title of the paper, the approach used in the paper only minimizes expected total cost over several scenarios.

To our knowledge, the RO method has not been applied much in the cell formation or facility layout areas and therefore constitutes a very relevant topic of research to pursue. The main questions are: a) What robust factors need to be considered in the cell-formation problem? b) What are the trade-offs between optimality and feasibility in cell design? c) How can the dynamic CFP be formulated and solved in a robust optimization context?

Chapter 3: An Efficient Multi-Commodity Network Flow Formulation for the Multi-Period Cell Formation Problem

3.1 Overview

This chapter addresses the first objective of the thesis, which is to develop an efficient integer linear programming formulation for the multi-period cell formation problem (MPCFP) for cellular manufacturing systems (CMS).

The cell formation problem (CFP) is an important problem in CMS design. The MPCFP is an extension of the CFP in which the cell composition may change over time due to changes in demand and/or product mix. In this thesis, the CFP/MPCFP literature is discussed in section 2.1.

An underlying problem with models in the literature dealing with the MPCFP is that they are difficult to solve using exact methods. Therefore, the models use either restrictive assumptions (e.g. fixed routings) or heuristic solution methods. In this chapter, we will present a model for the MPCFP that can be solved computationally using the exact method. This model has an underlying structure (multi-commodity network flow) that provides modeling flexibility and solution tractability.

3.2 Cell Formation and Multi-Period Cell Formation Problems

The objective in both CFPs and MPCFPs is to minimize the total costs of production, material handling, machine acquisition, machine relocation, and machine disposal over the planning horizon. The decision variables are the parts and machines being allocated to cells. In the case of a CFP, there is a single planning horizon; thus, as a result, there are no relocation or disposal costs. In an MPCFP, relocation and disposal costs also come into play, and machine acquisition may be staggered. In both CFPs and MPCFPs, it is assumed that part demands and routings are known (across all periods, in the case of the latter).

There are several implicit complexities in the CFP/MPCFP. The simpler version of the problem is to form cells based on the machine-part incidence matrix in order to minimize the number of exceptional elements (Boctor, 1991). Such a formulation implicitly assumes that all machine replicates of a given type are in one cell in the final solution. This model, however, does not take into account the fact that if five replicates of a particular machine type are needed, they may be spread across three different cells. In practice, the layout designer would be interested in making a trade-off between the cost of material handling due to an inter-cell material handling transfer and the cost of replication of machines across various cells.

A cellular manufacturing system is considered pure if no transfer of parts is allowed between cells. However, the part mix and the machine requirements are such that some transfer between cells is inevitable. One way to reduce cell transfers is by duplicating machines in cells so that all process requirements of the parts allocated to a cell are in that cell. One extreme solution to the CFP is to duplicate as many machines as necessary to eliminate inter-cell transfers. However, this strategy results in higher acquisition costs. Another extreme solution is to purchase as few machines as possible, but this strategy would likely result in high material handling costs as well as undue complexity of the manufacturing system. The CFP models in the literature, such as the one in (R. G. Askin & Standridge, 1993), try to balance these costs. The drawback of this model is that it does not allow for flexible routings in alternate processing plans and is difficult to use when solving larger problems.

In this chapter, we present a new formulation for the CFP/MPCFP. The formulation is based on a multi-commodity network flow problem. The inputs are: multi-period demand; machine type-to-type routings (including alternate process plans); the costs of acquisition, disposal, and relocation of machines within the facility; and the cost of inter- and intra-cell material handling. The output is the flow allocation to the routings and the cell composition, which, in MPCFPs, are for each period. The objective is to minimize the total costs of cell formation (e.g., machine acquisition, disposal, and relocation costs), production, and inter- and intra-cell material handling.

While this model does not have a revenue component, it can handle fixed and variable costs of production and material handling. The goal of the MPCFP is to obtain a cellular design that performs well if/when the part mix and demand change over time.

While there are many models in the literature to solve the MPCFP, none of them recognize and exploit the underlying multi-commodity flow structure. The model in (Wicks & Reasor, 1999) is considered one of the main reference models for facilities design literature. Therefore, we will use the (Wicks & Reasor) model as a benchmark to assess and compare to our proposed model.

3.3 Structure of the Multi-Commodity Network Flow-Based Formulation for the MPCFP

To solve the MPCFP, we present a new multi-commodity network flow-based structured formulation. In this formulation, the problem may be visualized as a series of parallel flow networks, one for each period. This structure is powerful and is a very important contribution of this thesis. One reason why this structure can be implemented computationally is the rapid progress seen over the past few decades in software and hardware capabilities.

Let us first consider the CFP, which, by definition, is a single-period problem. The CFP has the following inputs:

- A set of products to be routed (with associated demands).
- A set of machines through which the products are routed (with associated capacities).
- Routings or processing sequences that are from machine type to machine type (with associated processing times for the process steps).

With regard to routings, one of the gaps in the literature is that most models do not easily accommodate alternate routings. The model developed in this chapter provides full flexibility and allows for any number of alternate routings to be specified. It naturally follows that increasing the number of routings increases the solution time.

The goal of the CFP is to form cells (i.e., to allocate machines and parts simultaneously to cells) so that total costs of production, material handling, and machine acquisition are minimized. While the cost of production depends on the machines being used and the products being manufactured, the cost of material handling increases when a product has to traverse one or more cells. This is usually captured using a linear inter-cell cost. The cost of machine acquisition varies depending on the type of machine and its technology (i.e., how quickly it processes, how many different processes it can handle, etc.). The solution to the CFP should be feasible, meaning that there should be sufficient capacity to handle production requirements. There are three parts to a CFP solution. These are:

- Product flows from machine replicate to machine replicate (modelled usually as continuous variables).
- Number of machines to purchase (modelled as integer variables).
- Machine allocation to cells (modelled as binary variables).

The MPCFP problem is similar to the CFP, the only difference being that cell formation in the MPCFP occurs over a discrete planning horizon. The solution in a planning period influences the solution in the next period while being influenced by the solution in the previous period. This is because machines can be purchased, relocated (from cell to cell), or disposed of.

In our formulation, the MPCFP is viewed as a series of flow networks, each corresponding to a discrete time period, as shown in Figure 3-1. Let us assume that we have a pre-specified number of cells based on a designer's considerations for each time period. Since each product has to go through one or more sequences of machines, the routings in each time period may be viewed as a multi-commodity flow network.

Each flow network (corresponding to a time period) in Figure 3-1 includes nodes and arcs. Nodes represent machine-cell combinations, and arcs represent product flows between machines. We allow for each machine type to be present in each cell in each time period. Therefore, the nodes represent combinations of machine types j, cell types k, at the time-period t. The flow of each product (commodity) from source (*S*) to destination (*E*) in each time period may be represented as a continuous variable.

The flow-networks in Figure 3-1 are linked through the n_{jkt} variables (i.e., the number of machines of type j in cell k in period t). The connection in the mathematical model between the flow networks happens through node balance constraints which allow n_{jkt} to change from period to period based on machine acquisition, relocation, or disposal.

The resulting model has an underlying multi-commodity flow network structure in each time period, which provides the structure for the CFP/MPCFP and gives better computational performance.

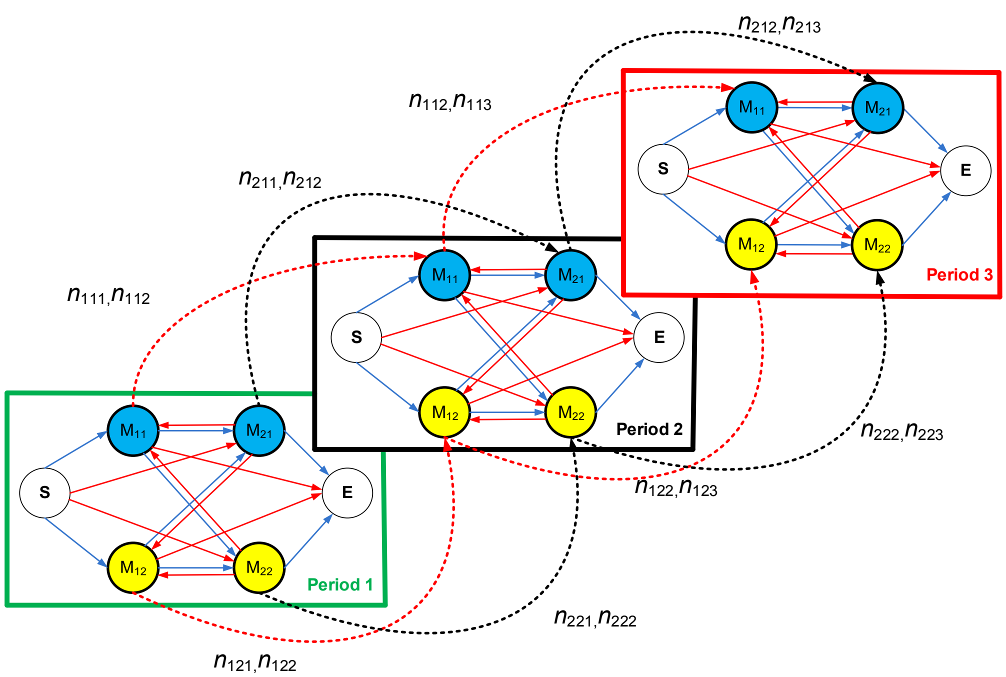


Figure 3-1: Time-phased arc-path multi-commodity network for multi-period cell design.

Referring to Figure 3-1, let us assume that there is a product that needs to be routed through machine types 1 and 2 (M_1 and M_2). Let us also assume that the routing sequences (M_1,

M_2) and (M_2, M_1) are valid for this product. There are two cells: cell one is represented in blue and cell two in yellow. Both machine types are duplicated in both cells. Let M_{jk} denote machine type j in cell k; note that there can be eight different paths or alternative processing sequences for the product. In the first time period or first flow network, an (S,E) path for the product could be any one of the following:

$(S, M_{11}), (M_{11}, M_{21}), (M_{21}, E)$

$(S, M_{11}), (jM_{11}, M_{22}), (M_{22}, E)$

$(S, M_{12}), (M_{12}, M_{22}), (M_{22}, E)$

$(S, M_{12}), (M_{12}, M_{21}), (M_{21}, E)$

$(S, M_{21}), (M_{21}, M_{11}), (M_{11}, E)$

$(S, M_{21}), (M_{21}, M_{12}), (M_{12}, E)$

$(S, M_{22}), (M_{22}, M_{11}), (M_{11}, E)$

$(S, M_{22}), (M_{22}, M_{12}), (M_{12}, E)$

The set $\{P_i\}$ is used in the model to represent all feasible start-to-finish paths for product i, such as the ones enumerated above for this example. $\{P_i\}$ may be generated beforehand, which makes our formulation an arc-path multi-commodity network flow model. By default, we assume full enumeration that takes into account all possible alternate routing sequences. However, partial enumeration is also possible if the designer wishes to omit alternative process plans or include constraints such as:

- Production for a product can occur only in one cell.
- Production for a product can occur in up to two cells.
- Production for a product can occur in up to two cells, but no more than one process can occur in the second cell.

It is important to stress that inter-cell material handling costs as well as manufacturing costs (i.e., the second term in the objective function) are path or routing dependent. They can be estimated in a pre-processing step where all combinations of alternate routings and cell visits are enumerated. The only questions for the model, then, are to decide how much flow should be allocated to a sequence and how many machines of each type should be maintained, period to period, by purchasing, discarding, or relocating them.

3.4 The Proposed Model for the Multi-Period Cell Formation Problem (MPCFP)

We can now formulate the MPCFP model. In brief, it can be defined as a non-spatial cell design model that considers multiple products with routing flexibility, bearing in mind that demands may vary across periods. In formulating our model, the following assumptions are made:

- A set of products i to be manufactured with alternative processing sequences S_i assigned to each part i, such that each step in S_i has a different machine of type j.
- Each cell k must have enough production capacity for machine type j if a sequence step is to be performed in that cell for part i. This is ensured by the variable n_{jkt} for cell k, machines type j, and time period t.
- Some assumptions are known beforehand, such as the cost of purchasing and discarding one unit of machine type j; the cost of relocating machine type j between cells; the cost of inter-cell material handling for each part i; the upper and lower limits of machine type j of each cell; the processing time per unit for part i on machine j; and the demand for part i in period t.
- The capacity of each machine type j is limited and known.

The indices in the model are:

i = An index of parts that need to processed.

j = An index of machine types.

k = An index of cells.

t = An index of time periods over which the system is being designed.

p = An index representing a start-to-finish path for part i.

The following set is used in the model:

$\{S_i\}$ = The set of all possible sequences for part i.

$\{P_i\}$ = The set of start-to-finish paths for product i based on $\{S_i\}$. This is pre-enumerated.

The parameters are:

D_{it}= Expected demand of part i in period t.

C_j= Time availability of one unit of machine type j per time period.

c_j= Cost of purchasing one unit of machine type j.

c'_j= Cost of discarding one unit of machine type j.

R_j= Cost of relocating machine type j between cells.

H_i^1= Cost of inter-cell material handling associated with the movement of one unit of part i.

H_i^2 = Cost of intra-cell material handling associated with the movement of one unit of part i.

LM = Minimum number of machines per cell (Lower limit).

UM = Maximum number of machines per cell (Upper limit).

h_p^1= Number of inter-cell transfers in path p.

h_p^2= Number of intra-cell transfers in path p.

m_{ipj}= Manufacturing cost per unit for part i on machine type j when routed through path p.

q_{ipjk}= Processing time per unit for part i on machine type j in cell k when routed through path p.

The decision variables are:

x_{ipt}= Number of parts of type i routed through path p in period t.

n_{jkt}= Number of machines of type j available in cell k in period t.

u_{jkt}= Number of machines of type j moved into cell k in period t.

v_{jkt}= Number of machines of type j moved out of cell k in period t.

a_{jkt}= Number of machines of type j purchased in cell k in period t.

b_{jkt}= Number of machines of type j discarded from cell k in period t.

The objective function is to minimize the total costs of:

- Purchasing, discarding, and relocating machines.
- Inter-cell/intra-cell material handling.
- Manufacturing.

Using the above notation, the mathematical formulation, including the objective function and system constraints, are now written in MIP form as follows:

Minimize Z:

$$\sum_j \sum_k \sum_t \left(c_j a_{jkt} + c'_j b_{jkt} + R_j u_{jkt}\right) + \sum_i \sum_{p \in P_i} \sum_j \sum_t (H_i^1 h_p^1 + H_i^2 h_p^2 + m_{ipj}) x_{ipt} \quad (3.1)$$

Subject to:

$$n_{jkt} = \begin{cases} a_{jkt} & if \quad t = 1 \\ n_{jk(t-1)} + a_{jkt} - b_{jkt} + u_{jkt} - v_{jkt} & if \quad t > 1 \end{cases} \quad \forall j, \forall k \quad (3.2)$$

$$\sum_k u_{jkt} = \sum_k v_{jkt} \quad \forall j, \forall t \quad (3.3)$$

$$LM \leq \sum_j n_{jkt} \leq UM \quad \forall k, \forall t \quad (3.4)$$

$$\sum_i \sum_{p \in P_i} q_{ipjk} x_{ipt} \leq n_{jkt} C_j \quad \forall j, \forall k, \forall t \quad (3.5)$$

$$\sum_{p \in P_i} x_{ipt} = D_{it} \quad \forall i, \forall t \quad (3.6)$$

The overall objective of the multi-period part family/cell formation problem is to minimize the total system cost. The total system cost in the objective function (3.1) consists of two sums. The first sum in the objective function (3.1) minimizes the sum of purchasing, discarding and relocating costs of overall machines, cells, and time periods. Calculating purchasing and discarding costs is done by multiplying the number purchased a_{jkt} or discarded b_{jkt} of each machine type j in cell k during time period t by the respective unit cost *(*c_j or c_j'*)* and sum of these over j, k, and t. For the relocating cost, every machine is relocated out of one cell v_{jkt} and into another cell u_{jkt}. It is sufficient to multiply the sum of the u_{jkt} variables by the unit cost of relocation R_j. The total cost of purchasing, discarding and relocating machines is $\sum_j \sum_k \sum_t \left(c_j a_{jkt} + c'_j b_{jkt} + R_j u_{jkt}\right)$.

The second sum in the objective function (3.1) measures inter-/intra-cell material handling and manufacturing costs. These are summed as overall parts i and part routings p in each multi-commodity flow network for each time period. H_i^1 stands for the unit handling cost per inter-cell transfer of part i, while H_i^2 is the unit handling cost per intra-cell transfer of part i. The total number of inter-cell and intra-cell transfers in path $p \in P_i$ may be pre-computed based on the sequence of cells visited by the path. Similarly, the manufacturing cost per unit m_{ipj} on path p may be calculated by summing the production costs on each of the machine types visited by the routings. Since x_{ipt} is the flow of part i using path p in time period t, the total cost of inter-/intra-cell material handling, and manufacturing (which is the second term in the objective function) is $\sum_i \sum_{p \in P_i} \sum_j \sum_t (H_i^1 h_p^1 + H_i^2 h_p^2 + m_{ipj}) x_{ipt}$.

The constraints in the model are (3.2) to (3.6), while constraint sets (3.2) and (3.3) are machine balance constraints. Constraint (3.2) ensures that the number of machines in a cell in the first time period $(t = 1)$ is equal to the number of machines purchased a_{jkt}. Otherwise, the number of machines in a cell during time period t >1 is equal to the number of machines in time period $(t - 1)$ plus the number of machines purchased a_{jkt} minus the number of machines discarded b_{jkt} plus the number of machines relocated u_{jkt} into the cell minus the number of machines relocated out of the cell v_{jkt}. Constraint (3.3) ensures that the total number of machines of type j relocated (moved into cell k) during time period t must be equal to the number of machines of type j relocated (moved out of cell k) in period t.

Constraint (3.4) limits the number of machines in each cell k during each time period t

based on lower and upper bounds. Constraint (3.5) is the capacity constraint in the model. Here, capacity is written in terms of processing time. However, it may be extended to include the availability of tools, labor, and other inputs such as machine setup time. The total processing time on machine type j, of which several might exist, in cell k during time period t overall part routings $p \in P_i$ is $\sum_i \sum_{p \in P_i} q_{ipjk} x_{ipt}$. This has to be less than or equal to the time availability of machine type j in cell k during time period t, i.e., $n_{jkt} C_j$. Constraint (3.6) ensures that the sum of production $\sum_{p \in P_i} x_{ipt}$ of part i through path p during period t should be equal to the demand for part D_{it}.

3.5 Multi-commodity Network Flow Structure in MPCFP

The multi-commodity network flow problem is well known in the Operations Research literature (Hu, 1963). The MPCFP model is structured around the arc path formulation of multi-commodity network flow problem (in our case, the MPCFP is also time-phased, with one multi-commodity network for each time period. The nodes of the multi-commodity network in each period are the machines in the various cells (*j,k* combinations), the commodities are the products (*i*).

The x_{ipt} variables (feasible routings) may be thought of as the start-to-finish arc-paths for each time period *t*. The cost of an arc-path x_{ipt} in the second sum of objective function is the sum of the costs of inter-cell and intra-cell material handling and the manufacturing cost at each machine in the arc-path.

The multi-commodity network flow problem has capacity constraints. The equivalent in the MPCFP is constraint set (3.5). These constraints ensure that the flow through the arcs in an arc-path is such that the capacity at each node is not violated. Similarly, the multi-commodity network flow problem has a demand constraint, which in the MPCFP is represented in constraint set (3.6).

Therefore, the second sum in the objective function and constraint sets (3.5) and (3.6) provide a time-phased multi-commodity network flow substructure within the MPCFP.

3.6 Toolkit for Solving the Optimization Problem (MPCFP)

A toolkit for developing optimization-based analytical decision support applications was developed. The MPCFP formulation was coded using the IBM CPLEX Optimization Studio (version 12.5).

Python 2.7 was used to generate $\{P_i\}$ using complete enumeration. It is the set of start-to-finish paths for product i based on the alternative process plans in $\{S_i\}$. As it generates the sequences, the Python code also pre-computes the inter-cell/intra-cell material handling and manufacturing costs associated with the paths. The Python output is then entered in CPLEX Optimization Studio. Appendices I and II list the CPLEX Studio and Python codes, respectively.

3.7 Numerical Examples and Computational Results (MPCFP)

Two problems are selected from the literature to evaluate the performance of the proposed model (MPCFP). These examples are representative of what is available in the literature and provide a general perspective on the applicability of the proposed model for solving the MPCFP.

The first illustrative example is adopted from (Wicks & Reasor, 1999) and the second case study is adopted from (Jayakumar & Raju, 2010). While the former paper is older, it has often been cited in the literature, having appeared in the widely read journal IIE Transactions. The latter is a relatively new paper that roughly summarizes the state of the art in MPCFP formulation and solution.

As stated in section 3.5, both problems are solved using IBM CPLEX Optimization Studio 12.5 to solve the optimization problem. We compare our results with those in the papers. It may be noted that the design decisions made by the MPCFP model are: part routings and quantities; cell configurations; and machine acquisition, disposal, and relocation decisions. The costs of inter-cell/intra-cell material handling along with production and machine acquisition, disposal, and relocation are a function of the design decisions.

3.7.1 First Illustrative Example

The illustrative example (Wicks & Reasor) consists of three time periods (index *t* in the model), 11 machine types (index *j* in the model), and 25 parts for production (index *i* in the model. Table 3-1 shows the parts, their processing sequences, inter-cell material transfer cost, and demand. In this example, there are no alternative processing sequences.

Table 3-1: Data for part types for (Wicks & Reasor illustrative example)

Part Number	Inter-Cell Material handling cost per unit	Operation Sequences	Demand		
		Machine (processing time)	Period 1	Period 2	Period 3
1	5	J(1)-A(5)-I(2)	300	200	500
2	5	E(6)-H(4)	700	600	500
3	5	A(1)-B(3)-K(4)	0	600	400
4	5	C(1)-J(1)-F(6)	0	700	800
5	5	B(3)-E(1)-I(4)	800	600	1000
6	5	E(4)-J(5)-H(6)	600	300	0
7	5	F(3)-E(6)-J(2)	0	900	800
8	5	D(4)-I(6)-K(1)	400	800	200
9	5	F(2)-J(6)-K(3)	300	200	600
10	5	C(2)-K(4)	400	1000	500
11	5	C(6)-A(3)-D(4)	0	0	200
12	5	G(3)-I(1)	700	700	1000
13	5	G(6)-A(4)-E(2)	100	600	800
14	5	G(1)-H(3)-J(3)	100	200	0
15	5	C(3)-I(2)-D(1)	0	0	300
16	5	D(3)-J(6)	500	800	500
17	5	F(6)-E(3)	100	900	400
18	5	A(2)-F(3)-J(3)	1000	1000	400
19	5	C(4)-F(1)-E(3)	0	700	1000
20	5	K(3)-I(4)-D(6)	800	300	500
21	5	H(1)-G(2)	500	400	1000
22	5	J(5)-B(3)-K(2)	0	100	100
23	5	I(1)-F(6)-J(3)	400	500	800
24	5	G(2)-B(4)	0	0	500
25	5	B(5)-G(6)-F(2)	0	0	400

Table 3-2 shows resource data about the available machines, their acquisition and relocation costs, and their capacities. The disposal cost for all machines is assumed to be zero (since disposal is not considered in the example). The operation sequences, processing times, machine capacities, and acquisition and relocation costs are assumed to be constant over the three time periods.

Table 3-2: Resource data for (Wicks & Reasor illustrative example)

Machine type	Acquisition cost ($)	Relocation cost ($)	Capacity (time units/period)	Number available at start of period 1
A	4000	2000	15000	1
B	7000	3500	18000	1
C	5000	2500	18000	1
D	9000	4500	19000	1
E	5000	2500	15000	1
F	3000	1500	17000	1
G	9000	4500	17000	1
H	7000	3500	19000	1
I	5000	2500	18000	1
J	8000	4000	15000	1
K	3000	1500	19000	1

The following cell capacity constraints were placed on the design of the CMS to be consistent with the example in the paper: 1) three machine cells to be formed; 2) the machine lower limit is three per cell, while the upper limit of machines in each cell is open-ended; and 3) each part has only one operation sequence, which implies that the manufacturing sequences is pre-defined.

3.7.1.1 Cell Formation for Each Period (Wicks and Reasor)

The authors in (Wicks & Reasor) develop a mathematical model that can be found in (APPENDIX 1) to solve the multi-period part family/machine cell formation problem. Since the model is a non-linear integer program with cubic and quadratic terms in the first constraint set, it is very hard to solve optimally. The authors use a genetic algorithm-based solution methodology to solve the problem. The solutions to the example problem obtained over the multi-period horizon are presented in Tables 3-3, 3-4, and 3-5 using the machine-part incidence matrix. These tables illustrate the recommended (Wicks & Reasor) multi-period design for each period (i.e., the recommended part family/machine cell formation for each period in the planning horizon).

Table 3-3: Machine-part matrix for period 1 (Wicks & Reasor model)

Cell	Machine (QTY)	Part Families															
		Part F1								Part F2			Part F3				
		1	8	9	10	13	16	20	23	14	17	18	2	5	6	12	21
1	A(1)	1				1											
	C(1)				1	1											
	D(1)		1				1	1									
	F(1)			1					1								
	I(1)	1	1					1	1								
	J(1)	1		1			1		1						*		
	K(1)		1	1	1			1									
2	A(1)											1					
	F(1)										1	1					
	G(1)									1							
	J(1)									1		1			*		
3	B(1)													1			
	E(1)					*					*		1	1	1		
	G(1)															1	1
	H(1)									*			1		1		1
	I(1)													1		1	

To accommodate the demands required in period one, cell one requires seven machine types (A, C, D, F, I, J and K) to process the part family of eight part types (1, 8, 9, 10, 13, 16, 20 and 23). Cell two requires four machine types (A, F, G and J) to process the part family of three part types (14, 17 and 18). In contrast, cell three requires five machine types (B, E, G, H and I) to process the part family of five part types (2, 5, 6, 12 and 21). Table 3-3 illustrates the formation of three cells for period one.

In period 1, the part flow between cells involves part types 13, 14, 17 and 6. In Table 3-3, the matrix elements with superscript * are known exceptional elements and are responsible for inter-cell movements. For instance, part type 13 in period one is processed in cell 1 by machine types A and C and then processed by machine E in cell 3. Therefore, part type 13 flows between cells on one occasion. In the same manner, part type 14 is processed in cell 2 by machine type G, then is processed by machine H in cell 3, and finally is processed by machine type J in cell 3. Therefore, part type 14 flows between cells on two occasions. The other parts that have superscript * also indicate flows between cells.

Table 3-4: Machine-part matrix for period 2 (Wicks & Reasor model)

Cell	Machine (QTY)	Part Families																				
		Part F1														Part F2			Part F3			
		1	4	6	7	8	9	10	13	16	17	18	19	20	23	3	14	22	2	5	12	21
1	A(1)	1							1			1										
	C(1)		1					1	1				1									
	D(1)					1				1				1								
	E(1)			1	1				1		1		1									
	F(2)		1		1		1				1	1	1		1							
	I(1)	1				1								1	1							
	J(1)	1	1	1	1		1			1		1			1							
	K(1)					1	1	1						1								
2	A(1)															1						
	B(1)															1		1				
	F(1)																					
	G(1)																1					
	J(1)																1	1				
	K(1)															1		1				
3	B(1)																			1		
	E(1)																		1	1		
	G(1)																				1	1
	H(1)			*													*		1			1
	I(1)																			1	1	

To accommodate the required demands in period two, cell one requires a group of nine machines, two units of machine type (F), and one unit of machine of types (A, C, D, E, I, J and K) to process the part family of fourteen part types (1, 4, 6, 7, 8, 9, 10, 13, 16, 17, 18, 19, 20 and 23), while cell two requires a group of six machine types (A, B, F, G, J and k) to process a part family of three part types (3, 14 and 22). In contrast, cell three requires a group of five machine types (B, E, G, H and I) to process a part family of four part types (2, 5, 12 and 21). Table 3-4 illustrates the formation of three cells for period two.

In period 2, the parts flowing between cells are part types 6 and 14. For instance, part type 6 is processed in cell 1 by machine types E and J and then processed by machine H in cell 3. In the same manner, part type 14 is processed in cell 2 by machine type G, then processed by machine H in cell 3, and finally processed by machine type J in cell 2.

Table 3-5: Machine-part matrix for period 3 (Wicks & Reasor model)

Cell	Machine (QTY)	Part Families																						
		Part F1																Part F2			Part F3			
		1	4	7	8	9	10	11	13	15	16	17	18	19	20	22	23	3	24	25	2	5	12	21
1	A(1)	1						1	1				1											
	C(1)		1				1	1	1	1				1										
	D(1)				1			1		1	1				1									
	E(1)			1					1			1		1										
	F(2)		1	1		1						1	1	1			1							
	I(1)	1			1					1					1		1							
	J(1)	1	1	1		1					1		1			1	1							
	K(1)				1	1	1								1	1								
2	A(1)																	1						
	B(1)															*		1	1	1				
	F(1)																			1				
	G(1)																		1	1				
	J(1)																							
	K(1)																	1						
3	B(1)															*						1		
	E(1)																				1	1		
	G(1)																						1	1
	H(1)																				1			1
	I(1)																					1	1	

To accommodate the demands required in period three, cell one requires a group of nine machines, two units of machine type (F), and one machine of types (A, C, D, E, I, J and K) to process a part family of sixteen part types (1, 4, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20, 22 and 23), while cell two requires six machine types (A, B, F, G, J and k) to process a part family of three part types (3, 24 and 25). In contrast, cell three requires five machine types (B, E, G, H and I) to process a part family of four part types (2, 5, 12 and 21). Table 3-5 illustrates the formation of three cells for period three.

In period 3, part type 22 flows between cells. This part type is processed in cell 1 by machine type J, then processed by machine B in cells 2 and 3, and finally processed in cell 1 by machine type K. Therefore, part type 22 flows twice between cells.

3.7.1.2 Implementing the MPCFP Model to Solve the Wicks and Reasor Example Problem

The data presented in the Wicks and Reasor illustrative example is used to test the MPCFP model in this thesis. The developed code in Python was used to generate all possible sequences for each part routing through machines and cells. The Python code can be found in (APPENDIX 2).

As mentioned earlier, each part in the Wicks and Reasor illustrative example has only one operation sequence. However, the MPCFP model in section 3.4 is capable of alternative sequences, although that is not mandatory. The Python code resulted in 549 possible sequences for this example. This is the number of start-to-finish feasible sequences considering the number of parts, the unique routing for each part, and the replications of each machine in each cell.

These are entered as input in the optimization model in IBM ILOG CPLEX Studio with the Wicks and Reasor data. The developed OPL model can be found in (APPENDIX 3). The CPLEX Studio runs on a 64-bit machine with an Intel i5 chipset running at 3.20 Ghz. The program came up with an optimal solution in the very short timeframe of 12.93 seconds. The outputs of our model are part routings with the given demands for the three time periods, cell configuration for all time periods, machine acquisition and relocation cost, and inter-cell material handling and manufacturing costs.

3.7.1.3 Cell Formation for Each Period

The optimal solution to Wicks and Reasor is presented in Tables 3-6, 3-7, and 3-8. First: To accommodate the demand required for period one, cell one requires three machine types (E, F and K) to process part type (17), while cell two requires four machine types (F, J(2) and K) to process part type (9). On the other hand, cell three requires eleven machine types

(A, B, C, D, E, F, G, H, I, J and K) to process a part family of fourteen part types (1, 2, 5, 6, 8, 10, 12, 13, 14, 16, 18, 20, 21 and 23). Table 3-6 illustrates the formation of three cells for period one.

Table 3-6: Machine-part matrix for period 1 (MPCFP model)

Cell	Machine (QTY)	Part Families															
		17	9	1	2	5	6	8	10	12	13	14	16	18	20	21	23
1	E(1)	1															
	F(1)	1															
	K(1)	*															
2	F(1)		1														
	J(2)		1														
	K(1)		1														
3	A(1)			1							1			1			
	B(1)					1											1
	C(1)								1		1						
	D(1)							1					1		1		
	E(1)				1	1	1				1						
	F(1)													1			
	G(1)									1		1				1	1
	H(1)				1		1					1				1	
	I(1)			1		1		1		1					1		
	J(1)			1			1					1	1	1			
	K(1)							1	1						1		

Second: To accommodate the demands required for period two, cell one requires three machine types (E, F and K) to process part type (17), while cell two requires three machine types (F, J and K) to process part type (9). On the other hand, cell three requires eleven machine types (A, B, C, D, E, F, G, H, I, J and K) to process a part family of fourteen part types (1, 2, 3, 4, 5, 6, 7, 8, 10, 12,13, 14, 16, 18, 19, 20, 21, 22 and 23). Table 3-7 illustrates the formation of three cells for period two.

Table 3-7: Machine-part matrix for period 2 (MPCFP model)

Cell	Machine (QTY)	Part Families																				
		17	9	1	2	3	4	5	6	7	8	10	12	13	14	16	18	19	20	21	22	23
1	E(1)	1																				
	F(1)	1																				
	K(1)	*																				
2	F(1)		1																			
	J(2)		1																			
	K(1)		1																			
3	A(1)			1		1								1			1					
	B(1)					1		1													1	
	C(1)						1					1		1				1				
	D(1)										1					1			1			
	E(1)				1			1	1	1				1				1				
	F(1)						1			1							1	1				1
	G(1)												1		1					1		
	H(1)				1				1						1					1		
	I(1)			1				1			1		1						1			1
	J(1)			1			1		1	1					1	1	1				1	1
	K(1)					1					1	1							1		1	

Third: To accommodate the demands required for period three, cell one requires three machine types (E, F and K) to process part type (17), while cell two requires three machine types (F, J and K) to process part type (9). On the other hand, cell three requires eleven machine types (A, B, C, D, E, F, G, H, I, J and K) to process a part family of fourteen part types (1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 15, 16, 18, 19, 20, 21, 22, 23, 24 and 25). Table 3-8 illustrates the formation of three cells for period three.

Table 3-8: Machine-part matrix for period 3 (MPCFP model)

Cell	Machine (QTY)	Part Families																						
		17	9	1	2	3	4	5	7	8	10	11	12	13	15	16	18	19	20	21	22	23	24	25
1	E(1)	1																						
	F(1)	1																						
	K(1)	*																						
2	F(1)		1																					
	J(2)		1																					
	K(1)		1																					
3	A(1)			1		1						1		1			1							
	B(1)					1		1													1		1	1
	C(1)						1				1	1		1	1			1						
	D(1)									1		1			1	1			1					
	E(1)				1			1	1					1				1						
	F(1)						1		1								1	1				1		1
	G(1)												1							1			1	1
	H(1)				1															1				
	I(1)			1				1		1			1						1			1		
	J(1)			1			1		1							1	1				1	1		
	K(1)					1				1	1								1		1			

3.7.1.4 Experimental Results and Comparison

Even though part demands are not the same in all periods, no change occurs in part family/machine cell formations for cells 1 and 2 in all periods, as exhibited in Tables 3-6, 3-7, and 3-8. This means that once cells 1 and 2 are established for period one, they remain unchanged for the other two periods. However, in cell 3, the part families change from period to period, but the machine groups are the same across all periods.

In all periods, cell 1 has one machine of types E, F and K. For instance, in period 1, cell 1 has three machines: the processing steps for part type (17) are restricted to machine types E and F in every period, while machine type K remains idle. In reality, cell 1 should not have machine K. However, this occurs as a result of the machine limit constraint in Wicks and Reasor's illustrative example, where the lower limit in each cell is three machines. Our developed model adjusts this constraint through the acquisition and use of the cheapest machine.

Table 3-9: Comparison between Wicks and Reasor and MPCFP model results

		Wicks &Reasor ($)	MPCFP model ($)
Machine acquisition cost	Period 1	94,000	90,000
	Period 2	18,000	0
	Period 3	0	0
	Sub-total	**112,000**	**90,000**
Material handling cost	Period 1	8,000	0
	Period 2	3,500	0
	Period 3	1,000	0
	Sub-total	**12,500**	**0**
Machine relocation cost	All Periods	0	0
Total system cost	Period 1	102,000	90,000
	Period 2	21,500	0
	Period 3	1,000	0
	Total	**124,500**	**90,000**

Table 3-9 shows the comparison between the solution obtained by our model and that obtained by Wicks and Reasor:

- While the material handling costs in Wicks and Reasor for the three periods are $8,000 for period one, $3,500 for period two, and $1,000 for period three, with a total cost of $12,500 for all periods, there is no cost associated with material handling in the solution.
- Each part is manufactured in one cell, and thus the inter-cell cost transfer (material handling between cells) in every period is zero.

By implementing our model, the machine acquisition cost for the first and subsequent periods is $90,000. This means that the number of machines is constant in all periods; once purchased, the machines in period one remain for the subsequent periods. In contrast, the cost of the machines purchased in the Wicks and Reasor model for the first period is $94,000; then, in period two, two machines (E and F) are added to cell 1 at an extra purchasing cost of $8,000, and two machines (B and K) are added to cell 2 at an extra purchasing cost of $10,000. The sum of these costs is $112,000.

3.7.1.5 Limitations of the Wicks and Reasor Model

Despite addressing the dynamic nature of the production environment by considering a multi-period forecast of product demand for cell formation, the Wicks and Reasor approach suffers from the following limitations:

- The model is non-linear (quadratic and cubic terms) and has integer variables, making it hard to solve. The authors propose a genetic algorithm to solve the problem.
- The model assumes that each part has only one machine type sequence, which makes it very restrictive. With the choice of technologies in modern manufacturing, it may be possible to use a 5-axis CNC milling center to machine a part as one possible sequence. On the other hand, the part may also be machined using a routing through conventional machines such as a lathe, a milling machine, and a drilling machine. It is important to be able to model the inherent trade-offs resulting from alternate routings, i.e., the trade-off between the costs of high technology equipment with simpler routings versus lower technology equipment costs with complex routings.
- The model implicitly assumes growth in demand. The authors have a positive variable b_{jl} to represent the number of machines of type j acquired in period l. However, with negative demand scenarios, it should be possible to discard machines.

3.7.1.6 Benefits of the MPCFP Model in Comparison to Wicks & Reasor

As stated earlier, the (Wicks & Reasor, 1999) is considered one of the main reference models in the facilities design literature. Therefore, a comparison between our proposed model and the Wicks and Reasor model is conducted to understand the benefits of our model. In brief, these benefits are as follows:

- The objective function in our model is to some extent similar to that of Wicks and Reasor, the only exception being that the Wicks and Reasor model does not handle the cost of discarding machines.
- Our model simplifies the flow network structure inherent but not clearly recognized in Wicks and Reasor.
- The advantage of our model is that it can be solved optimally using standard commercial or open-source integer-linear programming packages.

3.7.2 Second Illustrative Example

(Jayakumar & Raju, 2010) present a case study problem to illustrate the applicability and formulation of the MPCFP in a dynamic production environment. The model has the advantage of forming machine cells and part families simultaneously. The case study was conducted in a valve manufacturing company that faced problems with a job shop or functional layout. Their aim was to convert the job shop layout into CMS layout to overcome typical problems associated with the job shop layout.

In the case study, they consider 8 different type of machines (index *j* in the model), 12 part types (index *i* in the model), and three time periods (index *t* in the model). Resource data and part type attributes for the three time periods are respectively given in Tables 3-10 and 3-11.

Table 3-10: Resource data for the Jayakumar and Raju case study

Machine Type		Acquisition Cost($)	Operation Cost per Hour ($)	Relocation Cost ($)	Capacity (time unit/period) (hrs)
1	A	1900	6	900	600
2	B	1300	8	700	600
3	C	1400	7	750	600
4	D	1800	6	700	600
5	E	1500	6	700	600
6	F	1400	7	650	600
7	G	1300	8	550	600
8	H	1500	9	750	600

Table 3-11: Data of part types for the Jayakumar and Raju case study problem

Part Type	Intra cell Material Handling Cost/batch	Inter cell Material Handling Cost/batch	Operation	Operation Sequence	Time	DEMAND		
						Period 1	Period 2	Period 3
1	6	40	1	7,8	0.5,0.33	400	650	0
			2	5,7	0.94,0.24			
			3	5,8	0.81,0.42			
2	6	40	1	2,5	0.28,0.86	650	0	400
			2	2,5	0.44,0.76			
			3	3,4	0.97,0.47			
3	6	40	1	2,5	0.19,0.74	0	450	0
			2	4,6	0.49,0.45			
			3	5,7	0.65,0.59			
4	6	40	1	1,4	0.44,0.71	750	500	600
			2	4,8	0.36,0.49			
			3	1,7	0.67,0.51			
5	6	40	1	3,6	0.48,0.23	550	0	750
			2	3.7	0.57,0.24			
			3	2,6	0.67,0.77			
6	6	40	1	2,3	0.61,0.63	0	500	350
			2	2,5	0.68,0.55			
			3	3,4	0.88,0.63			
7	6	40	1	1,4	0.89,0.24	450	0	300
			2	1,6	0.81,0.34			
			3	2,6	0.51,0.71			
8	6	40	1	5,6	0.58,0.96	650	0	350
			2	5,7	0.13,0.36			
			3	3,6	0.19,0.89			
9	6	40	1	4,8	0.45,0.67	750	350	0
			2	3,6	0.76,0.61			
			3	4,5	0.35,0.78			
10	6	40	1	3,7	0.78,0.81	900	450	700
			2	3,6	0.47,0.12			
			3	2,6	0.57,0.48			
11	6	40	1	6,8	0.39,0.44	0	0	700
			2	7,8	0.48,0.72			
			3	7,8	0.36,0.66			
12	6	40	1	1,7	0.49,0.67	350	600	0
			2	1,4	0.72,0.59			
			3	1,8	0.33,0.48			

The operating cost, machine capacities, and acquisition and relocation costs are assumed to be constant over the three time periods. The inter- and intra- cell movement costs are constant for all moves (regardless of the distance traveled), and their values are $40 and $6, respectively. The following cell capacity constraints were placed on the design of the CMS.

- The machines are to be grouped into three relatively independent cells and reconfiguration is to be performed at the beginning of the second period to respond to changes in production demand.
- The upper limit on the number of machines in each cell is five. However, the lower limit of machines in each cell is not given, and the authors state that smaller cells are preferable.
- Each operation of a part can be performed on two alternative machines, and each part type has three operations and is processed using one of the alternatives.

3.7.2.1 Cell Formation for Each Period (Jayakumar and Raju)

The authors present a multi-objective non-linear programming model to solve the MPCFP. Since the model is a non-linear mixed integer program with absolute terms in the third and fourth terms of the objective function, it cannot be solved using ILP software. The authors use linearization to transform the absolute value terms into corresponding linear terms (with the introduction of a non-negative variable).

The solutions to the example problem obtained via the multi-period are presented in Tables 3-12, 3-13 and 3-14 using a machine-part incidence matrix. These tables illustrate the recommended (Jayakumar & Raju) multi-period design for each period.

First Period: To accommodate the demands required for period one, cell one requires five machines, two units of machine type M3 and one unit of machine types (M6, M7 and M8) to process part family (PF1) of three part types (P1, P7 and P10). Cell two requires five machines, two units of machine type (M1) and one unit of machine types (M2, M4 and M7) to process part family (PF2) of three part types (P2, P4 and P12). On the other hand, cell three requires five machines, two units of machine type (M4) and one unit of machine types (M2, M5 and M6) to process part family (PF3) of three part types (P5, P8 and P9). Table 3-12 illustrates the formation of the three cells for period one. The numbers inside the table indicate which machine/cell combination is required for which operation on the part. For example, in the case of P1, the first operation is done on M8 in cell 1, the second operation on M7, also in cell 1, and the third operation back on M8 in cell 1.

Table 3-12: Machine-part matrix for period 1 (Jayakumar and Raju model)

Cell	Machine (QTY)	Part Families								
		P1	P7	P10	P2	P4	P12	P5	P8	P9
1	M3 (2)			1					3*	
	M6 (1)		2,3	2,3						
	M7 (1)	2								
	M8 (1)	1,3								
2	M1 (2)					1	1,2,3			
	M2 (1)				1,2					
	M4 (1)				3	2				
	M7 (1)					3	1	2*		
3	M2 (1)							3		
	M4 (2)		1*							1,3
	M5 (1)								1,2	
	M6 (1)							1		2

Second Period: To accommodate the demands required for period two, cell one requires a group of three machines types (M3, M7 and M8) to process part family (PF1) of two part types (P1 and P10). Cell two requires three machines, two units of machine type (M1) and one unit of machine type (M4) to process part family (PF2) of two part types (P4 and P12). On the other hand, cell three requires four machines, one unit of machine types (M2, M4, M5 and M6) to process part family (PF3) of three part types (P3, P6 and P9). Table 3-13 illustrates the formation of three cells for period two.

Table 3-13: Machine-part matrix for period 2 (Jayakumar and Raju model)

Cell	Machine (QTY)	Part Families						
		P1	P10	P4	P12	P3	P6	P9
1	M3 (1)		1,2					
	M7 (1)	2						
	M8 (1)	1,3						
2	M1 (2)			1,3	1,3			
	M4(1)			2	2			
3	M2 (1)					1	1	
	M4 (1)						3	1,3
	M5 (1)					3	2	
	M6 (1)		3*			2		2

Third Period: To accommodate the demands required for period three, cell one requires a group of four machine types (M2, M3, M6 and M7) to process part family (PF1) of two part types (P5 and P10). Cell two requires three machines, two units of machine type (M4) and one unit of machine type (M7) to process part family (PF2) of part type (P4). On the other hand, cell three requires five machines, one unit of machine types (M2, M4, M5, M6 and M7) to process part family (PF3) of five part types (P2, P6, P7, P8 and P11). Table 3-14 illustrates the formation of three cells for period three.

Table 3-14: Final solution for period 3 (Jayakumar and Raju model)

Cell	Machine Type (No. of M/c)	Part Families							
		P5	P10	P4	P2	P6	P7	P8	P11
1	M2 (1)	3							
	M3 (1)		1					3*	
	M6 (1)	1	2						
	M7 (1)	2	3						
2	M4(2)			1,2					
	M7(1)			3					
3	M2 (1)				1,2	1			
	M4 (1)				3	3	1		
	M5 (1)					2		1,2	
	M6 (1)						2,3		1
	M7 (1)								2,3

3.7.2.2 Implementing the MPCFP Model to Solve the Jayakumar and Raju Case Study

The data presented in the (Jayakumar & Raju) case study was used to test our proposed model. The developed code in Python is used to generate all possible sequences for each part routing through machines and cells, using exactly the same data as in their case study. Each part in the Jayakumar and Raju case study has alternative routings, whereby each operation can be fulfilled by one of two defined machines. This results in hundreds of routing sequences in an attempt to find the best processing routes for each part. We are also able to determine the inter-cell material handling cost for each sequence. Since the MPCFP model can accept any number of alternate routings for a part, the restrictive assumption in the Jayakumar and Raju data with two alternatives per operation is easily handled.

As a final step, the sequences are entered as input to the optimization model in CPLEX Studio (running CPLEX in the background) with the Jayakumar & Raju data. The CPLEX Studio running on a 64-bit machine with an Intel i5 chipset running at 3.20 GHz, is able to get to an optimal solution in 7.94 seconds. The authors report their best solutions after run times of 8 hours using Lingo 11.0.

3.7.2.3 Cell Formation for Each Period

In this section, we discuss the results that emerged from solving the case study using our MPCFP model. The recommended solutions are presented in Tables 3-15, 3-16 and 3-17. First Period: To accommodate the demands required for period one, cell one requires five machine types (M2, M3, M4, M5 and M6) to process part family (PF1) with five part types (P2, P7, P8, P9 and P10), while cell two requires four machine types (M2, M4, M6 and M7) to process part family (PF2) with four part types (P2, P5, P9 and P10). In contrast, cell three requires four machine types (M1, M4, M7 and M8) to process part family (PF3) with three part types (P1, P4 and P12). Table 3-15 illustrates the formation of the three cells for period one.

Table 3-15: Cell formation for period 1 (MPCFP model)

Cell 1	Part family (PF1)	Parts P2, P7, P8, P9, P10
	Machine group 1	Machines M2, M3, M4, M5, M6
Cell 2	Part family (PF2)	Parts P2, P5, P9, P10
	Machine group 2	Machines M2, M4, M6, M7
Cell 3	Part family (PF3)	Parts P1, P4, P12
	Machine group 3	Machines M1, M4, M7, M8

Second: To accommodate the demands required for period two, cell one requires five machine types (M2, M3, M4, M5 and M6) to process part family (PF1) with three part types (P3, P6 and P10), while cell two requires four machine types (M2, M4, M6, and M8) to process part family (PF2) with only one part type (P9). On the other hand, cell three requires four machines types (M1, M4, M7 and M8) to process part family (PF3) with three part types (P1, P4 and P12). Table 3-16 below illustrates the formation of three cells for period two.

Table 3-16: Cell formation for period 2 (MPCFP model)

Cell 1	Part family (PF1)	Parts P3, P6, P10
	Machine group 1	Machines M2, M3, M4, M5, M6
Cell 2	Part family (PF2)	Part P9
	Machine group 2	Machines M2, M4, M6, M7
Cell 3	Part family (PF3)	Part P1, P4, P12
	Machine group 3	Machines M1, M4, M7, M8

Third: To accommodate the demands required in period three, cell one requires five machine types (M2, M3, M4, M5 and M6) to process part family (PF1) with five part types (P2, P6, P7, P8 and P10), while cell two requires four machine types (M2, M4, M6, and M7) to process part family (PF2) with two part types (P7 and P11). On the other hand, cell three requires three machines of types (M1, M4 and M7) to process part family (PF3) of with part types (P4 and P11). Table 3-17 illustrates the formation of three cells for period three.

Table 3-17: Cell formation for period 3 (MPCFP model)

Cell 1	Part family (PF1)	Parts P2, P6, P7, P8, P10
	Machine group 1	Machines M2, M3, M4, M5, M6
Cell 2	Part family (PF2)	Parts P7, P11
	Machine group 2	Machines M2, M4, M6, M7
Cell 3	Part family (PF3)	Parts P4, P11
	Machine group 3	Machines M1, M4, M7

3.7.2.4 Experimental Results and Comparison

Even though part demand is not similar in all periods, no change occurs in machine cell formations for cells 1, 2 and 3 in periods 1 and 2. In period 3, the cell formation is the same as in previous periods except that machine type M8 is disposed of from cell 3, as exhibited in Tables 3-15, 3-16 and 3-17. This means that once cells 1, 2 and 3 are established for period one, they remain for the subsequent periods without any incurring extra machine acquisition cost. However, in cells 1, 2 and 3, the part families change from period to period, whereas the machine groups remain the same for all periods.

Table 3-18: Comparison between (Jayakumar & Raju) model and MPCFP model results

	Jayakumar & Raju ($)	**MPCFP Model ($)**
Machine acquisition cost	~~56,400~~	19,700
Actual Cost	28,300	-
Machine disposal cost	0	0
Machine relocation cost	4,525	0
Material handling cost	20,114.29	25,950.4
Production cost	118,356	116,518.4
Total system cost	~~199,395.29~~	162,168.8
Actual system cost	**171,295.29**	162,168.8

Table 3-18 illustrates a comparison and cost analysis between the solution obtained by our proposed model and the solution obtained by the Jayakumar and Raju paper:

- By implementing our model, the machine acquisition cost for the first period and subsequent periods is $19,700. This means that the number of machines is constant in all periods, and that the machines purchased in period one remain for the subsequent periods. In contrast, the machine acquisition costs in Jayakumar and Raju for the three consecutive periods are $23,000, $15,800 and $17,600, for a total of $56,400 for all three periods. However, we believe that there is no machine acquisition cost associated with period two in their case as they used the same machines being used in period one. In period three, they added 4 machine types

(M2. M6, and M7(2)) with extra cost of $5300, this make their total machine acquisition cost equal to $28,300.

- In the Jayakumar and Raju model, the material handling cost (including inter-cell and intra-cell) for the three periods is $20,114.29. In contrast, the material handling cost in our developed model for the three periods, including inter-cell and intra-cell costs, is $25,950.4. The extra inter-cell material handling cost in our solution comes as a result of the extensive transfer between cells, as opposed to increasing or changing the number of machines in the cells.

- The machine-relocating cost in our model is $0. While there is a cost associated with relocating machines in Jayakumar and Raju, in period two, two machines (M3 and M6) are discarded in cell 1, with an extra removing cost of $700. Two machines (M2 and M7) are discarded in cell 2, with an extra removing cost of $625. One machine (M4) is discarded in cell 3, with an extra cost of $350. In period three, two machines (M2 and M6) are added to cell 1, with an extra adding cost of $675, and one machine (M8) is discarded, with an extra removing cost of $375. Two machines (M4 and M7) are added to cell 2, with an extra adding cost of $625, and two machines (M1) are discarded, with an extra removing cost of $900. One machine (M7) is added in cell 3, with an extra adding cost of $275. The sum of the machine-relocation costs in Jayakumar and Raju is $4,525.

- The production cost in our model is $116,518.4, whereas the cost associated with production in Jayakumar and Raju is reported as $118,356. There are some inconsistencies in the Jayakumar and Raju solution. For example, the routings in period 1 of parts 7, 8, 9, and 10 are inconsistent with the routing data. The same is true in the results of period 2 for parts 3, 9, and 10 and in period 3 of part 10.

3.7.2.5 Limitations of the Jayakumar and Raju Model

The Jayakumar and Raju model addresses the dynamic nature of the production environment by considering a multi-period forecast of product mix and demand during the formation of the PF/CF problem. The objective of the model is to minimize the sum of the machine constant cost, the operating cost, and the inter-cell/intra-cell material handling cost for the given periods. One extreme solution is to duplicate machines to reduce inter-cell/intra-cell transfers, but this strategy results in higher acquisition costs. Another extreme solution is to purchase as few machine as possible, but this results in more material transfers and higher costs in material handling. Although the Jayakumar and Raju model tries to balance these costs, the model nonetheless suffers from the following limitations:

- The original objective function is a non-linear integer equation with absolute value terms (that take on binary values 0 or 1) in the third and fourth terms of the objective function to represent whether cell transfer has occurred between two successive operations. The authors converted the non-linear absolute value function into a linear function by transforming the absolute terms into their corresponding linear terms (with the introduction of non-negative difference variables) to solve the problem.
- The model assumes that each part has three operations and that the operation of each part can be performed on two alternative machines, but this approach is restrictive. It is important to be able to model the inherent trade-offs resulting from alternate routings, i.e., the trade-off between the costs of high technology equipment with simpler routings versus lower technology equipment costs with complex routings and extra material handling costs.

3.7.2.6 Benefits of the MPCFP Model over Jayakumar & Raju

The model in (Jayakumar & Raju, 2010) calculates the inter-/intra-cell movement and the manufacturing costs, as does our model. This paper considered a recent publication that deals with the CMS design with routing flexibility and dynamic CF. However, we are looking to solve the multi-period cell formation problem with routing flexibility using a time-phased multi-commodity flow network. Therefore, a comparison between our proposed model and the Jayakumar and Raju model is conducted to understand the benefits of our model, which are as follows:

- Although the Jayakumar and Raju model has routing flexibility, our model simplifies the inherent flow network structure and clearly recognizes the optimal routing that decreases the overall cost of the system.
- Our model can be solved in a few seconds using standard commercial or open-source integer linear programming packages. However, the Jayakumar and Raju proposed model uses Lingo to find the optimal solution for only small- and medium-sized problems. It is not suitable for large problems because the memory and computational time requirements are extremely high and increase exponentially as the problem size increases.
- The MPCFP model can handle any number of operations and alternative sequences per part. Obviously, larger the problem, the longer it will take to solve.

3.8 Summary of the MPCFP model

- The MPCFP model is conceptually powerful and has an underlying multi-commodity network flow structure.
- The MPCFP model is linear with integer variables (as opposed to having a non-linear objective function that has been converted to a linear integer equation).
- The MPCFP model assumes that each part has possible alternative processing sequences (multiple machine type sequences).

In conclusion, the proposed multi-commodity network flow-based model for the MPCFP is structured, flexible and efficient. Since the start-to-finish paths are generated before optimization, further restrictions, such as those listed below, may be imposed during the pre-processing phase:

- Do not allow more than *n* cell transfers for any product routing.
- A certain product must be processed completely within a cell.
- A certain percentage of processing for a product should occur with a single cell.

The new MPCFP model presented in this thesis has both modelling and computational advantages over existing cell formation models and should be useful to both academics and practitioners.

In the next chapter, a model extension that allows for the design of a GT layout is presented, after which a robust optimization extension for the MPCFP is proposed.

Chapter 4: Understanding the Design Continuum Between Group Technology and Fractal Cell Designs for Manufacturing Systems Through the CBCMS

The MPCFP model developed in the previous chapter can be used as a springboard to explore the design continuum between group technology and fractal cell layout organizations for manufacturing systems. Broadly speaking, the group technology system is efficient when the cellular partitions are crisply defined and the product mix and demand parameters do not vary either within a time period or from period to period. While GT is efficient for a given product mix and demand, it is inflexible and unresponsive to variations in product mix and demand. On the other hand, the fractal cell organization is very flexible because it offers a variety of alternate processing capabilities. It may or may not be efficient from a cost perspective (the literature on the fractal cell system is still quite scant), depending on the level of cell duplication required for fully balanced fractal cells and setup requirements.

In this chapter, we first discuss a layout arrangement that we call the central backup cellular manufacturing systems (CBCMS). The CBCMS organization is inspired by the concept of the remainder cell in GT systems. The remainder cell in GT is a catch-all cell to provide flexibility (Suer et al., 2010). The idea is to allocate products that are not easily partitioned in the parts-machine incidence matrix to the remainder cell.

In the CBCMS system, the central backup cell (which is, in many ways, similar to the remainder cell) is explicitly designed in tandem with the other cells in the system. The CBCMS is a combined layout between GT and fractal cells designed to accommodate variability in a changing production environment. In this research, we are interested in variability due to internal and external disturbances in manufacturing systems. We intend to design an efficient and flexible system that can provide better working conditions under uncertainty issues in cellular manufacturing. The CBCMS provides the advantages of combining together two layouts – namely, GT and fractal layouts. The GT layout represents an efficient approach, while the fractal layout represents a flexible approach.

In this chapter, we begin with an illustration of how a CBCMS is constituted from both GT and fractal cells. Then we examine cell design and configuration issues for dynamic and uncertain production environments, with an emphasis on variability or uncertainty due to external and internal disturbances. Next, we discuss the functionality of CBCMS during certainty situations and how changing demands can be managed in CBCMS. To elaborate on the opportunities for designing efficient and flexible cellular manufacturing systems under uncertainty, we analyze the efficiency and flexibility of the CBCMS layout in comparison to GT and fractal layouts. Then we identify the capabilities of the CBCMS and present the concept of an efficient and flexible spectrum in the CBCMS. Finally, some results are presented to highlight the robustness of the CBCMS layout.

4.1 Framework for Central Backup Cellular Manufacturing Systems

The general layout arrangement of CBCMS, which is the combined layout between GT and fractal layouts, is illustrated in Figure 4-1. Here, all cells in the layout arrangement are conventional GT cells, with the exception of the cell in the center (which may be seen as a fractal cell). For example, cells 1 to 4 and 6 to 9 are regular GT cells to manufacture different types of products based on similar manufacturing operations or design attributes. Cell number 5, located in the center of the layout, is a central backup cell that includes all process and is designed to accommodate all part families manufactured in the CBCMS facility.

It is believed that a central backup cell will be sufficiently responsive and flexible to deal with abnormalities during production, while GT cells will be able to deal with scheduled production tasks. The manufacturing cells in a CBCMS layout can be made focused by using special purpose machines such as conventional lathe and milling machines, and the production support for these cells may be equipped with conventional material-handling systems. However, the manufacturing cells could be made flexible by using multi-purpose machines such as CNC lathe and milling machines, and the production support for these cells could be equipped with mechanized systems for material handling and industrial robots.

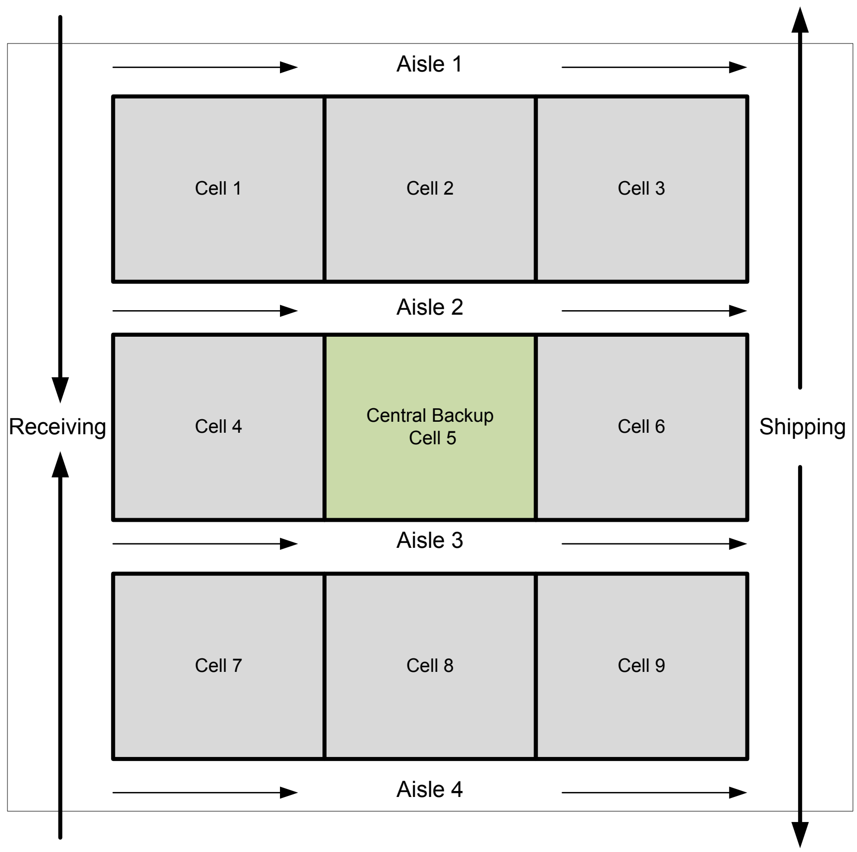


Figure 4-1: General schematic layout of central backup cellular manufacturing systems

A flexible machine is designed for a very rapid changeover, whereas a focused machine is designed to produce similar parts or products frequently. A flexible machine has higher costs compared to a focused machine, but a focused machine has higher efficiency compared to a flexible machine.

Layout design is a challenging task for facilities planners because there are many trade-offs that need to be considered. For example, by purchasing CNC machines (flexible machines), we may reduce the flow and cost of material handling, but operation and investment costs will increase because CNC machines are more expensive to purchase and operate. On the other hand, by purchasing conventional machines such as lathe or milling (focused machines), we may reduce investment and operation costs, but the flow and the cost of material handling will increase because more parts routing is needed. These kinds of trade-offs are implicit in the MPCFP model presented in the previous chapter.

In real-world business, industrial companies have different strategies to achieve their objectives. These strategies, along with other factors, usually form the companies' business models. The factors include, but are not limited to: 1) the level of competition in the market, 2) the available resources within the company, 3) product mix variability, and 4) changing demand. Based on these factors, manufacturers may need to choose between focused and flexible cells in their facility to attain their objectives.

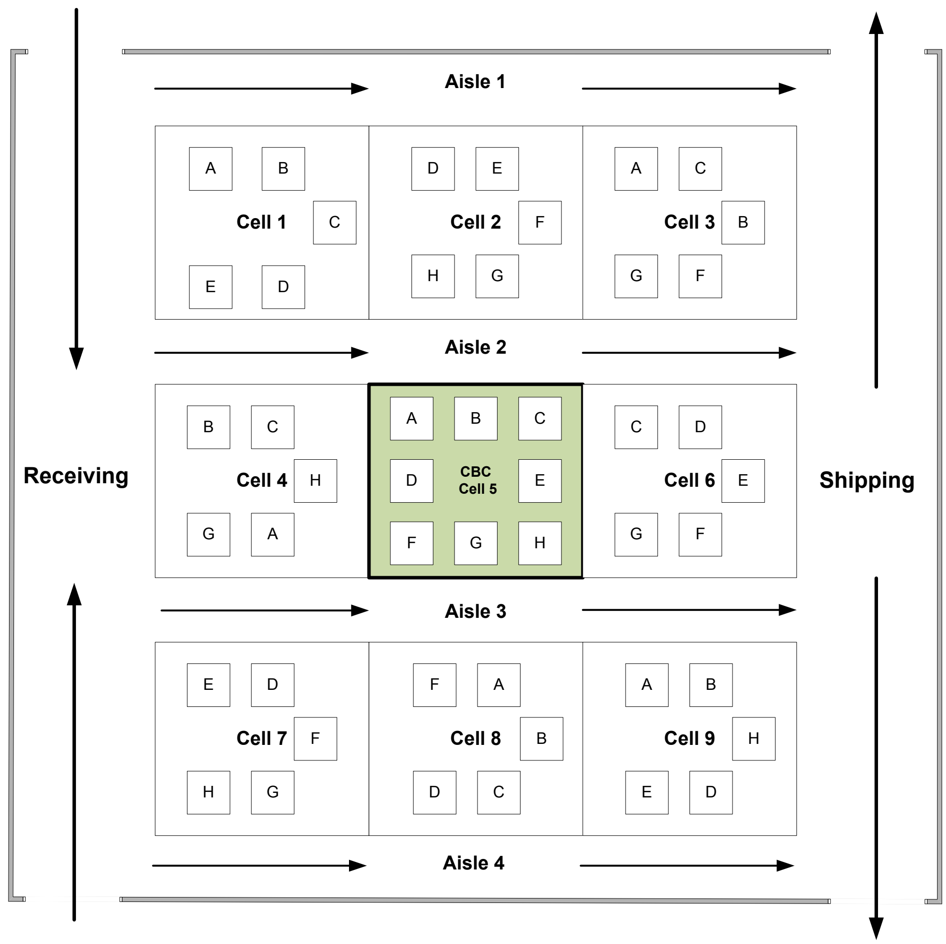


Figure 4-2: The location of the central backup cell within the proposed CBCMS layout

It should also be noted that in the CBCM system, the number of fractal cells is, in itself, a design decision. Although the illustration in Figure 4-1 shows only one fractal cell, many more are possible. In the CBCMS layout, the machines are arranged in the cells based on the concept of GT and fractal cells to provide an efficient and flexible system for manufacturing in a changing production environment. For example, cell 1 contains a group of special-purpose machines (A, B, C, D and E), as illustrated in Figure 4-2, which are assigned to manufacture Part Family 1. Hence, cell 1 is dedicated to Part Family 1, cell 2 is dedicated to Part Family 2, and so on. Meanwhile the central backup cell (cell 5) is designed to be a flexible cell that may contain a group of special-purpose machines that are capable of manufacturing all products. For instance, the central backup cell consists of group of special-purpose machines (A, B, C, D, E, F, G and H) that represent all types of machines in the facility.

We believe that the CBCMS arrangement is both efficient and responsive when performing a wide range of operations.

4.2 Variability Within the Context of CBCMS

Variability exists in all production systems and can have a significant impact on performance. Therefore, the ability to measure, understand and manage variability is crucial to effective manufacturing management (Hopp & Spearman, 2008). In this research, variability is classified into two types: 1) variability as a result of internal interruptions (or, as a consequence of events related to our activities and decisions; and 2) variability as a result of external forces that are beyond our immediate control. In this research, the term "uncertainty situation" is used interchangeably with the word "variability".

We are interested in investigating how variability can be managed in the context of central backup cellular manufacturing systems. Central backup cellular manufacturing systems can be used effectively during variability resulting from internal interruptions and external interruptions. Examples of variability resulting from internal interruptions include: equipment breakdowns, queuing delays, reject and rework, and variable process time. In contrast, variability resulting from external interruptions includes product mix variability, limited delivery time, and fluctuating demand. The focus at this stage is on how a CBCMS layout behaves in the face of variability.

4.2.1 Managing Variability Resulting from Internal Interruptions

In order to comprehend the various challenges mentioned above, it is important to elaborate on variability that results from internal interruptions. There are four main instances of this type of variability. First of all, equipment breakdowns (scheduled and unscheduled downtime) can greatly affect the production system. Scheduled downtime can be managed relatively easily because all affected activities are known beforehand. However, unscheduled downtime may occur suddenly (i.e., during a machine's performance of a job) and thus can greatly affect the flow of product within the facility. It is also important to note that frequent breakdowns are expected when the facility is used over a period of time due to the deterioration of machines. A second internal disruption is queuing delays (congestion caused by WIP) near a machine or workstation that can result in delayed tasks. A third instance of internal disruption may occur in the case of reject and reworks, when quality problems may cause some tasks to be repeated and some products to be scrapped. If a task was done incorrectly during production, rework may be conducted on the same part to correct the problem. On the other hand, if a part is completely scrapped, repeating the task from the beginning is necessary. A fourth example of internal disruption arises in the case of variable process time, where the process time differs from product to product. The processing time for a product may also vary due to differences in operator skills and machine capabilities.

The central backup cell in CBCMS layout, as illustrated in Figure 4-3, is capable of dealing with any of the uncertainty situations resulting from internal interruptions. The jobs that cannot be performed in the designated cells could be transferred and processed in the central backup cell. Since all of these interruptions might not happen at the same time, manufacturers will be assured that there is an opportunity to finish the jobs more or less on time. While this is happening, manufacturers can follow the required procedures to repair the broken equipment in the GT cells. Also, manufacturers will be able to investigate and cover the proper solutions to issues related to in-process inventory and quality issues.

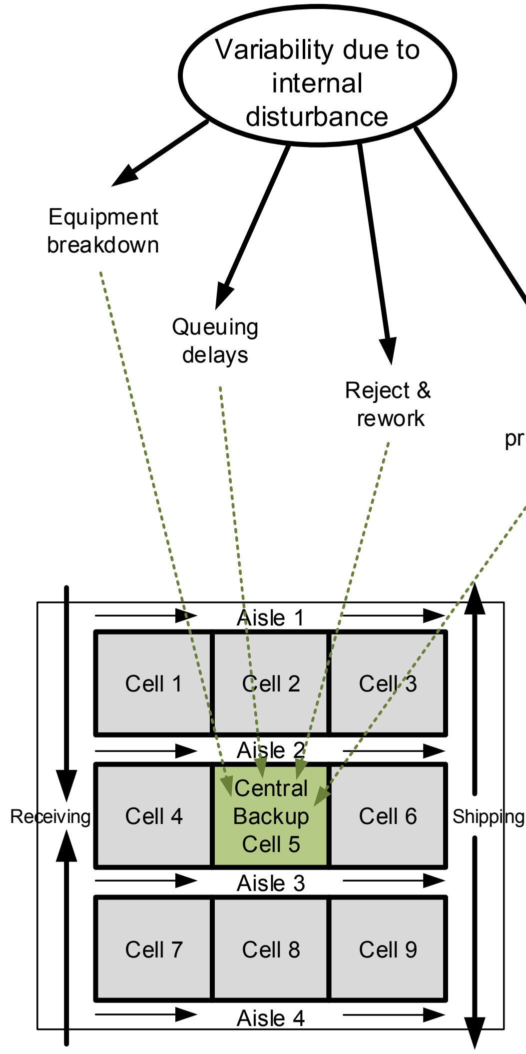


Figure 4-3: The CBCMS layout accommodates variability resulting from internal interruptions

4.2.2 Managing Variability Resulting from External Interruptions

It is important to elaborate on the variability resulting from external forces:

1. Product mix variability: in today's business environment, product life cycles are short, resulting in the regular introduction of new products or modifications of existing products. As a result, a broader or different product mix may be handled and manufactured in the facility.

2. Limited delivery time: customers prefer to customize their products and at the same time demand shorter delivery times.

3. Varying demands: demand for certain products may vary in response to the business environment.

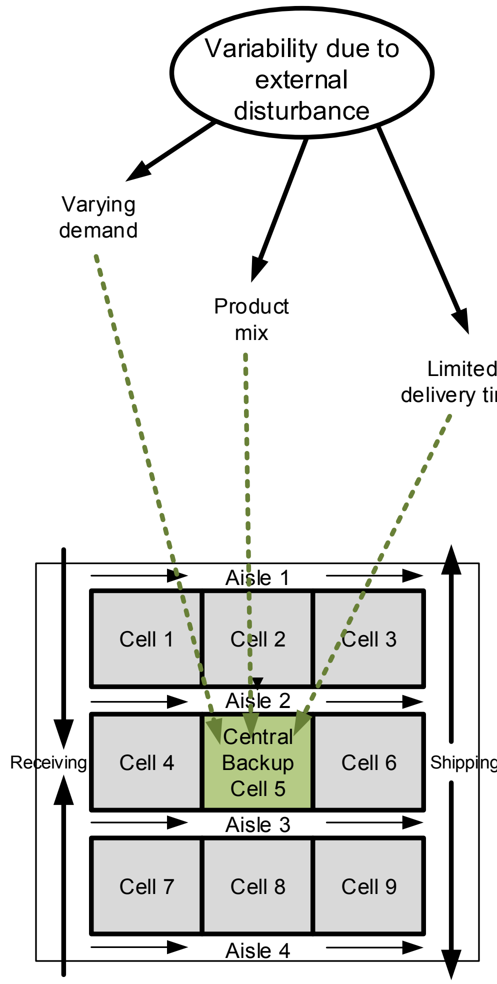


Figure 4-4: CBCMS layout accommodates variability resulting from external interruptions

The central backup cell is capable of dealing with situations of uncertainty resulting from external forces, as illustrated in Figure 4-4. For example, when there is a need for more product variety in smaller quantities, all extra jobs that cannot be done in the designated GT cells can be transferred and completed in the central backup cell. In the case of a product that has a limited delivery time, the central backup cell can be used when GT cells are busy or cannot perform the tasks on time. The ability to cope with varying demand can also be incorporated into the CBCMS design. This will be discussed in greater detail later in the thesis.

From the business environment prospective, the product-price and/or lead time is subject to change for any number of reasons at any given time. It is known that changes in the product-price and/or lead time may result in increased competition among competitors. Research in the MPCFP has ignored these considerations because they lead to more complexity and make it more difficult to come up with a design framework.

Although the proposed models in this research focus on minimizing the cost of manufacturing products, adding the product-price to such a model may require different business and management models to solve the problem. Therefore, the product-price and lead time issues will not be addressed in this research. However, they should certainly be considered issues for further research.

4.3 Other Aspects of CBCMS

Figure 4-5 describes other functional aspects of the CBCMS layout. The situations that we will discuss here are: providing training programs to machine operators, making prototypes, and using the central backup cell to expand the layout (i.e., initiating the first expansion cell). The central backup cell may handle some planned activities within the facility. For example, if training programs are required for operators in the facility, training may be performed in the central backup cell. As described earlier, the central backup cell contains different types of focused machines similar to machines that are available in the GT cells within the facility. In such cases, the trainees may use the available machines in the central backup cell during training sessions instead of using machines in GT cells. GT cells are dedicated for scheduled production.

The central backup cell can also be used for making prototypes whereby machines are required to create a physical model of a component or product. Therefore, the available machines in the central backup cell 5 can be used to develop the required prototypes instead of using machines in GT cells.

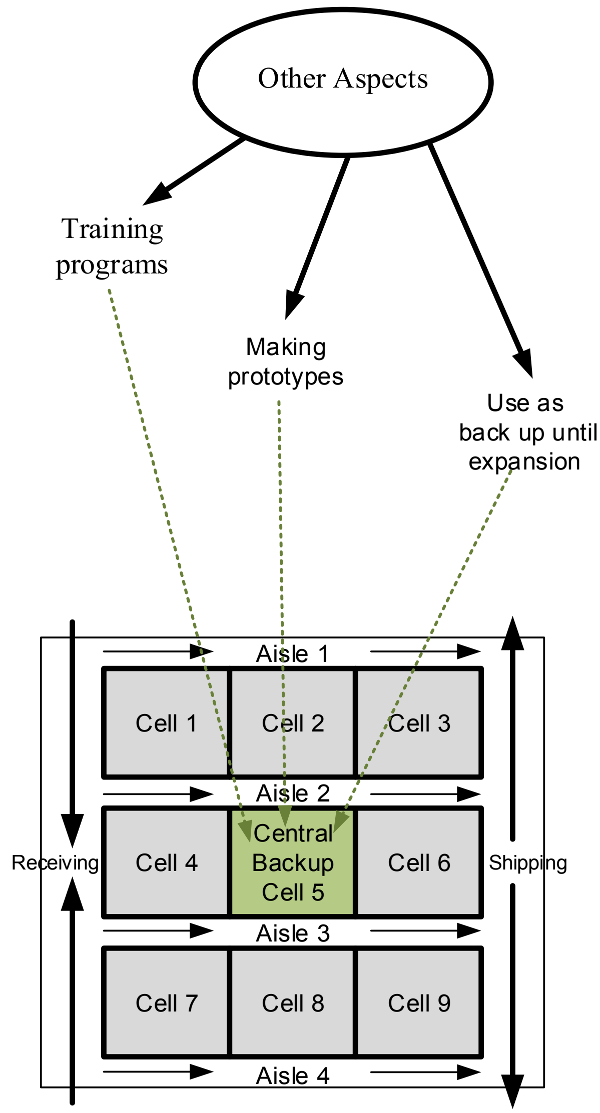


Figure 4-5: Using CBCMS layout for other situations

At some point, a manufacturing facility's capacity may be saturated, at which time expanding the existing facility is an option to enhance production. If this is the case, the central backup cell can be used temporarily until the designer recommends a complete plan for facility expansion.

4.4 Managing Demand Variability in the CBCMS Layout

Before discussing how to manage demand variability in the CBCMS layout, it is important to understand what is meant by demand in this study. Demand variability is considered one of the factors that affect layout design decisions. These days, researchers and facilities designers are interested in the issue of changing demand when designing a new or modifying an existing layout facility. In this section, we attempt to understand demand variability and look for ways to mitigate the effect of uncertain demand in the context of using CBCMS.

Demand is the quantity of manufactured goods consumers are willing and able to purchase at a given price over a particular period of time. In this research, demand is classified into three main categories based on a fast changing business environment: a) steady demand, b) seasonal demand, and c) varying demand. Steady demand is a relatively stable demand for products and usually has a range of definite quantities. Seasonal demand reflects a manufacturer's interest in manufacturing particular products only during a specific period during the calendar year. On the other hand, varying demand occurs when demand rises or falls suddenly in response to product technology, changing economic conditions, or customer spending patterns.

Another key aspect of demand variability is whether it is short-term or long-term. Also, there may be clear trends in demand as new products are designed and introduced to a manufacturing facility while others become obsolete.

As shown in Figure 4-6, GT cells are responsible for handling relatively stable demand, while the central backup cell is responsible for handling fluctuating and seasonal demand. For example, GT cells in the CBCMS layout may accommodate the entire steady demand of all parts. In contrast, the central backup cell may accommodate the excess in demand and a portion of the seasonal demand when GT cells are working at full capacity. We may also note that both GT cells and the central backup cell can be used for seasonal demand. This depends on the volume of the seasonal demand during each calendar year. In a fast-changing business environment, demand may go up or down sharply, as mentioned earlier.

Therefore, the central backup cell will be used to manage the excess demand. While we cannot escape variation in demand, we can consider the central backup cell as an effective means to cope with both short- and long-term demand variability.

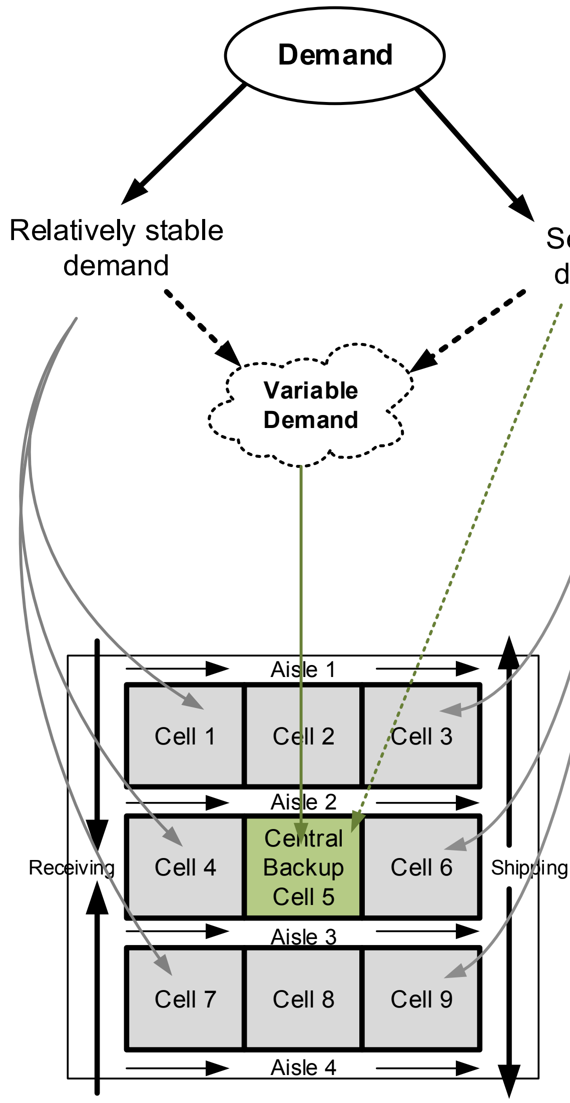


Figure 4-6: Handling demand variability in CBCMS layout

Finally, exploring how to manage demand variability in the CBCMS layout is an interesting task that necessarily includes understanding demand variability and the influence of changing demand on manufacturing companies. We believe that more research is required to comprehend the related issues in demand variability with respect to cellular manufacturing system and CBCMS.

4.5 An Approach for Implementing CBCMS Layout in a Changing Environment

The workflow diagram below provides a general overview of the flow of tasks within the CBCMS layout in a dynamic manufacturing environment. The flow chart shows the process of manufacturing a product using the CBCMS layout, starting from the work order.

Figure 4-7 presents a simplified flowchart for implementing the central backup cellular manufacturing system layout in a changing environment. The work order includes tasks for manufacturing a product family. The work order may also include other tasks, such as providing training programs to machine operators, creating product prototypes, and initiating the first cell expansion as indicated earlier. The GT cells in the CBCMS layout may be used to carry out the tasks indicated in the work order. Therefore, the job may be assigned to a specific GT cell that is dedicated to manufacturing the product family. If the GT cells cannot handle the task, then work has to be taken to the central backup cell.

Effective implementation of the central backup cell requires a consideration not only of processes and technologies, but also of organizational and human issues. Using the central backup cell in manufacturing improves the facility to respond to abnormalities more quickly and to ensure top operational practice to maintain a competitive position in the global market.

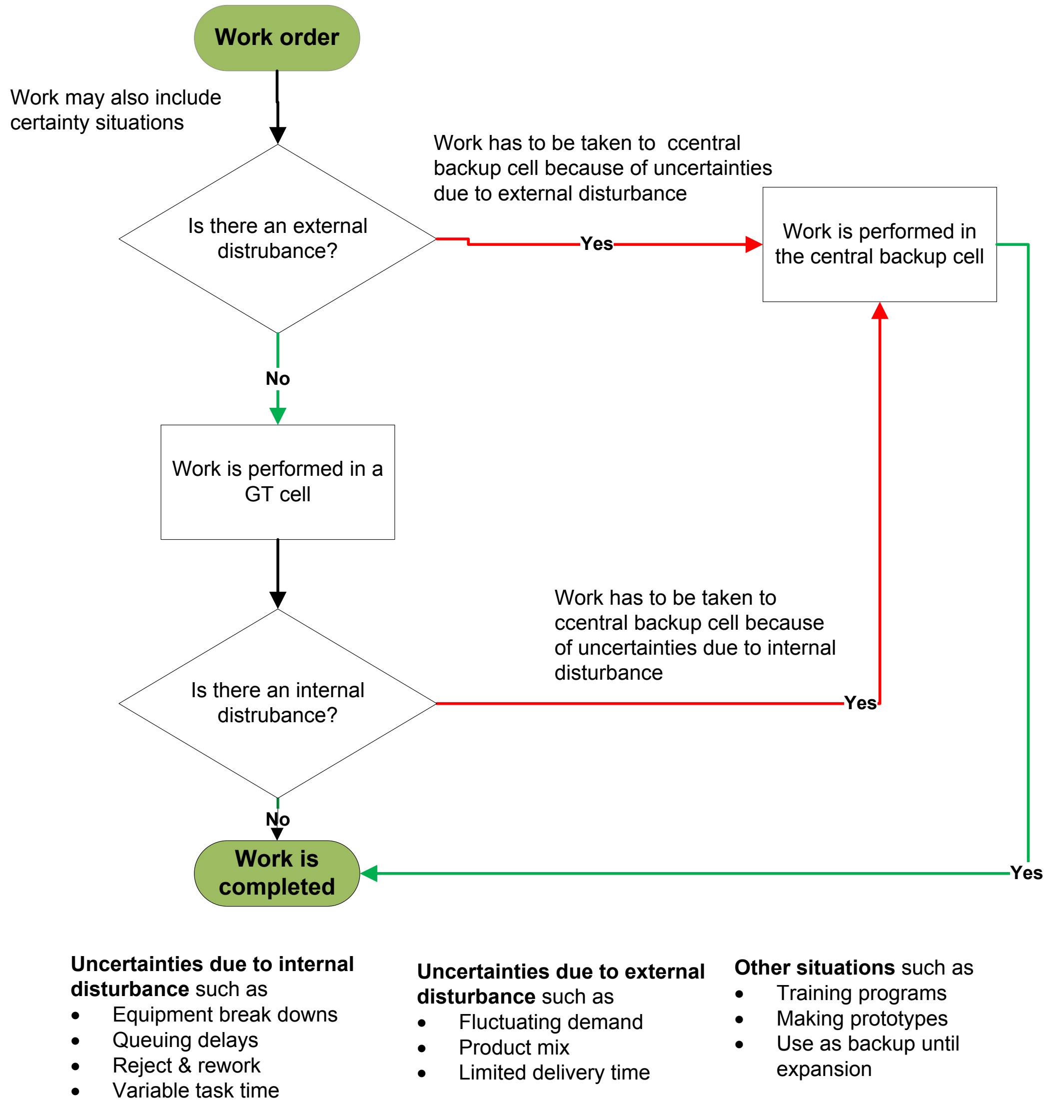


Figure 4-7: An approach for assigning work in a CBCMS environment

4.6 Understanding CBCMS Capabilities

In the literature, the traditional manufacturing systems are classified into job shop, batch and mass manufacturing systems. Job shop productions are appropriate for high part variety and low volume, whereas mass productions are suitable for high volume and low part variety. Batch productions are appropriate for medium volume and variety. Similarly, GT and fractal layouts represent cellular manufacturing systems, the arrangement of machines, cells, and workstations make the facilities more integrated, efficient, and flexible. However, these layouts are respectively considered either efficient or responsive. The proposed CBCMS layout will have characteristics of both GT and fractal layouts (efficiency and responsiveness).

First, we provide some definitions. GT is a manufacturing concept that seeks to take advantage of design and processing similarities among parts, such as grouping parts according to their geometric similarities or grouping parts according to their manufacturing similarities (Kalpakjian & Schmid, 2010). In the fractal layout, the fractals are similar units of production that are able to produce all products in all cells. Fractals are designed to minimize the WIP flow. Each fractal acts as an independent unit, generating a highly decentralized system. The fractal cells have more flexibility to handle high product variety compared with GT. However, investment and maintenance of the fractal layout can be very expensive when compared with other layouts for the same production (Venkatadri et al., 1997). The CBCMS is a combination of GT and at least one central backup cell. The central backup cell serves as a flexible cell that dynamically accommodates different types of product families. Parts not manufactured in GT cells can be re-located and processed in the central backup cell.

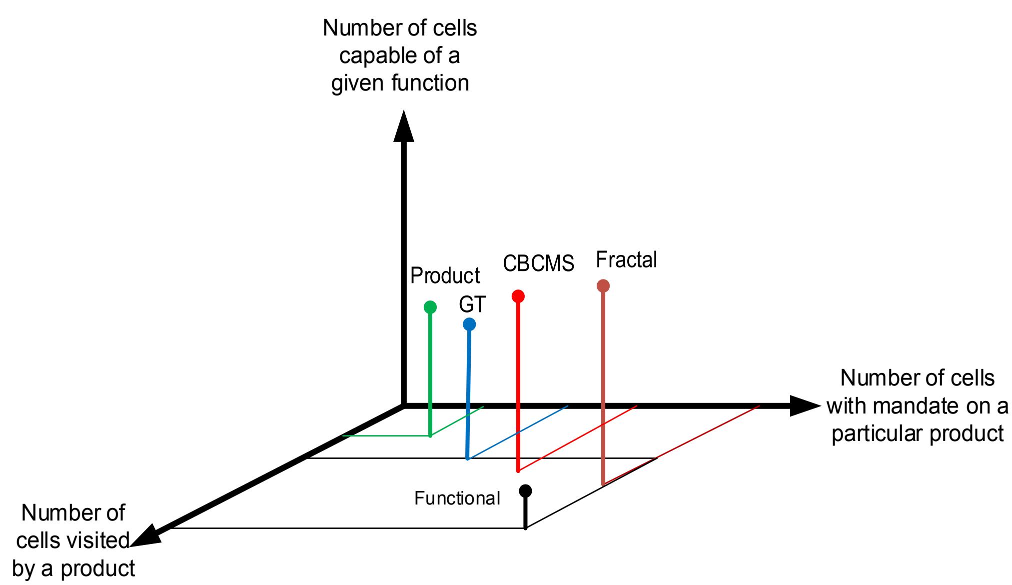


Figure 4-8: Relative position of CBCMS compared to other layouts

Figure 4-8 adapted from (Montreuil et al., 1999) illustrates the relative position occupied by the CBCMS layout in comparison to function, product, GT, and fractal layouts. In the three-dimensional coordinate system, we represent the relative position occupied by the CBCMS layout in comparison to other layouts such as GT and fractal layouts. Each axis in the three-dimensional coordinate system is labeled to represent the number of cells, respectively, with the mandate of meeting at least a fraction of a product demand, the number of cells visited by a product, and the number of cells capable of a given function.

From the point of view of number of cells capable of a given function and the number of cells with the mandate of producing a particular product, the following generalization is valid: (a) the GT layout has a lower cell capability index for a given function and a smaller number of cells with a mandate to produce a particular product; (b) the fractal layout has a higher cell capability index for a given function and a larger number of cells with the mandate to produce a particular product; and (c) the CBCMS is in between the GT and fractal layouts.

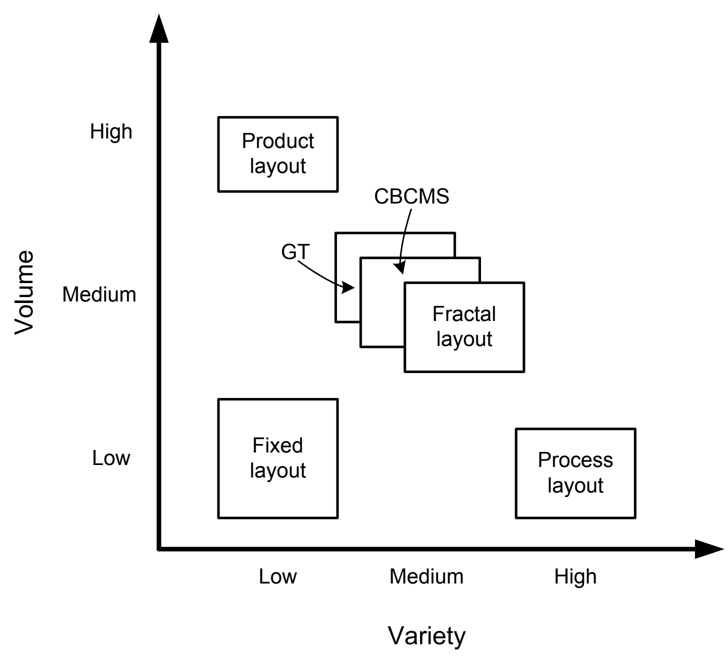


Figure 4-9: Volume-variety layout classification

(Tompkins, 2003) classified layouts based on product volume-variety, but they can also be classified as product, fixed location, process, or product family (GT). Figure 4-9, adapted from (Tompkins), shows the central backup cellular manufacturing system in comparison to product layout, fixed layout, and process layout.

The adapted figure illustrates the relative position of CBCMS and the fractal layout in comparison to GT. In cases of medium demand for a medium number of similar components, these components, according to the above classification, may be assigned to a GT manufacturing facility. However, in today's business environment, the demand may vary and the product type may change at any time. In this circumstance, GT is less responsive to these changes and therefore is not the best option. The CBCMS layout will be able to deal with product volume-variety changes and abnormalities due to uncertainty situations. A fractal layout offers even more flexibility, but may not be as efficient.

Facilities have different characteristics that influence their responsiveness and efficiency. Layout responsiveness includes a facility's capability to respond to changes in demand, meet short delivery times, handle a large variety of products, and deal with uncertainties due to internal and external disturbances. The more of these capabilities a facility has, the more responsive it is. Responsiveness, however, comes at a cost. For example, a fractal layout may have higher investment and operational costs compared to GT and CBCMS because of the higher number of similar machines distributed in all cells and the need to provide tooling and setup at multiple locations. However, fractal layouts are more responsive and flexible, and are thus able to handle a large variety of products.

The cost-responsiveness relation in Figure 4-10 shows the relative position of CBCMS in comparison to GT and fractal layouts. A GT layout has a lower responsiveness and relatively lower investment and operational costs, while a fractal layout has a higher responsiveness at relatively high costs. The CBCMS layout is, by definition, in between the two.

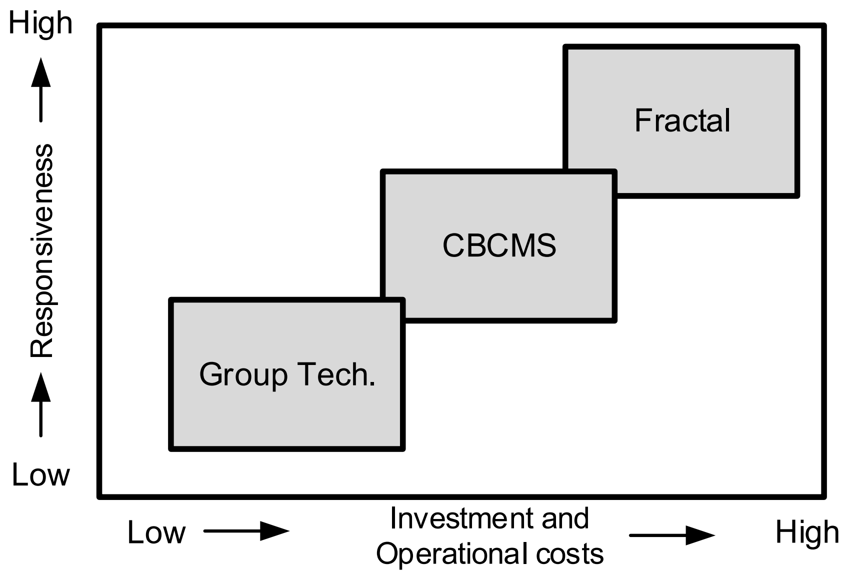


Figure 4-10: Cost and responsiveness relationship in GT, CBCMS, and fractal layouts.

4.7 The Concept of the Efficiency and Flexibility Spectrum in CBCMS

In order to adopt the CBCMS layout, we must analyze the efficiency and the flexibility of the CBCMS layout in comparison to GT and fractal layouts. Traditionally, layouts range from those that focus on being efficient to those that focus on being flexible. Generally speaking, the former is the design choice when design parameters are certain and the latter when the parameters are uncertain. We believe that, compared to GT, CBCMS is designed to be relatively more adaptive and robust, in that it responds well to changes and functions reasonably well under all scenarios. For example, the CBCMS has the capability to adjust to different kinds of variability, such as:

- Product mix variability.
- Demand uncertainty.
- Delivery time.

At the same time, it is probably CBCMS has less flexibility in comparison to the fractal layout. However, flexibility comes at a cost. For example, to respond to product mix variability, layout flexibility must be increased (i.e., move toward fractal layout), which increases cost (e.g., investment, setup, tooling, etc.). Therefore, a GT layout may be more efficient than a fractal layout, but GT has limited flexibility to handle product mix and demand uncertainty. Here, once again, CBCMS is situated in between GT and fractal layouts.

Figure 4-11 illustrates the concept of the efficiency and flexibility spectrum in CBCMS, showing where some layouts fall on this spectrum. The spectrum highlights the trade-offs involved in various strategies available for restructuring the CBCM system. As mentioned previously, the number of fractal cells in CBCMS is itself a design decision. This means that we may implement only one fractal cell in the CBCMS setting or more fractal cells, based on the layout design parameters.

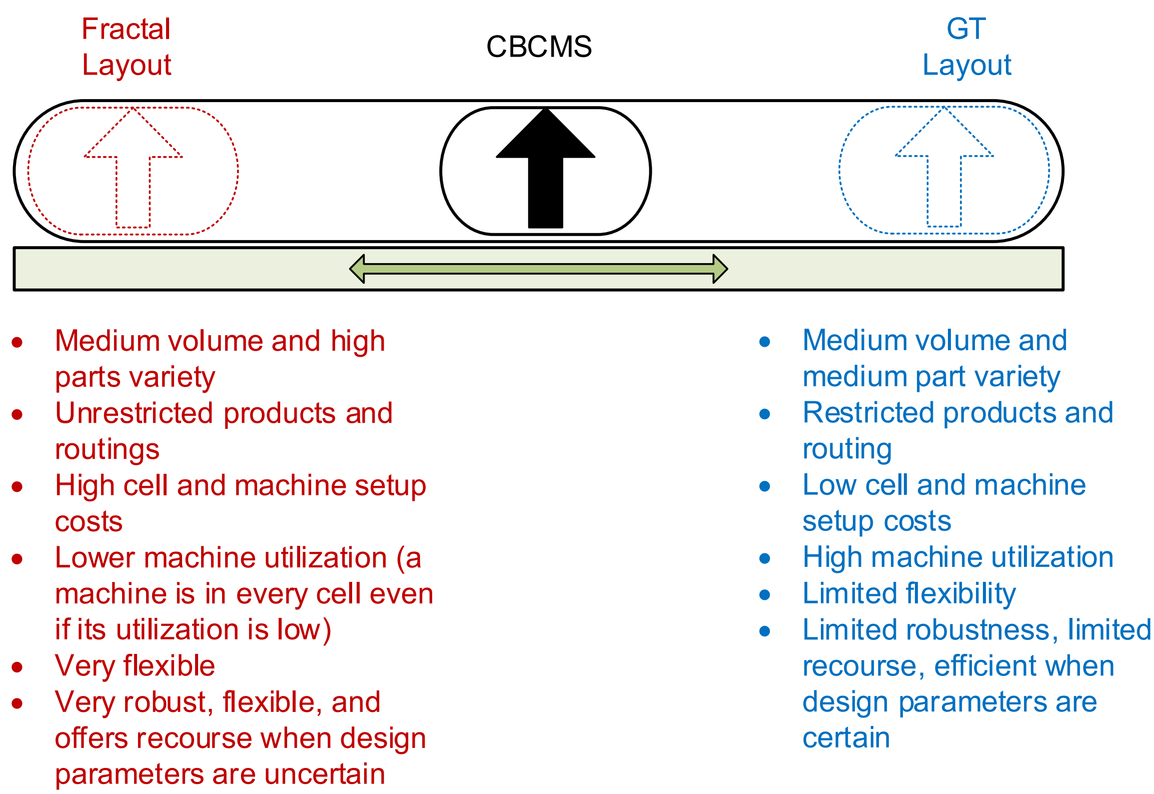


Figure 4-11: The concept of the efficiency and flexibility spectrum in CBCMS.

The positions of GT and fractal layouts are exhibited on the efficiency and flexibility spectrum in CBCMS. The GT layout that represents an inflexible layout may improve its flexibility by moving toward a fractal layout position by having more backup cells. On the other hand, it improves its efficiency by moving towards a GT layout by eliminating all backup cells. In fact, it is a trade-off between a more flexible but less efficient layout (i.e., more fractal cells) and an efficient but less flexible layout (i.e., only GT cells). The CBCMS layout may be adjusted to suit the desired level of flexibility and efficiency, allowing manufacturers to have the right balance of efficiency, adaptability, and robustness.

4.8 Analyzing the Efficiency and the Flexibility of the CBCMS Layout in Comparison to GT and Fractal Layout

In order to analyze the efficiency and the flexibility of the CBCMS layout in comparison to GT and fractal layouts, we investigate each layout separately by proposing a proper model for each and solving the cell formation problem. The MPCFP model presented in Chapter 4 can be adapted to solve the cell formation for the three layout design settings, namely GT, fractal, and CBCMS.

The data presented in (Irani, 2013) is used to test the adapted models for solving CFP of the three layout design settings. This published dataset has 19 parts (index *i* in the model) that are produced in a hypothetical machine shop that consists of 12 machines (index *j* in the model). There is only one time period in the example (index *t* in the model).

Table 4-1 contains the part number, demand, and sequence of operations (parts routings). A complete procedure and computational results for testing the adapted proposed models with the special constraint for solving the CFP is presented. Each part in the Irani case study has only one operation sequence, implying that the manufacturing sequences are pre-defined. However, in our reformulated model, all parts may be routed through alternate sequences in one time-period (i.e., through machines replicated in cells). This results in hundreds of sequences in an attempt to assign the best processing route for each part. Also, it is worth noting that we determine the inter-cell material-handling cost for each sequence in the pre-processing step.

Table 4-1: Data for part types for the Irani case study problem

Part No.	Demand	Routing (Sequence of Operations(Op))							No of operations
		Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	
1	10642	M1	M4	M8	M9				4
2	4270	M1	M4	M7	M4	M8	M7		6
3	1471	M1	M2	M4	M7	M8	M9		6
4	4364	M1	M4	M7	M9				4
5	5013	M1	M6	M10	M7	M9			5
6	4679	M6	M10	M7	M8	M9			5
7	5448	M6	M4	M8	M9				4
8	5339	M3	M5	M2	M6	M4	M8	M9	7
9	9117	M3	M5	M6	M4	M8	M9		6
10	8935	M4	M7	M4	M8				4
11	7100	M6							1
12	8611	M11	M7	M12					3
13	9933	M11	M12						2
14	3824	M11	M7	M10					3
15	1359	M1	M7	M11	M10	M11	M12		6
16	1235	M1	M7	M11	M10	M11	M12		6
17	8581	M11	M7	M12					3
18	3963	M6	M7	M10					3
19	2309	M12							1

In the following sections, we formulate and solve the Irani problem for the three layout settings (GT, fractal, and CBCMS) and show how to form the part families and machine groups (cell formation) that will constitute the cells for each layout setting.

As in Chapter 3, IBM CPLEX Optimization Studio is used to solve the optimization problems.

4.8.1 Reformulating the (MPCFP) Model with Complexity Cost for GT Design

To solve the CFP in a cellular manufacturing system, we use the proposed multi-commodity network flow-based formulation for the MPCFP presented in Chapter 3. To adapt the proposed formulation to a cellular manufacturing system based on a GT design setting, we introduce into the model the complexity cost for opening machines in cells and solve the model with various cost factors.

The complexity cost is an artificial cost in the optimization model, though practitioners will appreciate the need to keep machines of the same type in the same cell for reasons of maintenance, flexibility, and consolidation of operator expertise. The complexity cost is defined as follows: when a machine type is first introduced to a cell, a cost is incurred. However, when more machines of that type are introduced to the cell, there is no additional cost, since this does not increase cell complexity. As usual, the other cost factors include machine cost, inter-cell and intra-cell material handling costs, capacity, and demand.

The mathematical formulation of the MPCFP with the complexity cost may be written as follows:

Minimize Z:

$$\sum_j \sum_k \sum_t \left(c_j a_{jkt} + c'_j b_{jkt} + R_j u_{jkt} + g_j w_{jkt}\right) + \sum_i \sum_{p\in P_i} \sum_j \sum_t (H_i^1 h_p^1 + H_i^2 h_p^2 + m_{ipj}) x_{ipt} \quad (4.1)$$

Subject to:

$$n_{jkt} = \begin{cases} a_{jkt} & if \quad t = 1 \\ n_{jk(t-1)} + a_{jkt} - b_{jkt} + u_{jkt} - v_{jkt} & if \quad t > 1 \end{cases} \quad \forall j, \forall k \quad (4.2)$$

$$\sum_k u_{jkt} = \sum_k v_{jkt} \quad \forall j, \forall t \quad (4.3)$$

$$LM \leq \sum_j n_{jkt} \leq UM \quad \forall k, \forall t \quad (4.4)$$

$$\sum_i \sum_{p\in P_i} q_{ipjk} x_{ipt} \leq n_{jkt}\, C_j \quad \forall j, \forall k, \forall t \quad (4.5)$$

$$\sum_{p\in P_i} x_{ipt} = D_{it} \quad \forall i, \forall t \quad (4.6)$$

$$w_{jkt} \geq n_{jkt}/BigM \quad \forall j, \forall k, \forall t \quad (4.7)$$

$$w_{jkt} \in \{0,1\} \qquad \forall j, \forall k, \forall t \qquad (4.8)$$

w_{jkt} is a binary variable that takes the value 1 when machine type j is present in cell k during period t and 0 otherwise. The parameter g_j in the objective function is the opening complexity cost for machine type j. *BigM* is a large number; for computational efficiency, it should be *UM*, the maximum number of machines in a cell for a given time period.

The Irani data is used to test the adapted models for solving CFP for GT design setting. The operating cost, machine capacities, and acquisition and relocation cost are assumed to be constant over the five scenarios. We use the following additional parameters: machine acquisition cost of \$30,000, machine availability of 19,200 minutes, and a base opening complexity cost of \$30,000. The complexity cost is varied over some of the scenarios. The inter- and intra-cell movement costs are constant for all moves, regardless of the distance traveled, and their values used are \$100 and \$0, respectively. Each cell has a machine upper limit of 12, and a machine lower limit of 7. The data for part types are given in Table 4-1.

Scenario 1:

We run the model based on the given resource data mentioned in the previous paragraph. To observe how the model behaves before we use the opening complexity cost, we run the model with opening complexity cost = 0, achieved by using a G Factor value of 0 (the opening complexity cost is defined as \$30,000 x G Factor). The part family/machine cell formations resulting from implementing our reformulated model to solve the Irani dataset can be seen in the machine-part matrix in Table 4-2.

To accommodate the required demands for period one, cell one requires seven machine types [M4, M6, M7 (2), M10, M11 and M12] to process a part family of six types (11, 12, 14, 17, 18 and 19), while cell two requires nine machine types [M1, M4, M6, M7, M8, M9, M10, M11 and M12] to process a part family of 11 types (1, 4, 5, 6, 7, 10, 12, 13, 14, 15 and 16). In contrast, cell three requires 12 machine types [M1, M2, M3, M4 (3), M5, M6, M7, M8 (2) and M9] to process a part family of eight part types (1, 2, 3, 4, 7, 8, 9 and 10). Table 4-2 illustrates the formation of three cells.

From Table 4-2 and the description of cell formation in Figure 4-12, we notice that parts 1, 4, 7, 10, 12 and 14 are routed and manufactured in two cells, while the remaining parts are routed and manufactured in one cell (parts 11, 17, 18 and 19 are manufactured in cell one, parts 5, 6, 13, 15 and 16 are manufactured in cell two, and parts 2, 3, 8 and 9 are manufactured in cell three). We also notice machine replication. For example, machines M4, M6 and M7 are replicated in all cells and machines M1, M8, M9, M10, M11 and M12 are replicated in two cells, whereas machines M2, M3 and M5 in cell three are not replicated. On closer inspection, it can be seen that even though a cellular manufacturing design is encouraged in the model, the resulting design is closer to a fractal cell design.

Table 4-2: Machine-part matrix based on GT design setting "scenario 1"

Cell	Machine (QTY)	Part Families																								
		Part F1						Part F2											Part F3							
		P11	P12	P14	P17	P18	P19	P1	P4	P5	P6	P7	P10	P12	P13	P14	P15	P16	P1	P2	P3	P4	P7	P8	P9	P10
Cell 1	M4(1)																									
	M6(1)	1				1																				
	M7(2)		2	2	2	2																				
	M10(1)			3		3																				
	M11(1)		1	1	1																					
	M12(1)		3		3		1																			
Cell 2	M1(1)							1	1	1							1	1								
	M4)1)							2	2			2	1,3													
	M6(1)									2	1	1														
	M7(1)								3	4	3		2	2		2	2	2								
	M8(1)							3			4	3	4													
	M9(1)							4	4	5	5	4														
	M10(1)									3	2					3	4	4								
	M11(1)													1	1	1	3,5	3,5								
	M12(1)													3	2		6	6								
Cell 3	M1(1)																		1	1	1	1				
	M2(1)																				2			3		
	M3(1)																							1	1	
	M4(3)																		2	2,4	3	2	1	5	4	1,3
	M5(1)																							2	2	
	M6(1)																						2	4	3	
	M7(1)																			3,6	4	3				2
	M8(2)																		3	5	5		3	6	5	4
	M9(1)																		4		6	4	4	7	6	

The reason for this is a complex interplay between three factors:

1. The complexity cost encourages the GT design over the fractal cell design.

2. When the machine-part incidence matrix cannot be partitioned easily into groups, machines are duplicated across cells. This fragmentation may make the resulting design look like a fractal cell design.

3. A high material handling cost encourages fewer cell transfers and increases machine duplication.

In this example, although cell one has seven machines, the processing steps for part family F1 are restricted to machine types M6, M7, M10, M11 and M12. In the design, machine type M4 remains idle. Although, in reality, cell one does not need machine M4, it is acquired because the lower bound on the number of machines in a cell is seven. The machine-cell graph in Figure 4-12 shows the commonality of machines across cells. The machines in green are present in two cells, while those in red are present in all three cells.

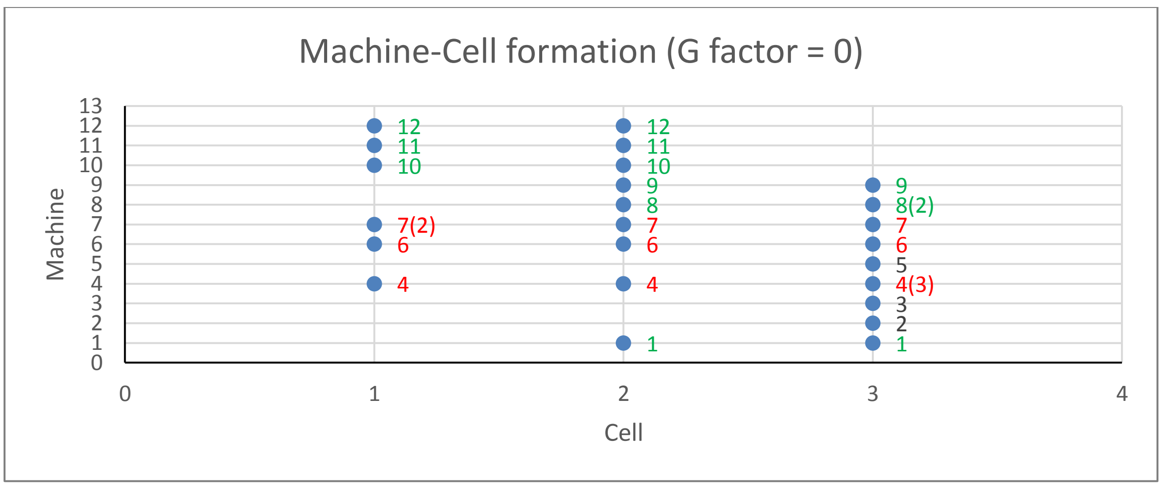


Figure 4-12: Machine-Cell graph for scenario 1.

Scenario 2:

To observe how the model behaves after we introduce the opening complexity cost, we run the model with the G factor = 1, making the opening cost per machine type $30,000. The part family/machine cell formations resulting from implementing our reformulated model to solve the Irani dataset can be seen in the machine-part matrix in Table 4-3.

To accommodate the demands required for period one, cell one requires eight machine types [M1, M4, M6 (2), M7, M8, M9 and M10] to process a part family of seven types (1, 5, 6, 7, 10, 11 and 18). Cell two requires 12 machine types [M1, M2, M3, M4 (3), M5, M6, M7, M8 (2) and M9] to process a part family of eight types (1, 2, 3, 4, 7, 8, 9 and 10). In contrast, cell three requires eight machine types [M1, M7 (2), M10, M11 (2) and M12 (2)] to process a part family of seven types (12, 13, 14, 15, 16, 17 and 19). Table 4-3 illustrates the formation of three cells for one time period.

Table 4-3: Machine-part matrix based on GT design setting "Scenario 2"

Cell	Machine (QTY)	Part Families																					
		Part F1							Part F2								Part F3						
		P1	P5	P6	P7	P10	P11	P18	P1	P2	P3	P4	P7	P8	P9	P10	P12	P13	P14	P15	P16	P17	P19
Cell 1	M1(1)	1	1																				
	M4(1)	2			2	1,3																	
	M6(2)		2	1	1		1	1															
	M7(1)		4	3		2		2															
	M8(1)	3		4	3	4																	
	M9(1)	4	5	5	4																		
	M10(1)		3	2				3															
Cell 2	M1(1)								1	1	1	1											
	M2(1)										2			3									
	M3(1)													1	1								
	M4(3)								2	2,4	3	2	2	5	4	1,3							
	M5(1)													2	2								
	M6(1)												1	4	3								
	M7(1)									3,6	4	3				2							
	M8(2)								3	5	5		3	6	5	4							
	M9(1)								4		6	4	4	7	6								
Cell 3	M1(1)																			1	1		
	M7(2)																2		2	2	2	2	
	M10(1)																		3	4	4		
	M11(2)																1	1	1	3,5	3,5	1	
	M12(2)																3	2		6	6	3	1

From Table 4-3 and the description of cell formation in Figure 4-13, we notice that only three parts out of 19 are routed and manufactured in two cells (cell one and two), while the remaining parts are routed and manufactured in one cell (parts 5, 6, 11, and 18 are manufactured in cell one; parts 2, 3, 4, 8 and 9 are manufactured in cell two; and parts 12, 13, 14, 15, 16, 17 and 19 are manufactured in cell three, respectively). We also notice some machine replication, for example: machines M1 and M7 are replicated in all cells and machines M4, M6, M8, M9 and M10 are replicated in two cells, while the remaining machines M2, M3, M5, M11 and M12 are not replicated. Figure 4-13 shows the machine-cell graph for this scenario.

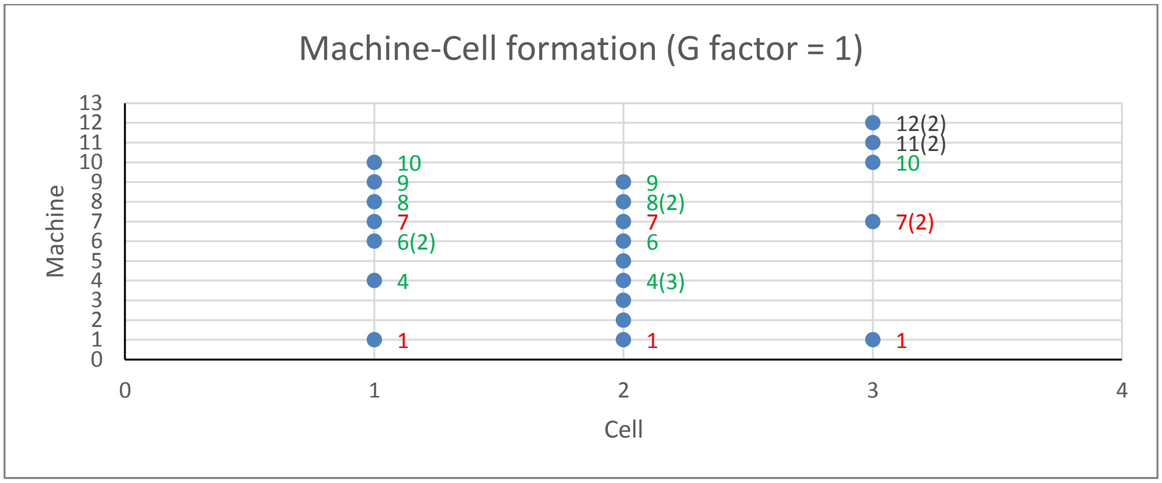


Figure 4-13: Machine-Cell graph for scenario 2.

Scenario 3:

To observe how the model behaves after we increase the opening complexity cost, we run the model with G Factor = 10. The part family/machine cell formations resulting from implementing our reformulated model to solve the Irani dataset can be seen in the machine-part matrix in Table 4-4.

To accommodate the required demands for one time period, cell one requires 11 machine types [M1, M2, M3, M4 (2), M5, M6 (2), M8 and M9(2)] to process part family one of six types (1, 3, 7, 8, 9 and 11), while cell two requires seven machine type [M7(2), M10, M11 (2) and M12 (2)] to process part family two of seven types (12, 13, 14, 15, 16, 17 and 19). In contrast, cell three requires ten machine types [M1, M4 (2) M6, M7 (2), M8 (2), M9 and M10] to process part family of seven types (2, 4, 5, 6, 7, 10 and 18). Table 4-4 illustrates the formation of three cells for one time period.

From Table 4-4 and the cell formation in Figure 4-14, we notice that only part 7 is routed and manufactured in two cells (cells one and three), while the remaining parts are routed and manufactured in one cell (i.e., parts 1, 3, 8, 9 and 11 are manufactured in cell one; parts

12, 13, 14, 15, 16, 17 and 19 are manufactured in cell two; and parts 2, 4, 5, 6, 10 and 18 are manufactured in cell three). For example, part family one, made up of parts 12, 13, 14, 15, 16, 17 and 19, is manufactured in cell one, which consists of machines M1, M2, M3, M4, M5, M6, M8 and M9. Part family two, which is made up of parts 12, 13, 14, 15, 16, 17 and 19, is manufactured in cell two, which consists of machines M7, M10, M11 and M12. In contrast, part family three, consisting of parts 2, 4, 5, 6, 7, 10 and 18, is manufactured in cell three using machine group three, which is made up of machines M1, M4, M6, M7, M8, M9 and M10. We note here as well some machine replication: machines M1, M4, M6, M7, M8, M9 and M10 are replicated in two cells, while machines M2, M3, M11 and M12 are not replicated.

Table 4-4: Machine-part matrix based on GT design setting "Scenario 3"

Cell	Machine (QTY)	Part Families																			
		Part F1						Part F2							Part F 3						
		P1	P3	P7	P8	P9	P11	P12	P13	P14	P15	P16	P17	P19	P2	P4	P5	P6	P7	P10	P18
Cell 1	M1(1)	1	1								1*	1*									
	M2(1)		2		3																
	M3(1)				1	1															
	M4(2)	2	3	2	5	4															
	M5(1)				2	2															
	M6(2)			1	4	3	1														
	M8(1)	3		3	6	5															
	M9(2)	4		4	7	6															
Cell 2	M7(2)							2		2	2	2	2								
	M10(1)									3	4	4									
	M11(2)							1	1	1	3,5	3,5	1								
	M12(2)							3	2		6	6	3	1							
Cell 3	M1(1)														1	1	1				
	M4(2)														2,4	2			2	1,3	
	M6(1)																2	1	1		1
	M7(2)		4*												3,6	3	4	3		2	2
	M8(2)		5*												1			4	3	4	
	M9(1)		6*													4	5	5	4		
	M10(1)																3	2			3

In Table 4-4, the matrix elements with superscript * are known exceptional elements and are responsible for inter-cell movements. For instance, the first three operations of part P3 are performed in cell 1 on machine types M1, M2 and M4, respectively, and the fourth, fifth, and sixth operations are performed in cell 3 on machine types M7, M8 and M9. Thus, part type P3 needs one inter-cell transfer between cell 1 and cell 3. In the same manner, inter-cell transfer occurs when part types P15 and P16 are moved from cell 1 (where the first operation is performed on machine type M1) to cell 2 (where the remaining operations are performed on machine types M7, M10, M11 and M12). Thus, part types P15 and P16 need one inter-cell transfer between cell 1 and cell 2.

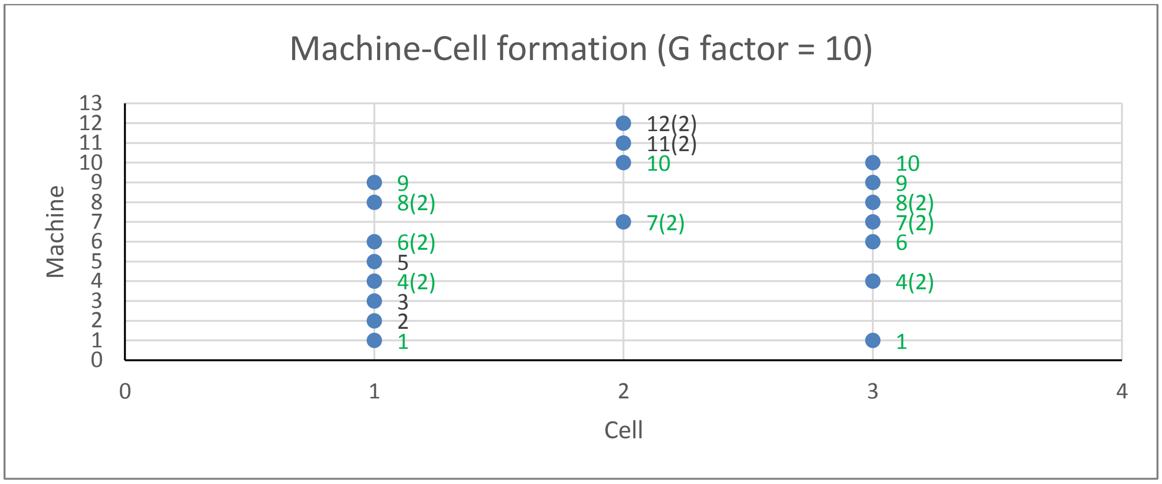


Figure 4-14: Machine-Cell graph for scenario 3

Scenario 4:

To observe how the model behaves after we increase the opening complexity cost, we run the model with G Factor = 25. The part family/machine cell formations resulting from implementing our reformulated model to solve the Irani dataset can be seen in the machine-part matrix in Table 4-5.

To accommodate the required demands for one time period, cell one requires seven machines [M7 (2), M10, M11 (2) and M12 (2)] to process part family 1 with seven products (12, 13, 14, 15, 16, 17 and 19), while cell two requires 12 machines [M1 (2), M4(2), M6, M7 (2), M8 (2), M9 (2) and M10] to process part family 2 of eight products (1, 2, 3, 4, 5, 6, 10 and 18). In contrast, cell three requires ten machines [M2, M3, M4 (2) M5, M6 (2), M8 (2) and M9] to process part family 3 of five types (1, 7, 8, 9 and 11). Table 4-5 illustrates the formation of three cells for one time period.

Table 4-5: Machine-part matrix based on GT design setting "Scenario 4"

Cell	Machine (QTY)	Part Families																			
		Part F1							Part F2								Part F3				
		P12	P13	P14	P15	P16	P17	P19	P1	P2	P3	P4	P5	P6	P10	P18	P1	P7	P8	P9	P11
Cell 1	M7(2)	2		2	2	2	2														
	M10(1)			3	4	4															
	M11(2)	1	1	1	3,5	3,5	1														
	M12(2)	3	2		6	6	3	1													
Cell 2	M1(2)				1*	1*			1	1	1	1	1				1*				
	M4(2)								2	2,4		2			1,3						
	M6(1)												2	1		1					
	M7(2)									3,6	4	3	4	3	2	2					
	M8(2)								3	5	5			4	4						
	M9(2)								4		6	4	5	5							
	M10(1)												3	2		3					
Cell 3	M2(1)										2*								3		
	M3(1)																		1	1	
	M4(2)										3*						2	2	5	4	
	M5(1)																		2	2	
	M6(2)																	1	4	3	1
	M8(2)																3	3	6	5	
	M9(1)																4	4	7	6	

From Table 4-5 and the description of cell formation in Figure 4-15, we notice that only part 1 is primarily manufactured in two cells (cells two and three), while the remaining parts are primarily manufactured in one cell (parts 12, 13, 14, 15, 16, 17 and 19 are manufactured in cell one; parts 2, 3, 4, 5, 6, 10 and 18 are manufactured in cell two; and parts 7, 8, 9 and 11 are manufactured in cell three). Part family 1, consisting of parts 12, 13, 14, 15, 16, 17 and 19, is manufactured in cell 1 using machine group 1 consisting of machine types M7, M10, M11, and M12. Meanwhile, part family 2, consisting of parts 1, 2, 3, 4, 5, 6, 10 and 18, is manufactured in cell 2 using machine group 2, which consists of machine types M1, M4, M6, M7, M8, M9 and M10. In contrast, part family 3, consisting of part types 1, 7, 8, 9 and 11, is manufactured in cell 3 using machine group 3 made up of machine types M2, M3, M4, M5, M6, M8 and M9. Here again we notice machine replication: for example, machine types M4, M6, M7, M8, M9 and M10 are replicated in two cells. The remaining machines (M1, M2, M3, M5, M11 and M12) are not replicated.

In Table 4-5, the matrix elements with superscript * are known exceptional elements and are responsible for inter-cell material handling movements. For instance, the first operation of parts P15 and P16 is performed in cell 2 on machine type M1 and the remaining operations are performed in cell 1 on machine types M7, M10, M11 and M12. Thus, each part types P15 and P16 need one inter-cell transfer each between cell 2 and cell 1. In the same manner, inter-cell transfer occurs when part type P3 is moved from cell 2 (where the first operation is performed on machine type M1) to cell 3 (where the second and third operations are performed on machine types M2 and M4) and then to cell 2 again (to perform the third, fourth and fifth operations on machines types M7, M8 and M9). Thus, part type P3 needs two inter-cell transfers between cell 2 and cell 3. In the same manner, inter-cell transfer occurs when part type P1 is moved from cell 2 (where the first operation is performed on machine type M1) to cell 3 (where the remaining operations are performed on machine types M4, M8 and M9). Thus, part type P1 needs one inter-cell transfer between cell 2 and cell 3.

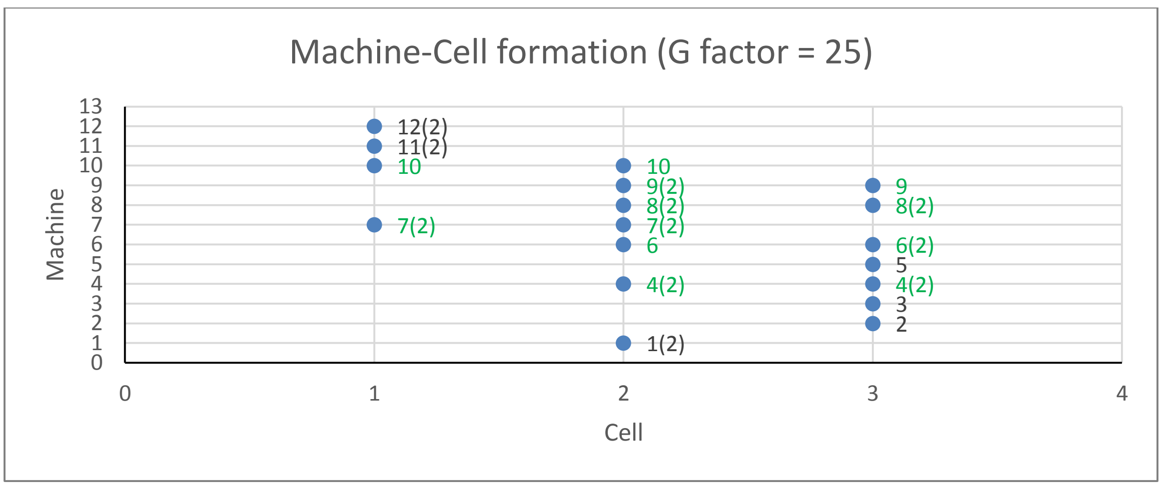


Figure 4-15: Cell-Machine graph for scenario 4

Scenario 5:

To observe how the cell formation changed and evolved after we increase the opening complexity cost, we run the model with G Factor = 100. The part family/machine cell formations resulting from implementing our reformulated model to solve the Irani dataset can be seen in the machine-part matrix in Table 4-6.

Table 4-6: Machine-part matrix based on GT design setting "Scenario 5"

Cell	Machine (QTY)	Part Families																		
		Part F1								Part F2								Part F3		
		P12	P13	P14	P15	P16	P17	P18	P19	P1	P2	P3	P4	P5	P6	P7	P10	P8	P9	P11
Cell 1	M7(3)	2		2	2	2	2	2			6*			4*	3*					
	M10(2)			3	4	4		3						3*	2*					
	M11(2)	1	1	1	3,5	3,5	1													
	M12(2)	3	2		6	6	3		1											
Cell 2	M1(2)				1*	1*				1	1	1	1	1						
	M4(4)									2	2,4	3	2			2	1,3	5*	4*	
	M7(1)										3	4	3				2			
	M8(3)									3	5	5			4	3	4	6*	5*	
	M9(2)									4		6	4	5	5	4		7*	6*	
Cell 3	M2(1)											2*						3		
	M3(1)																	1	1	
	M5(2)																	2	2	
	M6(3)							1*						2*	1*	1*		4	3	1

To accommodate the required demands for one time period, cell one requires nine machines [M7 (3), M10 (2), M11 (2) and M12 (2)] to process part family 1 of eight products (12, 13, 14, 15, 16, 17, 18 and 19), while cell two requires 12 machines [M1 (2), M4(4), M7, M8 (3) and M9 (2)] to process part family 2 of eight products (1, 2, 3, 4, 5, 6, 7 and 10). In contrast, cell three requires seven machines [M2, M3, M5 (2) and M6 (3)] to process part family 3 of three products (8, 9 and 11) Table 4-4 illustrates the formation of three cells.

From Table 4-6 and the description of cell formation in Figure 4-16, we notice that there is no part primarily manufactured in two or three cells, as was the case in our previous scenarios. All parts are primarily manufactured in only one cell (part types 12, 13, 14, 15, 16, 17, 18 and 19 are manufactured in cell one; part types 1, 2, 3, 4, 5, 6, 7 and 10 are manufactured in cell two; and part types 8, 9 and 11 are manufactured in cell three). For example, part family 1, consisting of part types 12, 13, 14, 15, 16, 17, 18 and 19, is manufactured in cell 1 using machine group 1 comprised of machine types M7, M10, M11 and M12. Part family 2, which consists of parts 1, 2, 3, 4, 5, 6, 7 and 10, is manufactured in cell two using machine group 2, which is made up of machines M1, M4, M7, M8 and M9. In contrast, part family 3, consisting of part types 8, 9 and 11, is manufactured in cell 3 using machine group 3, which consists of machine types M2, M3, M5 and M6. We notice here that only machine type M7 is replicated in two cells (cell 1 and cell 2). After a further increase in the complexity cost, we note that the cells are configured as cellular manufacturing based on GT.

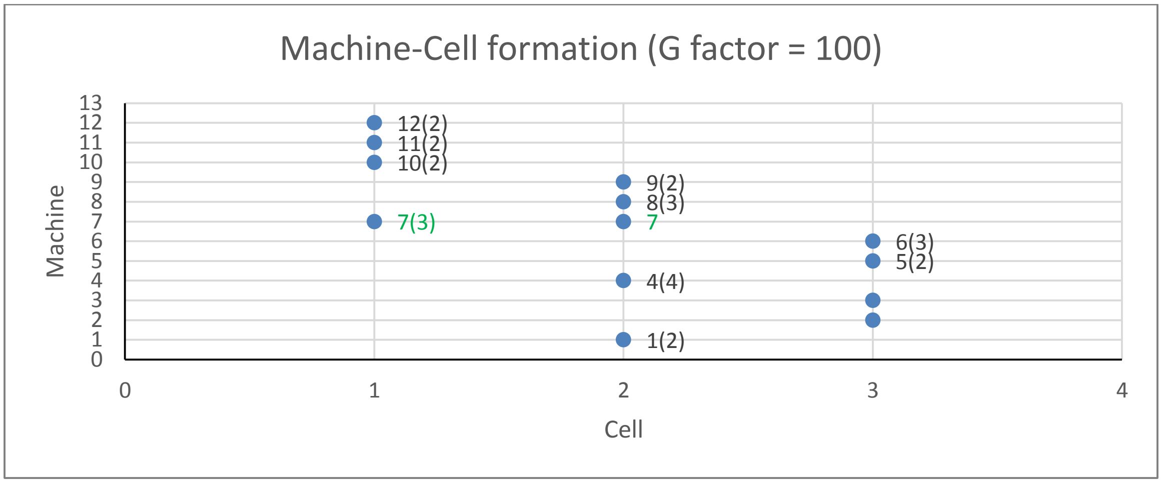


Figure 4-16: Cell-Machine graph for scenario 5

In Table 4-6, the matrix elements with superscript * are the exceptional elements and are responsible for inter-cell material handling movements. For instance, the first operation of parts P15 and P16 is performed in cell 2 on machine type M1, and the remaining operations are performed in cell 1 on machine types M7, M10, M11 and M12. Thus, each part types P15 and P16 needs one inter-cell transfer between cell 2 and cell 1. In the same manner, inter-cell transfer occurs when part type P18 is moved from cell 3 (where the first operation is performed on machine type M6) to cell 1 (where the second and third operations are performed on machine types M7 and M10). Thus, part type P18 needs one inter-cell transfer between cell 3 and cell 1. Inter-cell transfer further occurs when part type P2 is moved from cell 2 (where the first five operations are performed on machine types M1, M4, M7, M4 and M8) to cell 1 (where the sixth operation is performed on machine type M7). Thus, part type P2 needs one inter-cell transfer between cell 2 and cell 1. Similarly, inter-cell transfer occurs when part type P3 is moved from cell 2 (where the first operation is performed on machine type M1) to cell 3 (where the second operation is performed on machine type M2) and then to cell 2 again (to perform the third to sixth operations on machines types M4, M7, M8 and M9). Thus, part type P3 needs two inter-cell transfers between cell 2 and cell 3.

Inter-cell transfer occurs when part type P5 is moved from cell 2 (where the first operation is performed on machine type M1) to cell 3 (where the second operation is performed on machine type M6) and then to cell 1 again (to perform the second and third operations on machines types M7 and M10) to cell 2 again (to perform the fifth operation on machines type M9). Thus, part type P5 needs three inter-cell transfers between all cells. In the same manner, inter-cell transfer occurs when part type P6 is moved from cell 3 (where the first operation is performed on machine type M6) to cell 2 (where the second and third operations are performed on machine types M10 and M7, respectively) and then to cell 2 again (to perform the fourth and fifth operations on machines types M8 and M9, respectively). Thus, part type P6 needs two inter-cell transfers between cell 3 and cell 1 and between cell 1 and cell 2.

Inter-cell transfer occurs when part type P8 is moved from cell 3 (where the first four operations are performed on machine types M3, M5, M2 and M6, respectively) to cell 2 (where the fifth, sixth and seventh operations are performed on machine types M4, M8 and M9, respectively). Thus, part type P2 needs one inter-cell transfer between cell 3 and cell 2. In the same manner, inter-cell transfer occurs when part type P9 is moved from cell 3 (where the first three operations are performed on machine types M3, M5 and M6, respectively) to cell 2 (where the fourth, fifth, and sixth operations are performed on machine types M4, M8 and M9, respectively). Thus, part type P2 needs one inter-cell transfer between cell 3 and cell 2.

4.8.2 Solution Analysis for GT

Table 4-7 shows the number of machines in different cells across all scenarios. The second to fourth rows show the units of different machines placed in the three cells for scenario 1. As can be seen from these three rows, one unit of machine types 4, 6, 10, 11 and 12 and two units of machine type 7 are placed in cell 1. In cell 2, there is one unit of machine types 1, 4, 6, 7, 8, 9, 10, 11 and 12. In cell 3, there is one unit of machine types 1, 2, 3, 5, 6, 7 and 9, two units of machine type 8, and three units of machine type 4. The other rows in this table give machine allocations for scenarios 2, 3, 4 and 5.

The standard deviation column in the table (last column on the right) shows the standard deviation values for each of the scenarios. A low standard deviation is an indicator of how close all of the cell values are to the mean. In a fractal cell design, one expects a relatively lower value of this measure, since all cells have the same process capabilities. In GT cell design, the standard deviation increases because only certain cells have certain processes and there are several zeros in the matrix. In Table 4-7, the standard deviation measure increases with the scenario number (i.e., as the complexity cost value increases). Therefore, it may be concluded that as the complexity cost goes up, the design of the manufacturing system changes from fractal to GT.

Table 4-7: Unit of machines in different cells for all scenarios

	Cell	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	SD
	1				1		1	2			1	1	1	0.681
Scenario	2	1			1		1	1	1	1	1	1	1	
1	3	1	1	1	3	1	1	1	2	1				
	1	1			1		2	1	1	1	1			0.797
Scenario	2	1	1	1	3	1	1	1	2	1				
2	3	1						2			1	2	2	
	1	1	1	1	2	1	2		2	1				0.826
Scenario	2							2			1	2	2	
3	3	1			2		1	2	2	1	1			
	1							2			1	2	2	0.889
Scenario	2	2			2		1	2	2	2	1			
4	3		1	1	2	1	2		2	1				
	1							3			2	2	2	1.174
Scenario	2	2			4			1	3	2				
5	3		1	1		2	3							

Table 4-8 presents the part-cell allocation for different scenarios. For example, the second column shows the part-cell allocation for scenario 1. Here, all operations of parts 4, 6 and 7 are processed in all cells, all operations of parts 1, 8, 9, 10, 11 and 12 are processed in two cells, and all operations of parts 2, 3 and 5 are processed in cell 3. The sixth column shows the part-cell allocation for scenario 5. As we can see, all operations of part 7 are processed in cells 1 and 2, the other operations of parts 10, 11 and 12 are processed in cell 1, all operations of parts 1, 4, 8 and 9 are processed in cell 2, and all operations of parts 2, 3, 5 and 6 are processed in cell 3. The third, fourth, and fifth columns show the part-cell allocation for scenarios 2, 3 and 4.

Table 4-8: Part-cell allocation

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Cell 1	P4, P6, P7(2), P10, P11, P12	P1, P4, P6(2), P7, P8, P9, P10	P1, P2, P3, P4(2), P5, P6(2), P8(2), P9	P7(2), P10, P11(2), P12(2)	P7(3), P10(2), P11(2), P12(2)
Cell 2	P1, P4, P6, P7, P8, P9, P10, P11, P12	P1, P2, P3, P4(3), P5, P6, P7, P8(2), P9	P7(2), P10, P11(2), P12(2)	P1(2), P4(2), P6, P7(2), P8(2), P9(2), P10	P1(2), P4(4), P7, P8(3), P9(2)
Cell 3	P1, P2, P3, P4(3), P5, P6, P7, P8(2), P9	P1, P7(2), P10, P11(2), P12(2)	P1, P4(2), P6, P7(2), P8(2), P9, P10	P2, P3, P4(2), P5, P6(2), P8(2), P9	P2, P3, P5(2), P6(3)
Total cost $	1,243,196	1,273,196	1,649,696	2,128,396	7,034,196

Table 4-9 presents the solutions to the five scenarios. For instance, for scenario 1, the total number of machines required is 28, the total cost is $1,243,196, and machine duplication is high compared to the other scenarios that have higher complexity cost. The total number of required machines is allocated to the three cells and the design resembles a fractal cell configuration. Some part types are routed in more than one cell. All data in scenario 2 are the same as in scenario 1, except that the total number of machines is increased from 28 to 29 and the total cost is increased because the G factor increased. We notice that, in scenario 2, there are fewer instances of machine duplication compared to scenario 1 and that some parts are primarily routed in more than one cell but this number is lower compared to scenario 1.

In total, 28 machines are required in the solution of scenarios 1, 3 and 5 and 29 machines are required in scenarios 2 and 4. This indicates that 28 is the minimum number of machines required to process all operations of the parts in any of these scenarios. Comparing the solutions of all five scenarios, we find that the number of machines used to form the cells is identical, but the cell formation varies depending on the opening complexity cost associated with the scenarios.

Table 4-9: Solution results of all scenarios

	Scenarios				
	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
G Factor	0	1	10	25	100
All other factors	1	1	1	1	1
Machine upper limit	12	12	12	12	12
Machine lower limit	7	7	7	7	7
Number of cells	3	3	3	3	3
Number of machines	28	29	28	29	28
Machine cost	30000	30000	30000	30000	30000
Machine acquisition cost	840000	870000	840000	870000	840000
Intercell MH cost	0	0	406500	855200	5791000
Production cost	403196	403196	403196	403196	403196
Total cost	1243196	1273196	1649696	2128396	7034196
Opening complexity cost	0	630000	5700000	13500000	39000000
Machine duplications	12	9	7	6	1
Parts duplications	6	3	1	1	0
Total intercell entries	0	0	0.9	0.983	3.193
Type of layout	Fractal		Mixed between Fractal and GT		GT

The total cost and the total inter-cell entries (number of inter-cell transfers/number of parts) are increased when the opening complexity cost is increased. This is also reflected in the inter-cell MH cost row. In other words, as the complexity cost is increased, the model prefers to restrict duplication of machines to as few cells as possible. To compensate, parts have to move between cells, which is the reason for higher inter-cell transfer.

However, machine and part duplications decrease when the opening complexity cost is increased. In the GT setting, we can see how the cells evolve from basic fractal cells in scenario 1, to a combination between GT and fractal cells by scenario 3, and to GT cells in scenario 5. Based on the five different scenarios that have been explored in the GT setting, increasing the opening complexity cost influences the design, making it more fractal-like or GT-like.

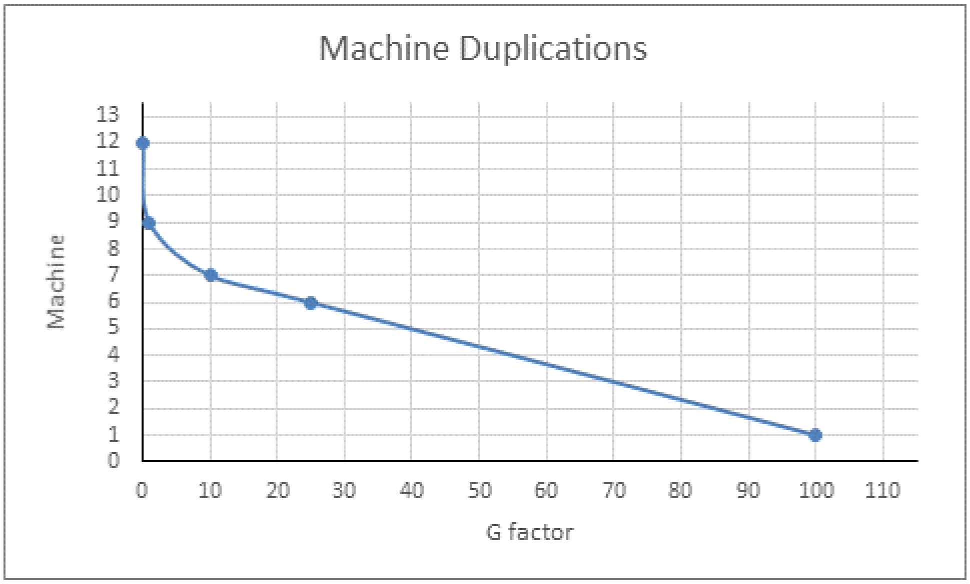


Figure 4-17: Machine duplication as a function of the G factor

Figure 4-17 illustrates the machine duplications (number of machines present in more than one cell) with respect to opening complexity cost. The machine duplications is high at lower values of the complexity factor. When this happens, processes are spread through the various cells and the manufacturing system resembles a fractal cell layout. However, the more we increase the opening complexity cost, the greater the reduction in machine duplications and the greater the specialization of cells, leading to an overall design that is closer to GT.

Figure 4-18 illustrates part duplications with respect to opening complexity cost. Some part types require more than one cell in order to process the required demand. If a part is processed, in say three cells, we say that the measure of parts duplication is 2 (since the part needs to be made somewhere, the number of part duplications is 3-1 = 2). Part duplication should not be confused with intercell transfers. Parts duplication occurs when a part's first operation is planned to occur in multiple cells. Intercell transfer is a different concept; there is intercell transfer if a part crosses cell boundaries in the course of its manufacturing sequence.

The number of part duplications is higher when the opening complexity cost is low. In other words, parts are manufactured in several cells, which is a fractal cell layout feature. However, when the opening complexity cost is increased, part duplications is decreased and part families are processed in one cell, which is a GT cell layout feature.

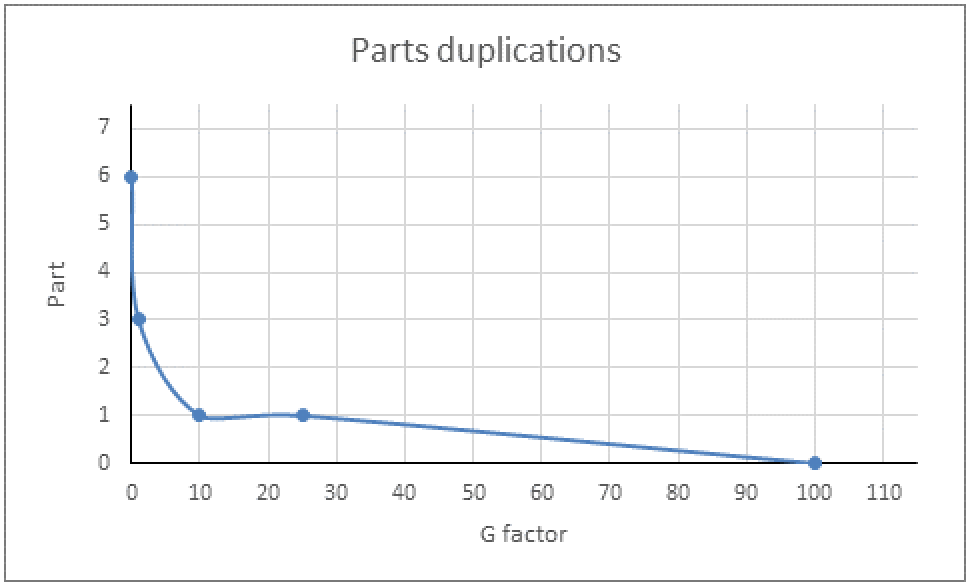


Figure 4-18: Parts duplications as a function of the G factor.

Figure 5.19 illustrates the number of inter-cell entries with respect to opening complexity cost. The transfer between cells increases as the opening complexity cost is increased. The cells are specialized for a few processes (GT cell feature) and more products need to visit multiple cells for their processing requirements.

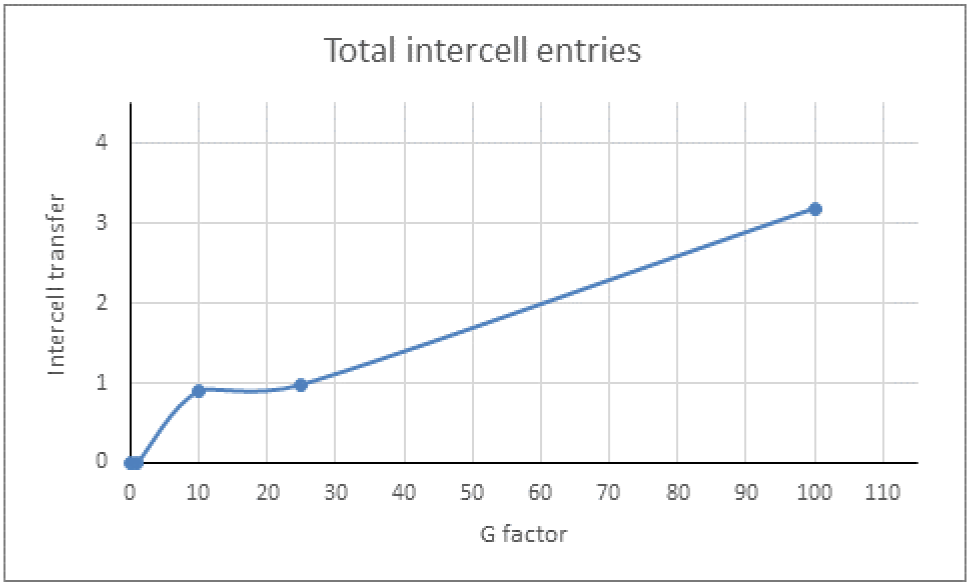


Figure 4-19: Inter-cell transfer as a function of the G factor.

Figure 4-20 illustrates the total cost with respect to the opening complexity cost. The more we increase the opening complexity cost, the higher the cost of creating cellular manufacturing.

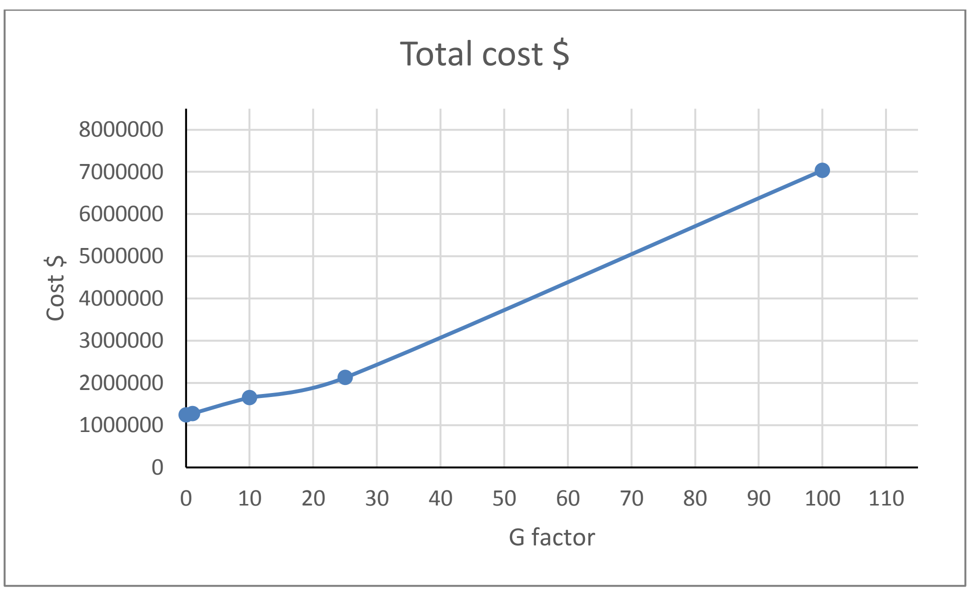


Figure 4-20: Total cost as a function of the G factor.

The computational results indicate the following:

- When the opening complexity cost equal zero, there is no penalty if we introduce any machine to a cell. In this case, the model distributes the machines on the available cells without any restrictions, resulting in a fractal-like layout. Some part families can be routed and manufactured in more than one cell to reduce material handling.
- When the opening complexity cost is increased, a fixed cost is incurred when a machine is introduced to a cell. However, when more machines of that type are introduced to the cell, there is no additional cost – we have explained earlier that this is because having a machine type available in multiple cell has a cost (there is more complexity, there are more setup and supervision requirements, etc.). In this case, the model distributes the machines to available cells without incurring any extra opening complexity cost. This results in a GT-like layout where each family part could be routed and manufactured in a cell. However, the number of inter-cell transfers increases because some parts requiring processes not in the cell have to be transferred to other cells.

4.8.3 Adapting the (MPCFP) Model to a Fractal Design Setting

We now use the proposed multi-commodity network flow-based formulation presented in Chapter 3 to solve the MPCFP in the fractal setting. To adapt the proposed formulation to a fractal design setting, we add a constraint called fractal constraint. The fractal constraint forces the MPCFP model to design fractal cells.

Unlike for GT design, the machine requirement calculation for fractal design setting may be done outside the optimization model and is based on the required and available processing time. The reader is referred to (Montreuil et al., 1999) for details on this calculation. Once the number of machines is calculated, the cell designer decides on the number of fractal cells and the number of machine type j in fractal cell k (N_{jkt}) in period t.

To enforce this in the MPCFP model, a simple constraint is added to ensure that the number of machines in each fractal cell is at least N_{jkt}.

$$n_{jkt} \geq N_{jk} \qquad \forall j, \forall k, \forall t$$

Scenario 1: Basic fractal cellular manufacturing system

In this scenario, we take the (Irani, 2013) example and allocate the total number of machines for all processes across three cells. In this example, we have a total of 28 machines allocated to three cells: cell one has ten machines, and cell two and three have nine machines each. This not a fully balanced fractal cell design.

The part sequences are entered as input to the optimization model in Cplex Studio with the Irani data. On all runs using a 64-bit machine with an Intel i5 chipset running at 3.20 GHz, the program comes up with an optimal solution in a very short time (about 10.24 seconds).

As in the GT scenario, the inter-cell material handling cost is $100, the machine acquisition cost is $30,000, and the machine availability per month is 19,200 minutes. The machine upper limit is set to 20 for each cell and the machine lower limit is set to 5 for each cell. Since the Irani problem has only one time period, the number of machines of type j in cell k (N_jk) is:

[[1,0,1],[1,0,0],[1,0,0],[1,1,2],[1,0,0],[1,1,1],[1,2,1],[1,1,1],[1,1,1],[1,1,0],[0,1,1],[0,1,1]]

The first array above ([1,0,1]) implies that one unit of machine one is available in cells one and three. The second array ([1,0,0]) implies that one unit of machine two is available only in cell one.

The part family/machine cell formations resulting from implementing our adapted fractal design setting model to solve the Irani dataset can be summarized as follows:

To accommodate the demands required for period one, cell one requires ten machine types [M1, M2, M3, M4, M5, M6, M7, M8, M9 and M10] to process part families (3, 4, 5, 8, 9 and 10). Cell two requires eight machine types [M4, M6, M7(2), M8, M9, M10, M11 and M12] to process part families (6, 7, 9, 10, 11, 12, 14, 16, 17 and 18). In contrast, cell three requires eight machine types [M1, M4(2), M6, M7, M8, M9, M11 and M12] to process part families (1, 2, 4, 10, 11, 12, 13, 15 and 19). Table 4-10 illustrates the formation of the three fractal cells.

Table 4-10: Cell formation for basic fractal design setting

Cell 1	Part family 1	Parts 3, 4, 5, 8, 9, 10
	Machine group 1	Machines M1, M2, M3, M4, M5, M6, M7, M8, M9, M10
Cell 2	Part family 2	Parts 6, 7, 9, 10, 11, 12, 14, 16, 17, 19
	Machine group 2	Machines M4, M6, M7(2), M8, M9, M10, M11, M12
Cell 3	Part family 3	Parts 1, 2, 4, 10, 11, 12, 13, 15, 19
	Machine group 3	Machines M1, M4(2), M6, M7, M8, M9, M11, M12

From Table 4-10 and the description of cell formation, we notice that parts 4, 9, 10, 11 and 12 are routed and manufactured in two cells, while the remaining parts are routed and manufactured in one cell (parts 3, 5 and 8 are manufactured in cell one; parts 6, 7, 14, 16, 17 and 18 are manufactured in cell two; and parts 1, 2, 13, 15 and 19 are manufactured in cell three). Thus, the cells are configured as basic fractal cells and the machine replication in this design setting follows the fractal arrangement. For example, machines M4, M6, M7, M8 and M9 are replicated in all cells, and machines M1, M10, M11 and M12 are replicated only in two cells. Machines M2, M3 and M5 in cell one are not replicated.

Scenario 2: Balanced fractal cellular manufacturing system

In this setting, we replicate machines and allocate the total number of machines for most processes equally across the three fractal cells, with the exception of an extra machine for machine 7 in cell 2 and machine 4 in cell 3. Therefore, in this scenario, we have a total of

38 machines allocated in three cells; cell one has twelve machines, and cells two and three have thirteen machines each.

As a final step, the sequences are entered as input to the optimization model in CPLEX Studio with the Irani data. The problem is again run on a 64-bit machine with an Intel i5 chipset running at 3.20 GHz, and the program is able to find the optimal solution in 9.60 seconds.

Design, in this scenario, is based on an inter-cell material handling cost of $100, a machine acquisition cost of $30,000, and a machine availability per month of 19,200 minutes. To observe how the adapted model behaves after adding the fractal constraint, we run the model with the following parameters: the number of cells equals three, and, in each cell, the machine upper limit is 20 while the lower limit is 5. N_jk, the number of allocated machines of type j in cell k for duplicated fractal cells is as follows:

[[1,1,1],[1,1,1],[1,1,1],[1,1,2],[1,1,1],[1,1,1],[1,2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]],

The machine one allocation is [1,1,1], which means that one unit of machine one is available in cells one, two, and three. All processes are balanced except for machine 4 in cell 3 and machine 7 in cell 2. The allocation of machine four is [1,1,2], which means that one unit of machine four is available in cells one and two, while cell three has two units of machine four. The allocation of machine seven is [1,2,1], which means that one unit of machine seven is available in cells one and three, while cell two has two units of machine seven.

The part family/machine cell formations resulting from implementing our adapted fractal design setting model to solve the Irani dataset can be summarized as follows:

To accommodate the demands required for period one, cell one has one machine each of machine types [M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12] to process part families of seven types (6, 9, 10, 13, 14, 16 and 19). Cell two has 12 machine types [M1, M2, M3, M4, M5, M6, M7 (2), M8, M9, M10, M11 and M12] to process part families of ten types (1, 2, 3, 4, 5, 11, 12, 17, 18 and 19). Cell three has 12 machine types [M1, M2, M3, M4 (2), M5, M6, M7, M8, M9, M10, M11 and M12] to process part families of six part types (1, 4, 7, 8, 10 and 15). Table 4-11 illustrates the formation of three cells for one time period.

Table 4-11: Cell formation for duplicated fractal design setting

Cell 1	Part family 1	Parts 6, 9, 10, 13, 14, 16, 19
	Machine group 1	Machines M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12
Cell 2	Part family 2	Parts 1, 2, 3, 4, 5, 11, 12, 17, 18, 19
	Machine group 2	Machines M1, M2, M3, M4, M5, M6, M7(2), M8, M9, M10, M11, M12
Cell 3	Part family 3	Parts 1, 4, 7, 8, 10, 15
	Machine group 3	Machines M1, M2, M3, M4(2), M5, M6, M7, M8, M9, M10, M11, M12

From Table 4-11 and the description of the cell formation, we notice that parts 1, 4, 10 and 19 are routed and manufactured in two cells, while the remaining parts are routed and manufactured in one cell (parts 6, 9, 13, 14 and 16 are manufactured in cell one; parts 2, 3, 5, 11, 12, 17 and 18 are manufactured in cell two; and parts 7, 8 and 15 are manufactured in cell three). This reflects that the cells are configured as balanced fractals. The machine replication in this design setting follows the fractal arrangement. For example, all machine types are replicated in all cells, with more machines of types M4 and M7.

Table 4-12 summarizes the solutions to the two scenarios. Scenario 1 has fewer machines (28) and a higher material handling cost ($286,300). Moreover, some part types are routed in more than one cell. For instance, part type (10) is processed in all cells in order to meet the required demand, while part types (4, 9, 11, 12 and 19) are processed in 2 cells in order to meet the required demand.

Table 4-12: Solutions to the two scenarios

	Scenarios	
	Scenario 1	Scenario 2
Design setting	Basic fractal	Balanced fractal
All cost factors	1	1
Machine upper limit	20	20
Machine lower limit	5	5
Number of cells	3	3
Machine availability (min/month)	19,200	19,200
Machine cost $	30,000	30,000
Inter-cell MH cost/movement $	100	100
Number of machine	28	38
Machine acquisition cost $	840000	1140000
Production cost $	403,196	403,196
Inter-cell MH cost $	286300	0
Total cost $	**1529496**	**1543196**

Scenario 2 has more machines (38) and a zero material handling cost ($0). We notice that some part types are primarily routed in more than one cell, but instances of this are fewer compared to scenario 1. For example, part types (1 and 4) are routed in cells 2 and 3, and part type (10) is routed in cells 1 and 2.

Because there are fewer machines in scenario 1, there is part flow between cells in order to meet the required production. This situation contrasts with scenario 2, where the presence of more machines permits the parts to be processed without incurring any flow between cells.

4.8.4 Adapting the (MPCFP) Model to the CBCMS Design Setting

We can use the MPCFP model presented in Chapter 3 for the CBCMS setting. To adapt the proposed multi-commodity network flow-based formulation to a CBCMS basic design setting, we add the CBCMS constraint. This constraint forces the MPCFP model to design a backup cell with no constraints on other cells. The backup cell in this design setting is similar to fractal cell and has at least one machine from each type of the available machines. The CBCMS constraint is added to our model to ensure that one machine of each machine of type j is available in the backup cell k in period t.

$$n_{jkt} \geq 1 \qquad\qquad k = 1 \quad \forall j, \forall k, \forall t$$

If the CBCMS design needs to have more than one backup cell, this constraint can be imposed for the other cells as well.

Scenario one: CBCMS basic design with no constraints on other cells

The data presented in (Irani, 2013) is used to test the proposed adapted CBCMS basic design setting model. CPLEX Studio is once again used to run on a 64-bit machine with an Intel i5 chipset running at 3.20 GHz. The optimal solution can be found in about 16 seconds.

The design in this scenario is based on an inter-cell material handling cost of $100, a machine acquisition cost of $30,000, and machine availability of 19,200 minutes per month. We run the model with the following parameters: number of cells equal three, the upper limit for machines in each cell is equal to 20, and the lower limit for machines in each cell is equal to 5.

The part family/machine cell formations resulting from implementing our adapted CBCMS design model to solve the Irani dataset can be summarized as follows;

- Total number of required machines = 28
- Machine acquisition cost = $840,000
- Production cost = $403,196
- Total cost = $1,243,196

Table 4-13: Cell formation for CBCMS design setting

Cell 1 (Backup cell)	Part family 1 Machine group 1	Parts 1, 3, 4, 5, 6, 8, 9, 12, 13, 15, 17,19 Machines M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12
Cell 2	Part family 2 Machine group 2	Parts 7, 10 Machines M4, M6, M7, M8, M9
Cell 3	Part family 3 Machine group 3	Parts 1, 2, 4, 5, 6, 10, 11, 12, 13, 14, 16, 18 Machines M1, M4(2), M6, M7(2), M8, M9, M10, M11, M12

To accommodate the demands required for period one, cell one (the backup cell) requires one machine of each machine type [M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12] to process 12 part families (1, 3, 4, 5, 6, 8, 9, 12, 13, 15, 17 and 19). Cell two requires five machine types [M4, M6, M7, M8 and M9] to process two part families (7 and 10). Cell three requires nine machine types [M1, M4(2), M6, M7(2), M8, M9, M10, M11 and M12] to process 12 part families (1, 2, 4, 5, 6, 10, 11, 12, 13, 14, 16 and 18). Table 4-13 illustrates the formation of the three cells in the CBCMS design setting.

From Table 4-13 and the model output, we notice that parts 1, 4, 5, 6, 10, 12 and 13 are routed and primarily manufactured in two cells, while the remaining parts are routed and primarily manufactured in one cell (parts 3, 8, 9, 15, 17 and 19 are manufactured in the backup cell one; part seven is manufactured in cell two; and parts 2, 11, 14, 16 and 18 are manufactured in cell three). This reflects that the cells are configured as central backup cellular manufacturing to process the required parts in one time period. We again notice machine replication, in that machines M4, M6, M7, M8 and M9 are replicated in all cells and machines M1, M10, M11 and M12 are replicated in two cells. However, machine M2, M3 and M5 in the backup cell one are not replicated.

Scenario two: Changing the capacity factor for CBCMS cell

In this scenario, we change the capacity factor for the CBCMS cell. The capacity factor denotes the fraction of time that machines in the CBCMS cell are available for production. Since the CBCMS cell may be used during internal or external disturbances or for other reasons, it is not desirable to allocate a high level of production to the CBCMS cell.

Capacity factors of (0.8, 0.6, 0.4, 0.2 and 0) are investigated. A capacity factor of 0.8, for example, implies that only 80% of machine time in the CBCMS cell may be assigned for production in the design phase.

To observe how the adapted model behaves after adding the CBCMS constraint and parameters, we run the model with the following parameters: the number of cells equal three, the upper limit for machines in each cell is equal to 20, and the lower limit for machines in each cell is equal to 5. Furthermore, inter-cell material handling cost is $100, the machine acquisition cost is $30,000, and the capacity factor is 1. This means that machine availability per month is 19,200 minutes for all cells except the CBCMS cell.

Table 4-14: Solution result for scenario two

	Scenario Two				
	Case One	Case Two	Case Three	Case Four	Case Five
Capacity factor for CBCMS cell	0.8	0.6	0.4	0.2	0
Machine availability for CBCMS cells	15360	11520	7680	3840	0
Capacity factor for other cells	1	1	1	1	1
Machine availability for other cells	19200	19200	19200	19200	19200
Inter-cell material handling factor	1	1	1	1	1
G Factor	0	0	0	0	0
Total number of machines	31	33	34	36	39
Machine acquisition cost	$930,000	$990,000	$1,020,000	$1,080,000	$1,170,000
Production cost	$403,196	$403,196	$403,196	$403,196	$403,196
Inter-cell material handling cost	0	0	0	0	0
Total cost	$1,333,196	$1,393,196	$1,423,196	$1,483,196	$1,573,196
Type of layout	CBCMS	CBCMS	CBCMS	CBCMS	Not CBCMS

The results of scenario two (five cases) are shown in Table 4-14, but for the sake of greater detail, we will elaborate on case one. In this case, the capacity factor for CBCMS cell is equal to 0.8, which means that the machine availability per month for a CBCMS cell is 15,360 minutes. The part family/machine cell formations resulting from implementing our CBCMS design model to solve the Irani example can be summarized as follows:

- Total number of required machines = 31
- Machine acquisition cost = $930,000
- Production cost = $403,196
- Total cost = $1,333,196

As seen in Table 4-14, as the capacity factor for the CBCMS cell increases, the number of machines required also increases. The cell formation for case one (based on CBCMS setting) is shown in Table 4-15

Table 4-15: Machine-part matrix based on CBCMS setting "Case one"

Machine type	Part Families																									
no. of machine	P1	P3	P4	P5	P6	P8	P9	P12	P13	P15	P17	P19	P7	P10	P1	P2	P4	P5	P6	P10	P11	P12	P13	P14	P16	P18
M1(1)	1	1	1	1						1																
M2(1)		1				1																				
M3(1)						1	1																			
M4(1)	1	1	1			1	1																			
M5(1)						1	1																			
M6(1)				1	1	1	1																			
M7(1)		1	1	1	1			1		1	1															
M8(1)	1	1			1	1	1																			
M9(1)	1	1	1	1	1	1	1																			
M10(1)				1	1					1																
M11(1)								1	1	2	1															
M12(1)								1	1	1	1	1														
M4(1)													1	2												
M6(1)													1													
M7(1)														1												
M8(1)													1	1												
M9(1)													1													
M1(1)															1	1	1	1							1	
M4(2)															1	2	1			2						
M6(1)																		1	1		1					1
M7(2)																2	1	1	1	1		1		1	1	1
M8(1)															1	1			1	1						
M9(1)															1		1	1	1							
M10(1)																		1	1					1	1	1
M11(1)																						1	1	1	2	
M12(1)																						1	1		1	

To conclude, this chapter showed the design continuum between fractal and GT cell layouts. The CBCMS design is an intermediate design within this continuum but illustrates the trade-offs between fractal and GT design choices. There is a high degree of flexibility in the fractal design because products can be routed in multiple cells. In contrast, there is a much lower degree of flexibility in GT layout design. However, the GT layout may be more efficient because part families are generally processed within a cell and thus common setup and tooling efficiencies may be exploited. There is also a trade-off between the number of machines in a layout and the inter-cell material handling cost.

Finally, this chapter discussed how the multi-commodity network based formulation for the MPCFP in Chapter 3 may be used with minor variations to examine fractal, GT, and CBCMS layout designs.

Chapter 5: A Robust Optimization Model for Solving Cell Formation Problem Under Uncertainty

5.1 Overview

In this chapter, we discuss an approach to introduce the notion of Robust Optimization (RO) to the MPCFP model. Product demand is very important to the facility designer when making facility layout decisions. Considerations of demand uncertainty during the process of facility design can significantly improve the quality of the solution arrived at.

We have seen how the MPCFP non-spatial model incorporates demand changes from period to period. In this chapter, we extend and reformulate the MPCFP model (presented in Chapter 3) to scenarios where demand is uncertain from period to period but may be captured using a finite set of demand scenarios.

In their book, (Ben-Tal et al., 2009) state that some data entries such as future demand and returns do not exist when the problem is solved, and hence they replace these entries with forecasts. Because data entries are subject to prediction errors, obtaining a satisfactory forecast for demand is critical. The researchers further contend that data uncertainties may affect the feasibility and optimality of a design solution. In some instances, small data uncertainties may not significantly change the optimal solution, while in other instances, a small percentage of data uncertainty (as low as 0.1%) can make the optimal solution infeasible and thus practically meaningless.

In the facilities design context, this rarely happens. If product demand is much higher than anticipated, the recourse is to expand the facility or, in the case of a cellular manufacturing system, route products to other cells that may have unused capacity. Another feasible option is to outsource production. If product demand is much lower than anticipated, a manufacturer may accept outsourced production contracts.

5.2 Robust Optimization Extension for the MPCFP

To solve the CFP in a dynamic environment, we use the robust optimization framework of (Mulvey & Vanderbei, 1995) to adapt the multi-commodity network flow-based formulation of the MPCFP.

The robust framework in the paper by (Mulvey & Vanderbei, 1995) introduces trade-offs for model robustness. To introduce robustness to the model, goal programming (penalty) coefficients are used to measure the shortfall or excess in demand and machine capacity. The MPCFP model has two types of decision variables: investment variables to reflect machines in the layout and flow variables to represent production. The recourse on investment variables is only from period -to -period. However, both GT and fractal layout designs have an inherent flow recourse.

The proposed model for solving the RMPCFP is a non-spatial cell formation model in which the underlying problem is a multi-period multi-commodity network flow problem. The robust model measures the shortfall or excess in machine capacity and demand.

The model can handle alternate machine routings and consider multiple products, uncertain demands, and multiple time periods. The objective of the mathematical formulation is to minimize total system costs while withstanding demand uncertainty. The total system costs include the costs of purchasing, discarding, relocating, inter-cell and intra-cell material handling, and manufacturing. The constraints are on:

- Calculating the number of machines in a cell by purchasing, discarding, and/or relocating machines from time period to time period.
- Respecting the constraints on the minimum and maximum number of machines in a cell.
- Respecting machine capacity.
- Ensuring that there is sufficient production of parts to satisfy demand.

- Ensuring that goal programming and penalty coefficient are used to measure the shortfall or excess in machine capacity.
- Ensuring that goal programming and penalty coefficient are used to measure the shortfall or excess in demand.

The following notation (consistent with the MPCFP model in Chapter 3) is used to develop the mathematical representation of the objective function and design constraints.

The indices in the model are:

i = An index of parts that need to processed.
j = An index of machine types.
k = An index of cells.
t = An index of time periods over which the system is being designed.
p = An index representing a start-to-finish path for part i.
s = An index representing a discrete scenario for part i.

The following sets are used in the model:

$\{S_i\}$ = The set of all possible sequences for part i.
$\{P_i\}$ = The set of start-to-finish paths for product i based on $\{S_i\}$.
$\{\Omega\}$ = A set of scenarios $\{1,2,3,\dots,s\}$.

The parameters are:

D_{its}= Demand for part i in period t under scenario s.
C_j= Time availability of one unit of machine type j per time period.
c_j= Cost of purchasing one unit of machine type j.
c'_j= Cost of discarding one unit of machine type j.
R_j= Cost of relocating machine type j between cells.
H_i^1= Cost of inter-cell material handling associated with the movement of one unit of part i.

H_i^2 = Cost of intra-cell material handling associated with the movement of one unit of part i.

LM = Minimum number of machines per cell (Lower limit).

UM = Maximum number of machines per cell (Upper limit).

h_p^1= Number of inter-cell transfers in path p.

h_p^2= Number of intra-cell transfers in path p.

m_{ipj}= Manufacturing cost/unit for part i on machine type j when routed through path p.

q_{ipjk}= Processing time/unit for part i on machine type j in cell k when routed through path p.

$Prob_s$= Probability of occurrence of demand scenario s

The decision variables are:

x_{ipts}= Number of parts of type i routed through path p in period t under scenario s.

n_{jkt}= Number of machines of type j available in cell k in period t.

u_{jkt}= Number of machines of type j moved into cell k in period t.

v_{jkt}= Number of machines of type j moved out of cell k in period t.

a_{jkt}= Number of machines of type j purchased in cell k in period t.

b_{jkt}= Number of machines of type j discarded from cell k in period t.

Error variables and penalty coefficients:

f_{jkts}^{+}= Error variable to measure the shortfall in machine capacity for scenario s.

f_{jkts}^{-}= Error variable to measure the excess in machine capacity for scenario s.

e_{its}^{+}= Error variable to measure the shortfall in demand for scenario s.

e_{its}^{-}= Error variable to measure the excess in demand for scenario s.

ω_1= Penalty coefficient for shortfall in machine capacity.

ω_2= Penalty coefficient for shortage in demand.

Using the above notation, the mathematical formulation including objective function and system constraints are now written in robust optimization equation form, as follows:

Minimize:

$$Z_R = \sum_j \sum_k \sum_t \left(c_j a_{jkt} + c'_j b_{jkt} + R_j u_{jkt}\right)$$

$$+ \sum_i \sum_{p \in P_i} \sum_j \sum_t \sum_s \left(H_i^1 h_p^1 + H_i^2 h_p^2 + m_{ipj}\right) prob_s\, x_{ipts} \quad (5.1)$$

$$+ \omega_1 \sum_j \sum_k \sum_t \sum_s f_{jkts}^{+} + \omega_2 \sum_i \sum_t \sum_s prob_s\, e_{its}^{+}$$

Subject to:

$$n_{jkt} = \begin{cases} a_{jkt} & if \quad t = 1 \\ n_{jk(t-1)} + a_{jkt} - b_{jkt} + u_{jkt} - v_{jkt} & if \quad t > 1 \end{cases} \quad \forall j, \forall k, \quad (5.2)$$

$$\sum_k u_{jkt} = \sum_k v_{jkt} \quad \forall j, \forall t \quad (5.3)$$

$$LM \leq \sum_j n_{jkt} \leq UM \quad \forall k, \forall t \quad (5.4)$$

$$\sum_i \sum_{p \in P_i} q_{ipjk} x_{ipts} - f_{jkts}^{+} + f_{jkts}^{-} = n_{jkt}\, C_j \quad \forall j, \forall k, \forall t, \forall s \quad (5.5)$$

$$\sum_{p \in P_i} x_{ipts} - e_{its}^{+} + e_{its}^{-} = D_{its} \quad \forall i, \forall t, \forall s \quad (5.6)$$

The overall objective of the robust multi-period part family/cell formation problem is to minimize the total system cost with respect to discrete demand scenarios where the probability of occurrence of a scenario ($prob_s$) is known. The total system cost in the objective function (5.1) consists of four terms. The first term in the objective function (5.1) minimizes the sum of purchasing, discarding and relocating costs of overall machines, cells, and time period. Calculating purchasing and discarding costs is done by multiplying the number purchased a_{jkt} or discarded b_{jkt} of each machine type j in cell k during time period t by the respective unit cost (c_j or c_j') and sum of these over j, k, and t. For the relocating cost, every machine is relocated out of one cell v_{jkt} and into another cell u_{jkt}. It is sufficient to multiply the sum of the u_{jkt} variables by the unit cost of relocation R_j. The total cost of purchasing, discarding and relocating machines is $\sum_j \sum_k \sum_t \left(c_j a_{jkt} + c'_j b_{jkt} + R_j u_{jkt}\right)$.

The second term in the objective function (5.1) minimizes the inter-cell/intra-cell material handling, and manufacturing costs. These are summed overall parts i and part routings p in each multi-commodity flow network for each time period. H_i^1 is the unit handling cost per inter-cell transfer of part i, while H_i^2 is the unit handling cost per intra-cell transfer of part i. The total number of inter-cell and intra-cell transfers in path $p \in P_i$ may be pre-computed based on the sequence of cells visited by the path. Similarly, the manufacturing cost per unit m_{ipj} on path p, may be calculated by summing the production costs on each of the machine types visited by the routings. Since x_{ipts} is the flow of part i using path p in time period t under scenario s, where s represents a discrete demand scenario with probability of occurrence $prob_s$. The total cost of inter-cell/intra-cell material handling, and manufacturing (the second term in the objective function) is $\sum_i \sum_{p \in P_i} \sum_j \sum_t \sum_s \left(H_i^1 h_p^1 + H_i^2 h_p^2 + m_{ipj}\right) prob_s \, x_{ipts}$.

The third and fourth terms in the objective function (5.1) introduces the trade-offs for model robustness. Two goal programming weights ω_1 and ω_2 are used for this reason. In the third term $\omega_1 \sum_j \sum_k \sum_t \sum_s f_{jkts}^+$, the penalty coefficient ω_1 for the shortfall in machine capacity is multiplied by the sum of error variable f_{jkts}^+ to measure the excess in machine capacity of machine j in cell k during period t under scenario s. In the fourth term $\omega_2 \sum_i \sum_t \sum_s prob_s \, e_{its}^+$, the penalty coefficient ω_2 for the shortage in the demand is multiplied by the sum of error variable e_{its}^+ to measure the shortfall in demand for part i during period t under scenario s with probability of occurrence $prob_s$.

The constraints in the model are (5.2) to (5.6), while constraint sets (5.2) and (5.3) are machine balance constraints. Constraint (5.2) ensures that the number of machines in a cell in the first time period $(t = 1)$ is equal to the number of machines purchased a_{jkt}. Otherwise, the number of machines in a cell during time period t >1 is equal to the number of machines in time period $(t - 1)$, plus the number of machines purchased a_{jkt}, minus the number of machines discarded b_{jkt}, plus the number of machines relocated u_{jkt} into the cell, and minus the number of machines relocated out of the cell v_{jkt}. Constraint (5.3) ensures that the total number of machines of type j relocated (i.e., moved into cell k) during time period t must be equal to the number of machines of type j relocated (i.e., moved out of cell k) in period t.

Constraint (5.4) limits the number of machines in each cell k during each time period t based on lower and upper bounds. Constraint (5.5) is the capacity constraint (machine availability) in the robust model. The sum of processing time per unit for part i on machine type j in cell k when routed through path p multiplied by the number of parts of type i routed through path p in period t under scenario s.

For every machine availability constraint, we add two error variables f_{jkts}^{+} and f_{jkts}^{-} to measure, respectively, the shortfall or excess in machine capacity for scenario s. The total processing time on machine type j, of which several might exist, in cell k during time period t over all part routings $p \in P_i$ is $\sum_i \sum_{p \in P_i} q_{ipjk} x_{ipts} - f_{jkts}^{+} + f_{jkts}^{-}$. This has to be equal to the time availability of machine type j in cell k during time period t, i.e., $n_{jkt} C_j$. Constraint (5.6) is the demand constraint in the robust model. The constraint ensures that the sum of production $\sum_{p \in P_i} x_{ipts} - e_{its}^{+} + e_{its}^{-}$ of part i routed through path p during period t under scenario s should be equal to the demand for that part D_{its}. For the demand constraint, we added two error variables e_{its}^{+} and e_{its}^{-} to measure, respectively, the shortfall or excess in demand for scenario s.

There are three cases that can be solved:

1. Solve the multi-period cell formation problem (model MPCFP) with $\overline{D}_{it} = \sum_s prob_s \, D_{its}$. This gives us an optimal design for average demand. Then solve s independent MPCF problems (with demands in scenario s) while fixing the design variables at the value indicated by the optimal design for average demand. Let Z_s be the values of the objective solutions to these problems. The expected value of the optimal design for average demand is $\hat{Z} = \sum_s prob_s \, Z_s$.
2. Solve s independent MPCF problems (with demands in scenario s) with no restrictions

on the design variables. Let Z'_s be the values of the objective solutions to these problems. The expected value of the objective function with perfect information (EVWPI) is $E(Z) = \sum_s prob_s Z'_s$. Therefore, the expected value of perfect information, $EVPI = \hat{Z}$-$E(Z)$.

3. Solve the robust optimization problem (RMPCFP).
4. Vary robust parameters to see trade-offs.

5.3 Toolkit for Solving the Optimization Problem (RMPCFP)

As mentioned in Chapter 3, a comprehensive toolkit for developing optimization-based analytical decision support applications was used during the research. The developed mathematical model for solving the RMPCFP was coded using the IBM CPLEX Optimization Studio (version 12.5). (APPENDIX 4) provide the used CPLEX Studio code and (APPENDIX 5) Python codes (for path generation).

5.4 Numerical Example and Computational Results for Solving (RMPCFP)

The applicability of the proposed robust optimization framework is illustrated through an example from the literature. The problem is selected from the literature to evaluate the robustness of the proposed model for solving the RMPCFP.

The majority of cellular manufacturing systems models in the literature are based on deterministic product demand. However, the illustrative example adopted from (Cao & Chen, 2005) considers product demands expressed through a number of probabilistic scenarios. We show how we apply the RMPCFP to this model to provide a general perspective on the applicability of robust optimization in the context of the work by (Mulvey & Vanderbei, 1995)

5.4.1 The Illustrative Example

The (Cao & Chen) illustrative example is used to demonstrate the formation of the manufacturing cells in reaching a compromise between system configuration cost and expected material handling cost in a manufacturing environment with production demands expressed in a number of probabilistic scenarios. In all three example problems in the paper, 4 machines types are placed in 2 manufacturing cells to manufacture 5 different parts. The inter-cell unit product movement cost is 0.25, the upper limit for cells is 8 machines, and the lower limit is 3 machines. Data for machine part operation and product demand are given in Tables 5-1 and 5-2, respectively.

Table 5-1: Machine part operation data

Machine type	Machine cost (Example 1)	Machine cost (Example 2)	Machine cost (Example 3)	Capacity requirement/ unit product				
				Part 1	**Part 2**	**Part 3**	**Part 4**	**Part 5**
1	6	15	15	0.02	0	0.014	0.026	0
2	6	15	15	0.024	0.028	0.016	0	0.024
3	6	15	15	0.01	0	0.01	0	0.01
4	6	15	15	0	0.02	0	0.016	0

Table 5-2: Product demand

Scenario	Probability (Example 1)	Probability (Example 2)	Probability (Example 3)	Product demand				
				Part 1	**Part 2**	**Part 3**	**Part 4**	**Part 5**
1	0.333	0.333	0.75	70	30	20	70	80
2	0.333	0.333	0.125	70	50	50	50	50
3	0.333	0.333	0.125	70	70	70	30	30

The unit costs to move the parts between cells as well as the machines' upper and lower limits for all cells are assumed to be constant over the three scenarios. The following cell capacity constraints were placed on the CMS design:

- The machines are to be grouped into two relatively independent cells.
- The machines' lower limit would be three machines in each cell, and the upper limit would be eight.
- Each part has only one operation sequence, implying that the manufacturing sequences is pre-defined.

5.4.2 Non-robust results

Although the illustrative case study in (Cao & Chen) has three example problems, we will discuss only example 2. In this example, the machine cost is 15, the product demand has three scenarios, and the scenario occurrence probability is 0.333 for each of the three scenarios.

While the Cao and Chen paper claims to be a robust optimization approach to solving the MPCFP, they actually solve the problem based on the expected value of demands. In order to understand the results presented in the paper, we will solve the example using our proposed model for the MPCFP. First, we solve the example based on the expected value of product demand for the 3 scenarios, and then we solve the 3 scenarios independently.

Tables 5-3 and 5-4 present the (Cao & Chen) solution to example 2. Table 5-3 shows the units of different machines placed in the two cells. Two units of machine types 1, 2 and 4, and one unit of machine type 3 are placed in cell 1. In cell 2, there are two units of machine type 1, four units of machine type 2, one unit of machine type 3, and zero units of machine type 4.

Table 5-3: Units of machines in different cells for example 2 in Cao & Chen

Cell	Machine 1	Machine 2	Machine 3	Machine 4	Number of machines
1	2	2	1	2	7
2	2	4	1	0	7

Table 5-4 shows the inter-cell movement in all scenarios. For example, for scenario 1 demand, all operations of parts 4 and 5 are processed in cell 1 and all operations of parts 1 and 3 are processed in cell 2. However, part 2 is processed in cells 1 and 2. We notice that inter-cell movement occurs in all scenarios. In scenarios 1 and 2, inter-cell movement occurs in processing part 2 in cells 1 and 2, while in scenario 3, inter-cell movement occurs in processing part 3 in cells 1 and 2. The three scenarios have the same occurrence probabilities (0.333), the unit machine cost is 15 for all machines, and inter-cell movement is 0.25. When the inter-cell cost is low, the system is designed with fewer machines and more transfer between cells. The solution uses 14 machines.

Table 5-4: Part-cell allocation for example 2

	Cell	Scenario 1	Scenario 2	Scenario 3
Example 2	1	P2, P4, P5	P2, P3, P4, P5	P2, P3, P4
	2	P1, P2, P3	P1, P2	P1, P3, P5

Table 5-5 illustrates a comparison and cost analysis between the solution obtained by (Cao & Chen) and the solution obtained by using the RMPCFP. The first column in the table shows the result obtained by Cao and Chen for machine cost, material handling cost, and total system cost. The second column show the results obtained by solving example 2 based on the expected value of demand for the 3 scenarios. The third, fourth, and fifth columns show the results obtained by solving the 3 scenarios independently in example 2. From this, we make the following observations:

- In the Cao and Chen paper, the results show that the machine acquisition cost is \$210. We solved example 2 in the Cao and Chen paper by using our model (RMPCFP) based on the expected value of demand, and get the same solution. However, we believe that the material handling cost for that solution should be

12.5, not 9.9 as reported in the Cao and Chen paper). The discrepancy seems to be the transferred units of part 2 in scenario 1.

- Using our proposed model (MPCFP), we solve each scenario independently. The machine acquisition cost for scenario 1 is $195 and for subsequent scenarios 2 and 3 is $210. The number of machines required for scenario 1 is 13, while the number of machines required for scenarios 2 and 3 is 14.

Table 5-5: Comparison and cost analysis between (Cao & Chen) and RMPCFP solutions

	Cao & Chen	Expected Demand	Scenario (1)	Scenario (2)	Scenario (3)
Machine cost	210	210	195	210	210
Material handling cost	~~9.9~~	12.5	2.5	4.167	5.833
Actual material Handling cost	12.5				
Total system cost	~~219.9~~	222.5	197.5	214.167	215.833
Total actual system cost	**222.5**				

In summary, in a scenario-based robust optimization for the MPCFP, one begins by solving the problem using the expected demand in each period and also solves each scenario independently. Since the solution for these problems can be arrived at in 20 seconds or less, it is not inconceivable to solve thousands of scenarios in practical cases, with parallel and/or multi-core computing capabilities.

5.4.3 Solving the robust optimization problem (RCFP)

The purpose of this section is, first, to show, using the example of Cao and Chen and the literature, how the proposed robust formulation can be used in the design of central back up cellular manufacturing systems, and secondly, to provide an assessment of the performance of the proposed robust model.

We vary the robust parameters (penalty coefficients) to observe the trade-offs. In the MPCFP, depending on the scenario, there is either unmet demand or excessive capacity. Two cases will be reviewed. In case one, we fix ω_2 the penalty coefficient for shortage in demand and vary (ω_1) the penalty coefficient for shortfall in machine capacity. However, in case two, we do the opposite – fixing ω_1 while varying ω_2.

The error variables in our model measure the excess or shortfall in machine capacity for scenario s and the excess or shortfall in demand for scenario s. For example, $f_{jkts}^{+}$is the error variable to measure the shortfall in machine capacity for scenario s, and f_{jkts}^{-} is the error variable to measure the excess in machine capacity for scenario s. As well, e_{its}^{+} is the error variable to measure the shortfall in demand for scenario s, and e_{its}^{-} is the error variable to measure the excess in demand for scenario s.

5.4.3.1 Case One:

In this case, the robust model is used to find the optimal solution, the penalty coefficient ω_2 is fixed with a value of 100, and the ω_1 value is varied as (1, 5, 10, 100, 1000). The results are shown in Table 5-6.

The second row in Table 5-6 shows the following results: the value of the objective function is 251.4, the penalty cost is 161.4, and the total number of machines is 6 based on a machine cost per unit of 15. The last three cells in the second row show the error variables, Fplus, Eminus and Fminus, Fplus is the error variable to measure the shortfall in machine capacity, and the value of 161.4 in row 1 shows that we have shortfall in machine capacity. However, Eminus is the error variable to measure the excess in demand (here, the value is zero) and Fminus is the error variable to measure the excess in machine capacity (again, the value is zero).

Table 5-6: Robust results for case one

Omega 1	Omega 2	Objective Function	Penalty Cost	Machine Cost	Number of Machines	Fplus	Eminus	Fminus
1	100	251.4	161.4	90	6	161.4	0	0
5	100	897	732	165	11	146.4	0	0
10	100	1554	1314	240	16	131.4	0	0
100	100	13380	13140	240	16	131.4	0	0
1000	100	18864	18624	240	16	0	186.242	1.214

Figure 5-1 represents the results for case one and shows the relation between the value of ω_1 (the penalty cost for shortfall in machine capacity) and the cost of machine acquisition to process the demand. The value of ω_2 is fixed at 100; as ω_1 increases (as shown in the logarithmic scale of the *x*-axis of Figure 5-1), the machine cost also increases. The model compensates for the higher cost of machine capacity shortfall by increasing the number of machines and reducing the expected cost of the shortfall in capacity.

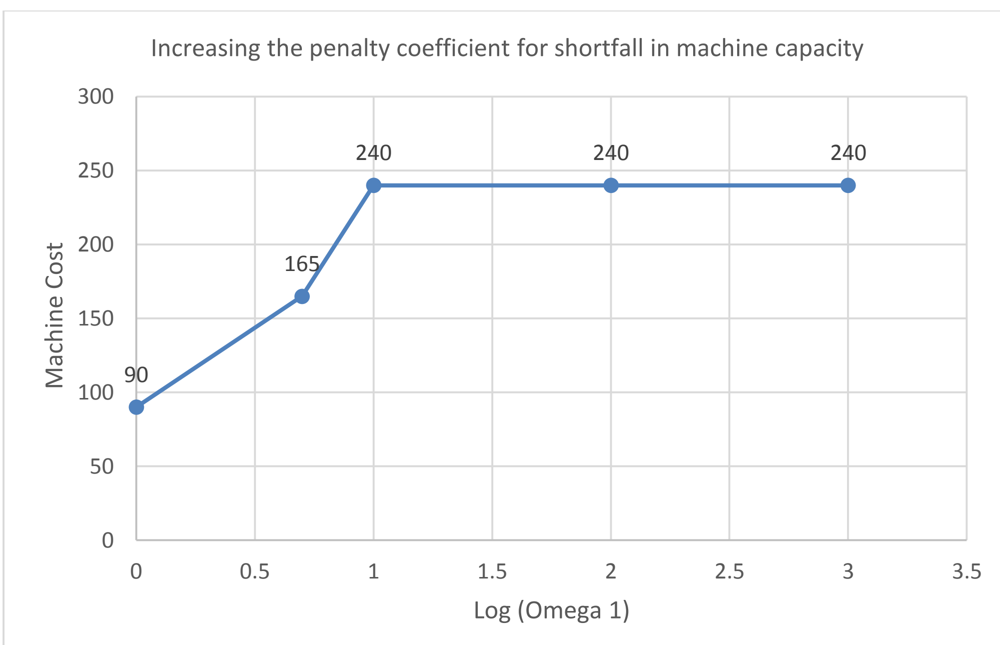


Figure 5-1: The result of increased penalty for shortfall in machine capacity

5.4.3.2 Case Two:

In this case, the robust model is used to find the optimal solution with the penalty coefficients ω_1 fixed at 100 and ω_2, varying as follows: (1, 2.85, 3, 3.05, 5, 10, 100, 1000). For example, we start our experiment with ω_1= 100 and ω_2= 1, using our robust model to solve the problem and examine the obtained results. The last three cells in the second row show the error variables Fplus, Eminus and Fminus.

Table 5-7: Robust result for case two

Omega 1	Omega 2	Objective Function	Penalty Cost	Machine Cost	Number of Machines	Fplus	Eminus	Fminus
100	1	327.54	237.54	90	6	0	237.54	1.566
100	2.85	765.87	630.87	135	9	0	221.358	0.676
100	3	800	635	165	11	0	211.534	0.976
100	3.05	804	609	195	13	0	199.551	0.216
100	5	1171.3	931.3	240	16	0	186.242	1.214
100	10	2103	1863	240	16	0	186.242	1.214
100	100	13380	13140	240	16	131.4	0	0
100	1000	13380	13140	240	16	131.4	0	0

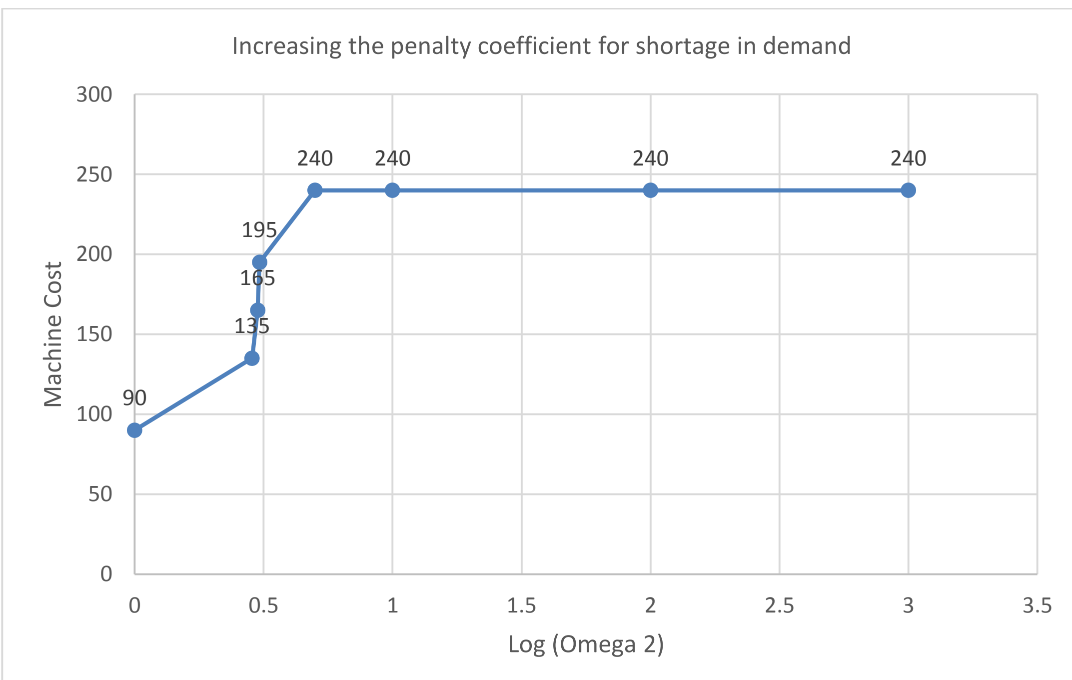


Figure 5-2: The result of increased penalty for shortage in demand

As the value of ω_2 increases (as also shown in the logarithmic scale in Figure 5-2), the cost of unmet demand decreases. The model compensates by purchasing more machines to reduce the error variable (E minus) that measures the excess in demand. The F minus values are residual values resulting from the constraint that the number of machines be integer. For very high values of ω_2 $(\text{e.g.}\ 100\ \text{and}\ 100)$ the model makes sure that demand is always met. However, in doing so, it chooses to use a system with 16 machines in a system with shortage in capacity (Fplus) rather than perhaps more machines and no shortage in either demand or capacity.

In Figure 5-3, the bar chart demonstrates a cost comparison between non-robust and robust solutions for example 2 in the Cao and Chen paper. The first bar shows the machine cost for a non-robust solution based on the expected value of product demand, and the second, third, and fourth bars show the machine cost for the three scenarios, respectively.

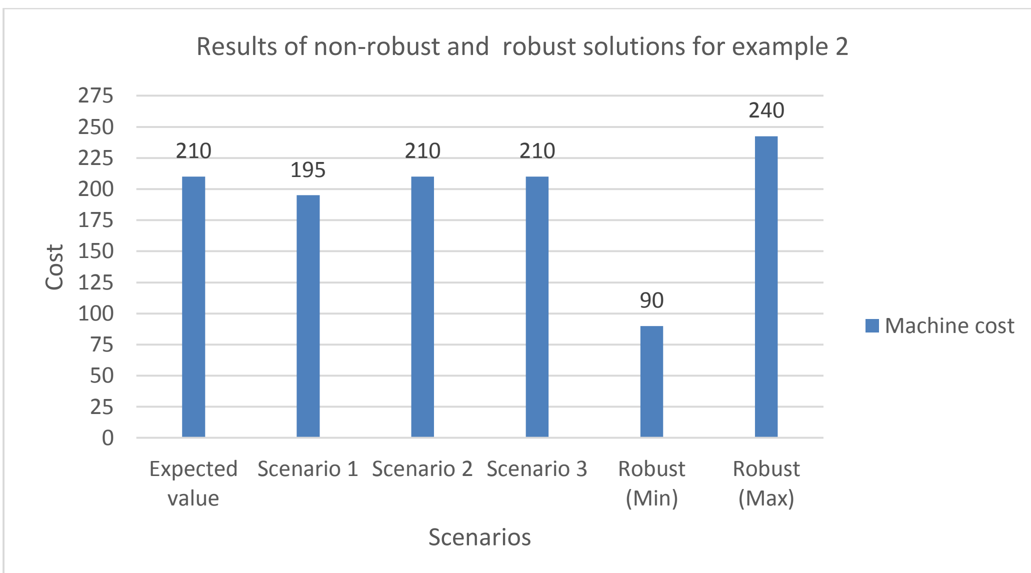


Figure 5-3: Cost comparison between non-robust and robust solutions.

The computational results of the robust solutions specify the limit of the most and least expensive cell designs. The limit of the least expensive cell design is known as the robust minimum solution, while the limit of the most expensive cell design is known as the robust maximum solution.

The fifth bar shows the robust minimum solution cost. This solution represents the cell configurations required to process the minimum demand for each part across the various scenarios. Naturally, this solution will have a shortage of machine capacity for any scenario with part demand greater than the minimum and in such cases, unmet demand will occur. The sixth bar shows the robust maximum machine cost. This solution represents the cell configurations required to process the maximum demand for each part across the various scenarios. Naturally, this solution results in excess machine capacity and therefore no unmet demand will occur, regardless of scenario.

For a given scenario, we can optimize cell design between the robust minimum solution and the robust maximum solution, depending on the penalty coefficients used by the designer.

In conclusion, a decision-maker can evaluate a set of demand scenarios and solve the RMPCFP. The final choice of cell configuration rests with the designer and is a compromise between having a shortage in demand, which often results in poorer customer satisfaction or higher production costs due to subcontracting, and excessive machine capacity, which results in higher investment costs.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

This thesis represents a new design approach to address the layout of manufacturing systems under uncertainties. In particular, it looks at the multi-period cell formation problem (MPCFP) and the broad question of what the design alternatives are (GT, fractal cell, and CBCMS). It also outlines a methodology for robust optimization within the MPCFP setting.

The contributions of this work may be summarized as follows:

1. We develop an efficient multi-commodity-based formulation for the MPCFP that can be solved in reasonable time to allow the designer to explore various alternatives for cell formation.
2. We introduce the central backup cellular manufacturing system (CBCMS). This contribution has not only practical design implications but also a theoretical one. The continuum between fractal and GT design alternatives has not been understood and this thesis addresses that gap by introducing the CBCMS.
3. The multi-commodity-based formulation for the MPCFP can be used with different parameter settings to look at fractal cell, CBCMS, and GT cell designs. This has helped develop insights into the performance of the fractal, cellular, and CBCMS layout designs.
4. We develop a robust extension of the MPCFP called the RMPCFP. This extension helps show the design tradeoff between excessive machine capacity and unmet demand. While we do not prescribe a solution for the designer, different alternatives may be generated by the use of the penalty coefficients in the RMPCFP model to understand the trade-offs.

6.2 Recommendation for Future Research

There are several avenues that are recommended as future research to further utilize or improve on the proposed methodology of robust cell formation and CBCMS.

6.2.1 The Spatial Cell and Plant Layout Design Problem

In this thesis, a multi-period cell formation under uncertainty (non-spatial) model (MPCFP) is presented for the dynamic composition of cells. The next step is to determine the actual layout of the cells and the machines within the cells, which are both critically important elements in the design of a facility's layout. In our research, the layout is broken down into cells, with the optimal allocations of machines to cells. The model could be formulated to locate machines within cells as well as cells with respect to each other to minimize the total flow distance of products in the period layouts suggested by non-spatial design.

We recommend using the sequence pair formulation to model the problem and a meta-heuristic framework such as simulated annealing or the GA to solve realistic size instances. The sequence pair formulation cuts down on the number of zero-one variables for the position of the machines and thus cuts down on the search tree during the optimization process.

6.2.2 Using Simulation to Evaluate the Fractal, CBCMS, and GT Designs

Simulation is considered one of the most important tools for analyzing performance in the manufacturing and service industries. In the scope of facilities layout design, simulation can be used to understand the behavior of a facility. Simulation allows the comparison of different choices and studies several scenarios in order to select the most suitable facility layout system. For evaluating the CBCMS design by comparing it to other designs (GT and fractal), we recommend using simulation. Designing a new layout or modifying a layout of an existing facility is a design decision. Therefore, it is crucial to investigate the proposed layout design before making a choice.

The advantage of using simulation is to provide a depth of understanding of how the three layout systems behave. We believe that running simulation experiments may help researchers determine the flexibility and efficiency of a design. Therefore, simulation models of the three layout types (GT, fractal and CBCMS) are essential to understand each layout and evaluate its benefits.

6.2.3 Product Price and /Lead Time Considerations

One important research challenge is investigating how a manufacturing system should be designed with regard to varying product price and lead-time considerations. When demand or lead-time is certain, it may be desirable to use product or product-based layouts such as GT. However, when the nature of the product varies, so does the price and lead time. A solution to this dilemma may be to use process, CBCMS or fractal layout design. Despite its importance, this compelling issue of varying product price and lead time has not yet been investigated in the literature.

References

Adil, G. K., Rajamani, D., & Strong, D. (1996). Cell formation considering alternate routings. *International Journal of Production Research, 34*(5), 1361.

Aryanezhad, M. B., Deljoo, V., & Al-e-hashem, S. M. J. Mirzapour. (2009). Dynamic cell formation and the worker assignment problem: A new model. *International Journal of Advanced Manufacturing Technology, 41*(3), 329-342. doi:10.1007/s00170-008-1479-4

Askin, R. G., & Standridge, C. R. (1993). *Modeling and analysis of manufacturing systems* Wiley.

Askin, R. G., Ciarallo, F. W., & Lundgren, N. H. (1999). An empirical evaluation of holonic and fractal layouts. *International Journal of Production Research, 37*(5), 961.

Askin, R., Selim, H., & Vakharia, A. (1997). A methodology for designing flexible cellular manufacturing systems. *IIE Transactions, 29*(7), 599-610. doi:10.1023/A:1018505631215

Balakrishnan, J., Jacobs, R. F., & Venkataramanan, M. A. (1992). Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research, 57*(2), 280-286.

Balakrishnan, J., & Cheng, C. H. (1998). Dynamic layout algorithms: A state-of-the-art survey. *Omega, 26*(4), 507-521. doi:10.1016/S0305-0483(97)00078-9

Balakrishnan, J., & Cheng, C. H. (2000). Genetic search and the dynamic layout problem. *Computers & Operations Research, 27*(6), 587-593. doi:10.1016/S0305-0548(99)00052-0

Balakrishnan, J., & Cheng, C. H. (2005). Dynamic cellular manufacturing under multiperiod planning horizons. *Journal of Manufacturing Technology Management, 16*(5/6), 516-530.

Balakrishnan, J., & Cheng, C. H. (2007). Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions. *European Journal of Operational Research, 177*(1), 281-309. doi:10.1016/j.ejor.2005.08.027

Balakrishnan, J., Chun, H. C., & Conway, D. G. (2000). An improved pair-wise exchange heuristic for the dynamic plant layout problem. *International Journal of Production Research, 38*(13), 3067-3077. doi:10.1080/00207540050117440

Balakrishnan, J., & Hung Cheng, C. (2009). The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty. *Omega, 37*(1), 165-177. doi:DOI: 10.1016/j.omega.2006.11.005

Benjaafar, S., Heragu, S. S., & Irani, S. A. (2002). Next generation factory layouts: Research challenges and recent progress. *Interfaces, 32*(6), pp. 58-76.

Benjaafar, S., & Sheikhzadeh, M. (2000). Design of flexible plant layouts. *IIE Transactions, 32*(4), 309-322.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization* (First Edition) Princeton University Press.

Ben-Tal, A., & Nemirovski, A. (1997). Robust truss topology design via semidefinite programming. *SIAM Journal on Optimization, 7*(4), 991-1016. doi:10.1137/S1052623495291951

Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming, 88*(3), 411.

Ben-Tal, A., & Nemirovski, A. (2002). Robust optimization – methodology and applications. *Mathematical Programming, 92*(3), 453.

Ben-Tal, A., & Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters, 25*(1), 1-13. doi:10.1016/S0167-6377(99)00016-4

Bertsimas, D., & Brown, D. B. (2009). Constructing uncertainty sets for robust linear optimization. *Operations Research, 57*(6), 1483-1495. doi:10.1287/opre.1080.0646

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review, 53*(3), 464-38. doi:http://dx.doi.org/10.1137/080734510

Bertsimas, D., Nohadani, O., & Teo, K. M. (2010). Nonconvex robust optimization for problems with constraints. *Informs Journal on Computing, 22*(1), 44-58. doi:10.1287/ijoc.1090.0319

Bertsimas, D., Pachamanova, D., & Sim, M. (2004). Robust linear optimization under general norms. *Operations Research Letters, 32*(6), 510-516. doi:10.1016/j.orl.2003.12.007

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35-53. Retrieved from http://www.jstor.org/stable/30036559

Beyer, H., & Sendhoff, B. (2007). Robust optimization – A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering, 196*(33–34), 3190-3218. doi:10.1016/j.cma.2007.03.003

Black, J. (1983). Cellular manufacturing systems reduce setup time, make small lot production economical. *Industrial Engineering, 15*, 36–48.

Boctor, F. F. (1991). A linear formulation of the machine-part cell formation problem. *International Journal of Production Research, 29*(2), 343.

Braglia, M., Zanoni, S., & Zavanella, L. (2005a). Robust versus stable layout design in stochastic environments. *Production Planning & Control, 16*(1), 71-80. doi:10.1080/09537280310001654835

Braglia, M., Zanoni, S., & Zavanella, L. (2005b). Layout design in dynamic environments: Analytical issues. *International Transactions in Operational Research, 12*(1), 1-19.

Bredström, D., Flisberg, P., & Rönnqvist, M. (2013). A new method for robustness in rolling horizon planning. *International Journal of Production Economics, 143*(1), 41-52. doi:10.1016/j.ijpe.2011.02.008

Cao, D., & Chen, M. (2005). A robust cell formation approach for varying product demands. *International Journal of Production Research, 43*(8), 1587-1605. doi:10.1080/00207540412331327754

Caux, C., Bruniaux, R., & Pierreval, H. (2000). Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics, 64*(1–3), 279-284. doi:http://dx.doi.org/10.1016/S0925-5273(99)00065-1

Chan, H. M., & Milner, D. A. (1982). Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems, 1*(1), 65-75. doi:10.1016/S0278-6125(82)80068-X

Chen, M. (1998). A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. *Annals of Operations Research, 77*(1-4), 109-128.

Conway, D. G., & Venkataramanan, M. A. (1994). Genetic search and the dynamic facility layout problem. *Computers & Operations Research, 21*(8), 955-960. doi:10.1016/0305-0548(94)90023-X

Deljoo, V., Mirzapour Al-e-hashem, S. M. J., Deljoo, F., & Aryanezhad, M. B. (2010). Using genetic algorithm to solve dynamic cell formation problem. *Applied Mathematical Modelling, 34*(4), 1078-1092. doi:10.1016/j.apm.2009.07.019

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control, 31*(2), 255-267. doi:DOI: 10.1016/j.arcontrol.2007.04.001

Drolet, J., Abdulnour, G., & Rheault, M. (1996). The cellular manufacturing evolution. *Computers & Industrial Engineering, 31*(1-2), 139-142. doi:DOI: 10.1016/0360-8352(96)00097-6

Dunker, T., Radons, G., & Westkamper, E. (2005). Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research, 165*(1), 55-69.

El Ghaoui, L., & Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications, 18*(4), 1035-1064. doi:10.1137/S0895479896298130

El Ghaoui, L., Oustry, F., & Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization, 9*(1), 33-52. doi:10.1137/S1052623496305717

Elbenani, B., & Ferland, J. A. (2012). An exact method for solving the manufacturing cell formation problem. *International Journal of Production Research, 50*(15), 4038-4045.

Erel, E., Ghosh, J. B., & Simon, J. T. (2003). New heuristic for the dynamic layout problem. *Journal of the Operational Research Society, 54*(12), 1275-1282.

Ghotboddini, M. M., Rabbani, M., & Rahimian, H. (2011). A comprehensive dynamic cell formation design: Benders' decomposition approach. *Expert Systems with Applications, 38*(3), 2478-2488. doi:10.1016/j.eswa.2010.08.037

Gunn, E., & Venkatadri, U. (2008). Strengthening the sequence-pair formulation for the facility layout problem. *10th International Material Handling Research Colloquim (IMHRC),* Dortmund, Germany. 662-669.

Harhalakis, G., Ioannou, G., Minis, I., & Nagi, R. (1994). Manufacturing cell formation under random product demand. *International Journal of Production Research, 32*(1), 47.

Heragu, S. S. (2008). *Facilities design* (Third Edition) CRC Press.

Heragu, S. S., & Gupta, Y. P. (1994). A heuristic for designing cellular manufacturing facilities. *International Journal of Production Research, 32*(1), 125.

Hopp, W. J., & Spearman, M. L. (2008). *Factory physics* (Third Edition). New York: McGraw-Hill/Irwin.

Houtzeel, A., & Brown, C. S. (1984). A management overview of group technology at works. *Society of Manufacturing Engineers, M*(1), 3-16.

Hu, T. C. (1963). Multi-commodity network flows. *Operations Research, 11*(3), 344-360. doi:10.1287/opre.11.3.344

Irani, S. A., Cavalier, T. M., & Cohen, P. H. (1993). Virtual manufacturing cells: Exploiting layout design and intercell flows for the machine sharing problem. *The International Journal of Production Research, 31*(4), 791-810.

Irani, S. A. (Ed.). (1999). *Handbook of cellular manufacturing systems* John Wiley & Sons.

Irani, S. A. (2013, June/July 2013). Job shop lean. *Gear Technology,* , 20-26.

Jayakumar, V., & Raju, R. (2010). An adaptive cellular manufacturing system design with routing flexibility and dynamic system reconfiguration. *European Journal of Scientific Research, 47*(4), 595-611.

Jayaswal, S., & Adil, G. K. (2004). Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings. *International Journal of Production Research, 42*(12), 2419-2433. doi:10.1080/00207540310001652914

Kaku, B. K., & Mazzola, J. B. (1997). A tabu search heuristic for the dynamic plant layout problem. [A tabu search heuristic for the dynamic plant layout problem] *INFORMS Journal on Computing, 9*(4), 374-384.

Kalpakjian, S., & Schmid, S. (2010). *Manufacturing Engineering and Technology* (Sixth Edition.) Prentice Hall Professional Technical Ref.

King, J. R., & Nakornchai, V. (1982). Machine--component group formation in group technology: Review and extension. *International Journal of Production Research, 20*(2), 117.

Kochhar, J. S., & Heragu, S. S. (1999). Facility layout design in a changing environment. *International Journal of Production Research, 37*(11), 2429-2446.

Kochikar, V., & Narendran, T. (1998). Logical cell formation in FMS, using flexibility-based criteria. *International Journal of Flexible Manufacturing Systems, 10*(2), 163-181. doi:10.1023/A:1008049515026

Krishnan, K. K., Cheraghi, S. H., & Nayak, C. N. (2008). Facility layout design for multiple production scenarios in a dynamic environment. *International Journal of Industrial and Systems Engineering, 3*(2), 105-133.

Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21 st century. *Journal of Intelligent Manufacturing, 18*, 273-284.

Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operational Research, 29*(3), 229-251.

Lahmar, M., & Benjaafar, S. (2005). Design of distributed layouts. *IIE Transactions (Institute of Industrial Engineers), 37*(4), 303-318.

Liu, C., Yin, Y., Yasuda, K., & Lian, J. (2010). A heuristic algorithm for cell formation problems with consideration of multiple production factors. *International Journal of Advanced Manufacturing Technology, 46*(9-12), 1201-1213. doi:10.1007/s00170-009-2170-0

Mahdavi, I., Paydar, M. M., Solimanpur, M., & Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications, 36*(3, Part 2), 6598-6604. doi:10.1016/j.eswa.2008.07.054

McKendall Jr., A. R., & Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research, 201*(1), 171-182.

Meller, R. D., Chen, W., & Sherali, H. D. (2007). Applying the sequence-pair representation to optimal facility layout designs. *Operations Research Letters, 35*(5), 651-659.

Meller, R. D., & Gau, K. Y. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems, 15*(5), 351-366.

Meng, G., Heragu, S. S., & Zijm, H. (2004). Reconfigurable layout problem. *International Journal of Production Research, 42*(22), 4709-4729.

Montreuil, B., & Venkatadri, U. (1991). Strategic interpolative design of dynamic manufacturing systems layouts. *Management Science, 37*(6), 682-694.

Montreuil, B., Venkatadri, U., & Rardin, R. L. (1999). Fractal layout organization for job shop environments. *International Journal of Production Research, 37*(3), 501-521.

Mulvey, S. M., & Vanderbei, R. J. (1995). Robust optimization of large-scale systems. *Operations Research, 43*(2), 264.

Murata, H., Fujiyoshi, K., Nakatake, S., & Kajitani, Y. (1996). VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15*(12), 1518-1524.

Nagi, R., Harhalakis, G., & Proth, J. (1990). Multiple routeings and capacity considerations in group technology applications. *International Journal of Production Research, 28*(12), 2243.

Nicol, L. M., & Hollier, R. H. (1983). Plant layout in practice. *Material Flow, 1*(3), 177-188.

Norman, B. A., & Smith, A. E. (2006). A continuous approach to considering uncertainty in facility design. *Computers & Operations Research, 33*(6), 1760-1775. doi:10.1016/j.cor.2004.11.017

Nsakanda, A. L., Diaby, M., & Price, W. L. (2010). A price-directed decomposition approach for solving large-scale capacitated part-routing problems. *International Journal of Production Research, 48*(14), 4273-4295. doi:10.1080/00207540802680526

Papaioannou, G., & Wilson, J. M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and directions for future research. *European Journal of Operational Research, 206*, 509-521.

Pullen, R. D. (1976). A survey of cellular manufacturing cells. *Production Engineer, 55*(9), 451-454. doi:10.1049/tpe:19760137

Ramabhatta, V., & Nagi, R. (1998). An integrated formulation of manufacturing cell formation with capacity planning and multiple routings. *Annals of Operations Research, 77*(1-4), 79-95.

Ribeiro, J. F. F. (2009). Manufacturing cells formation based on graph theory. Paper presented at the *Computers & Industrial Engineering, 2009. CIE 2009. International Conference On Computers and Industrial Engineering,* 658-662. doi:10.1109/ICCIE.2009.5223903

Romesburg, H. C. (1987). Cluster analysis for researchers: Lifetime learning publications, belmont, california, 1984, 334 pp. *Environmental Software, 2*(2), 107. doi:10.1016/0266-9838(87)90010-4

Rosenblatt, M. J. (1986). The dynamics of plant layout. *Management Science, 32*(1), 76.

Saad, S., & Lassila, A. (2004). Layout design in fractal organizations. *International Journal of Production Research, 42*(17), 3529-3550. doi:10.1080/00207540410001721790

Sankaran, S., & KasiIingam, R. G. (1990). An INTEGRATED approach to cell formation and part routing in group technology manufacturing systems. *Engineering Optimization, 16*(3), 235-245. doi:10.1080/03052159008941175

Saraswat, A. (2006). *Multi objective design for block layouts.* Dalhousie University).

Seifoddini, H. (1990). A probabilistic model for machine cell formation. *Journal of Manufacturing Systems, 9*(1), 69-75. doi:10.1016/0278-6125(90)90070-X

Seifoddini, H., & Wolfe, P. M. (1986). Application of the similarity coefficient method in group technology. *IIE Transactions, 18*(3), 271-277. doi:10.1080/07408178608974704

Selim, H. M., Askin, R. G., & Vakharia, A. J. (1998). Cell formation in group technology: Review, evaluation and directions for future research. *Computers & Industrial Engineering, 34*(1), 3-20. doi:10.1016/S0360-8352(97)00147-2

Sethi, A. K., & Sethi, S. P. (1990). Flexibility in manufacturing: A survey. *International Journal of Flexible Manufacturing Systems, 2*(4), 289-328.

Shin, M., Mun, J., & Jung, M. (2009). Self-evolution framework of manufacturing systems based on fractal organization. *Computers & Industrial Engineering, 56*(3), 1029-1039. doi:DOI: 10.1016/j.cie.2008.09.014

Sofianopoulou, S. (1999). Manufacturing cells design with alternative process plans and/or replicate machines. *International Journal of Production Research, 37*(3), 707.

Spiliopoulos, K., & Sofianopoulou, S. (2007). Manufacturing cell design with alternative routings in generalized group technology: Reducing the complexity of the solution space. *International Journal of Production Research, 45*(6), 1355-1367. doi:10.1080/00207540600719674

Suer, G. A., Huang, J., & Maddisetty, S. (2010). Design of dedicated, shared and remainder cells in a probabilistic demand environment. *International Journal of Production Research, 48*(19), 5613-5646. doi:10.1080/00207540903117865

Sule, D. (2009). *Manufacturing facilities location, planning, and design* (third edition ed.) Taylor.

Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Safaei, N., & Azaron, A. (2005). Solving a dynamic cell formation problem using metaheuristics. *Applied Mathematics and Computation, 170*(2), 761-780. doi:http://dx.doi.org/10.1016/j.amc.2004.12.021

Tharumarajah, A., Wells, A. J., & Nemes, L. (1996). Comparison of the bionic, fractal and holonic manufacturing system concepts. *International Journal of Computer Integrated Manufacturing, 9*(3), 217-226.

Tompkins, J. (2003). *Facilities planning* (Third Edition) Wiley.

Tunnukij, T., & Hicks, C. (2009). An enhanced grouping genetic algorithm for solving the cell formation problem. *International Journal of Production Research, 47*(7), 1989-2007. doi:10.1080/00207540701673457

Urban, T. L. (1998). Solution procedures for the dynamic facility layout problem. *Annals of Operations Research, 76*(1-4), 323-342.

Urban, T. (1993). A heuristic for the dynamic facility layout problem. *IIE Transactions, 25*(4), 57-63. doi:10.1080/07408179308964304

Vakharia, A. J. (1986). Methods of cell formation in group technology: A framework for evaluation. *Journal of Operations Management, 6*(3–4), 257-271. doi:10.1016/0272-6963(86)90002-1

Venkatadri, U., Rardin, R. L., & Montreuil, B. (1997). A design methodology for fractal layout organization. *IIE Transactions, 29*(10), 911-924.

Vila Gonçalves Filho, E., & José Tiberti, A. (2006a). A group genetic algorithm for the machine cell formation problem. *International Journal of Production Economics, 102*(1), 1-21. doi:10.1016/j.ijpe.2004.12.029

Vila Gonçalves Filho, E., & José Tiberti, A. (2006b). A group genetic algorithm for the machine cell formation problem. *International Journal of Production Economics, 102*(1), 1-21. doi:10.1016/j.ijpe.2004.12.029

Wang, J. (2003). Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm. *Automatica, 39*(9), 1607-1615. doi:10.1016/S0005-1098(03)00150-X

Wicks, E. M., & Reasor, R. J. (1999). Designing cellular manufacturing systems with dynamic part populations. *IIE Transactions, 31*(1), 11-20.

Yang, T., & Peters, B. A. (1998). Flexible machine layout design for dynamic and uncertain production environments. *European Journal of Operational Research, 108*(1), 49-64. doi:DOI: 10.1016/S0377-2217(97)00220-8

Yin, Y., & Yasuda, K. (2005). Similarity coefficient methods applied to the cell formation problem: A comparative investigation. *Computers & Industrial Engineering, 48*(3), 471-489. doi:10.1016/j.cie.2003.01.001

APPENDIX 1 Wicks and Reasor Model

Model indices

i = Index of parts to be processed, i = 1, 2,…, N;

j = Index of machine types, j = 1, 2,…, M;

k = Index of cells, k = 1, 2,…, C;

l = Index of time periods for design, l = 1, 2,…, P

Parameters

D_{il}= Expected demand of part i in period l,

S_{il} = Number of operations for part i in period l.

O_{irl}= Machine time required by rth operation of part i in period l.

T_{ijl}= Processing time for part i on machine type j in period l.

M_j= Number of type j machines available at the start.

C_j= Capacity of machine type j.

P_{jl} = Cost of acquiring a unit of machine j in period l.

H_{il}= Inter-cell unit material handling cost for part i in period l.

R_{jl}= Relocating cost of machine type j in period l.

LM= Minimum number of machines per cell.

LP = Minimum number of parts per cell.

A = A large number.

Decision Variables

$$X_{ikl}=\begin{cases} 1, if\ part\ i\ is\ assigned\ to\ cell\ k\ in\ period\ l \\ 0, otherwise \end{cases}$$

$$y_{jkl}=\begin{cases} 1, if\ machine\ type\ j\ is\ assigned\ to\ cell\ k\ in\ period\ l \\ 0, otherwise \end{cases}$$

n_{jkl}= Number of type j machines assigned to cell k in period l (integer).

q_{il}= Number of inter cell transfers of part i in period l.

b_{jl}= Number of type j machines acquired at the beginning of period l

u_{jl}= Number of type j machines relocated between periods $l-1$ and l.

The objective function is to minimize the total costs of:

- Inter-cell transfers of parts

- Between period re-configuration of cells (Relocation of machines)
- Machines duplication

Minimize Z:

$$\sum_{l=1}^{P}\left[\sum_{i=1}^{N} H_{il}\, D_{il} q_{il} + \sum_{j=1}^{M_j} P_{jl} b_{jl} + \sum_{j=1}^{M_j} R_{jl} u_{jl}\right]$$

Definitional constraints on number of inter-cell transfers, machines purchased, and machines relocated:

$$q_{il} = \sum_{k=1}^{C} X_{ikl}\left[\sum_{r=1}^{S_{il}-1}\left(1 - yO_{(i,r,l)kl}\, yO_{(i,r+1,l)kl}\right)\right] \forall i, \forall l$$

$$b_{jl} = max\left[0, \sum_{k=1}^{c} n_{jkl} - M_j - \sum_{s=1}^{l-1} b_{js}\right] \forall j, \forall l$$

$$u_{jl} = \sum_{k=1}^{C} max\ \left[0, n_{jkl} - n_{jk}\right] - b_{jl}\ \forall j, \forall l$$

Part demand constraint:

$$\sum_{k=1}^{C} X_{ikl} = D_{il}\ \forall i, \forall l$$

Machine capacity constraints (for each cell and overall):

$$\sum_{i=1}^{N} D_{il} T_{ijl} X_{ikl} y_{jkl} \leq C_j n_{jkl}\ \forall i, \forall k, \forall l$$

$$\sum_{i=1}^{N} D_{il} T_{ijl} \leq C_j \sum_{k=1}^{C} n_{jkl}\ \ \forall i, \forall l$$

Minimum number of machines and parts in a cell:

$$\sum_{j=1}^{M} y_{jkl} \geq LM \; \forall k, \forall l$$

$$\sum_{i=1}^{N} X_{jkl} \geq LP \; \forall k, \forall l$$

The number of units of a machine type in a cell is zero unless it has been assigned to the cell:

$$n_{jkl} \leq Ay_{jkl} \quad \forall j, \forall k, \forall l$$

The Wicks and Reasor model is a non-linear integer programing model with cubic and quadratic terms in the first constraint set. As a result, it is very hard to solve the problem optimally. In order to solve the problem, the authors use a genetic algorithm based solution methodology.

APPENDIX 2 Python code

(Generating all possible routing sequences for Wicks and Reasor illustrative example in Chapter 4)

```
/*********************************************
 * Python (V.2.7)
 *********************************************/

#Combinations function
#This function picks all possible combinations from a list
def myCombinations(list):
    r=[[]]
    for x in list:
        t = []
        for y in x:
            for i in r:
                t.append(i+[y])
        r = t
    return r;

#This function gives us vertical slices in the network

def vert_Slices(s,seq_num):
    nodes = [];
    for i in s[seq_num]:
        inodes = [];
        for j in cells:
            inodes.append([i]+[j]);
        nodes.append(inodes);
    return nodes;

#Define routings and cells

num_machine_types = 11;
routings = [[1,[10,1,9],[0,0,0],[1,5,2]],
            [2,[5,8],[0,0],[6,4]],
            [3,[1,2,11],[0,0,0],[1,3,4]],
            [4,[3,10,6],[0,0,0],[1,1,6]],
```

```
            [5,[2,5,9],[0,0,0],[3,1,4]],
            [6,[5,10,8],[0,0,0],[4,5,6]],
            [7,[6,5,10],[0,0,0],[3,6,2]],
            [8,[4,9,11],[0,0,0],[4,6,1]],
            [9,[6,10,11],[0,0,0],[2,6,3]],
            [10,[3,11],[0,0],[2,4]],
            [11,[3,1,4],[0,0,0],[6,3,4]],
            [12,[7,9],[0,0],[3,1]],
            [13,[3,1,5],[0,0,0],[6,4,2]],
            [14,[7,8,10],[0,0,0],[1,3,3]],
            [15,[3,9,4],[0,0,0],[3,2,1]],
            [16,[4,10],[0,0],[3,6]],
            [17,[6,5],[0,0],[6,3]],
            [18,[1,6,10],[0,0,0],[2,3,3]],
            [19,[3,6,5],[0,0,0],[4,1,3]],
            [20,[11,9,4],[0,0,0],[3,4,6]],
            [21,[8,7],[0,0],[1,2]],
            [22,[10,2,11],[0,0,0],[5,3,2]],
            [23,[9,6,10],[0,0,0],[1,6,3]],
            [24,[7,2],[0,0],[2,4]],
            [25,[2,7,6],[0,0,0],[5,6,2]]];
print routings;

cells = range(3);

#Print data
print "Cells:", cells;
for i in range(len(routings)):
    print "Routing ",i,":",routings[i];
print "Part names:"
for i in routings:
    print i[0];
print "Sequences:"
for i in routings:
    print i[1]
print "Manufacturing Costs:"
for i in routings:
    print i[2];
```

```
print "Processing times:"
for i in routings:
    print i[3];

#Write processing steps in steps list
steps = [];
for i in routings:
        for j in [i[1]]:
            steps.append(j);
print "Steps:";
print steps;

#Find all combinations of cell routings using the vert_Slices and
#myCombinations functions
paths = [];
for i in range(len(steps)):
    temp_paths = myCombinations(vert_Slices(steps,i));
    paths.append(temp_paths);

#Verify that the program has worked
print "There are ", len(paths), "routings:";
print "The total number of path combinations is", len(paths[0])+
len(paths[1]) + len(paths[2]);
print paths;
print;
print len(paths[0]);
print paths[0];
print;
print paths[0][0][1][1];
print len(paths);

lengths = [];
#The lengths list contains the number of material handling transfers
in each path

for i in range(len(paths)):
    for j in range(len(paths[i])):
        p_Length = 0;
```

```
        #print len(paths[i][j]);
        for k in range(len(paths[i][j])-1):
            if paths[i][j][k][1] != paths[i][j][k+1][1]:
                p_Length = p_Length + 1;
        lengths.append(p_Length);
print "lengths is", lengths;



#Write output for OPLStudio
output_String = "";
counter = 1;

#print "Mfg cost for routing 0 is: ", routings[0][2];

print "Range of length of paths is ", range(len(paths));

for i in range(len(paths)):
    for j in range(len(paths[i])):
        #Define mfg_Cost, processing time lists and set them to null
        mfg_Cost = [];
        p_Time = [];
        cell = [];
        for k in range(num_machine_types):
            mfg_Cost.append(0);
            p_Time.append(0);
            cell.append(0);
        #print mfg_Cost;
        for l in range(len(routings[i][1])):
            #print "l is ", l;
            #print "Routings are: ", routings[i][1][l];
            #print "Mfg costs are: ", routings[i][2][l]
            mfg_Cost[routings[i][1][l]-1] = routings[i][2][l];
            p_Time[routings[i][1][l]-1] = routings[i][3][l];
            cell[routings[i][1][l]-1] = paths[i][j][l][1]+1;
            #print mfg_Cost;
            #print "<", counter, ",", lengths[counter-1], ">";
        output_String = output_String + "<" + str(counter) + ","   +
str(routings[i][0])   +   ","   +   str(lengths[counter-1])   +   ","
```

```
+str(mfg_Cost) + "," + str(p_Time) + "," + str(cell) + ">,";
        counter = counter + 1;


print "Output string is: "
print output_String

#print mfg_Cost;

print "Hello:";
print "Paths 0 is:";
print paths[0];
print "Paths 01 is:";
print paths[0][1];

print "Paths 0101is:";
print paths[0][1][0][1];
```

APPENDIX 3 OPL Model (MPCFP)

Using IBM ILOG CPLEX Optimization Studio

(Integer linear programming model for solving Wicks and Reasor illustrative example in Chapter 4). This model is used to run the (MPCFP) model for solving the CFP

```
int   NbParts = ...;
int   NbMachine_types = ...;
int   NbCells = ...;
int   NbPeriods = ...;

range Part = 1..NbParts;
range Machine = 1..NbMachine_types;
range Cell = 1..NbCells;
range Period = 1..NbPeriods;

float D[Part][Period] = ...;
float C[Machine] = ...;
float small_C = 0.0001; /* Allows capacity constraint to be slightly
violated due to floating point errors */
float c[Machine] = ...;
float c_prime[Machine] = ...;
float R[Machine] = ...;
float H[Part] = ...;
int UM = ...;
int LM = ...;

/* Data to be generated outside OPL Studio using a script in Python or
any other language */
tuple path {
         int seq;
         int Part;
       int h;
       float m[Machine];
       float q[Machine];
       int cell[Machine];
}

{path} Path = ...;

dvar float+ x[Path][Period];

dvar int+ n[Machine][Cell][Period];
dvar int+ u[Machine][Cell][Period];
dvar int+ v[Machine][Cell][Period];
dvar int+ a[Machine][Cell][Period];
dvar int+ b[Machine][Cell][Period];

float MachineCostFactor = ...;
float IntercellHandlingFactor = ...;
float DemandFactor = ...;

/*execute {
      var i;
      for (i in Path) {
```

```
                  writeln(i.seq, i.Part, i.h, i.m[4]);
            }
      }*/

/* Objective function */

/* Is it the first minimize or the second? or they both the same? */

minimize
      sum (j in Machine, k in Cell, t in Period)
            (c[j] * a[j][k][t] * MachineCostFactor
            + c_prime[j] * b[j][k][t]
            + R[j]*u[j][k][t])
      + sum (p in Path, t in Period)
            (p.h  *  H[p.Part]  *  IntercellHandlingFactor  +  sum  (j  in
Machine) p.m[j])* x[p][t];

subject to {

/* Constraint 2 */

forall (j in Machine)
      forall (k in Cell)
            forall (t in Period)
                  if (t == 1)
                        {
                              n[j][k][t] == a[j][k][t];
                        }
                  else
                        {
                              n[j][k][t]  ==  n[j][k][t-1]  +  a[j][k][t]  -
b[j][k][t] + u[j][k][t] - v[j][k][t];
                        }

/* Constraint 3 */

forall (j in Machine)
      forall (t in Period: t >= 2) {
            sum (k in Cell) u[j][k][t] == sum (k in Cell) v[j][k][t];
      }

/* Constraint 4 */

forall (k in Cell)
      forall (t in Period) {
            LM <= sum (j in Machine) n[j][k][t] <= UM ;
      }

/* Constraint 5 */

forall (j in Machine)
      forall (k in Cell)
            forall (t in Period) {
                  n[j][k][t]    *    (C[j]+small_C)    >=    sum    (p    in
Path:p.cell[j]==k) p.q[j] * x[p][t];
       }
```

```
/* Constraint 6 */

forall (i in Part)
      forall (t in Period) {
            sum (p in Path: p.Part == i) x[p][t] == D[i][t]*DemandFactor;
      }

/*Test constraint
      n[11][1][1]==0;
      n[11][1][2]==0;
      n[11][1][3]==0;*/
}

 /*execute {for (var p in Path)
                  for (var t in Period)
                        if (x[p][t] > 0){
                              write("x[",p.seq,"][",t,"]=",x[p][t],    "
Part = ", p.Part);
                              writeln(" ");
                              }

      }*/

execute {

  writeln("Hello World [a].");

      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        if (a[j][k][t] > 0){

      write("a[",j,"][",k,"][",t,"]=",a[j][k][t]);
                              writeln(" ");
                              }
}

execute {

  writeln("Hello World [b].");

      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        if (b[j][k][t] > 0){

      write("b[",j,"][",k,"][",t,"]=",b[j][k][t]);
                              writeln(" ");
                              }
}

execute {

  //writeln("Hello World [u].");

      for (var j in Machine)
            for (var k in Cell)
```

```
                  for (var t in Period)
                        if (u[j][k][t] > 0){

      write("u[",j,"][",k,"][",t,"]=",u[j][k][t]);
                              writeln(" ");
                              }
}

execute {

  writeln("Hello World [v].");

for (var j in Machine)
      for (var k in Cell)
            for (var t in Period)
                  if (v[j][k][t] > 0){
                        write("v[",j,"][",k,"][",t,"]=",v[j][k][t]);
                        writeln(" ");
                        }
}

execute {

  writeln("Hello World [n].");

for (var t in Period)
      for (var j in Machine)
            for (var k in Cell)
                  if (n[j][k][t] > 0){
                        write("n[",j,"][",k,"][",t,"]=",n[j][k][t]);
                        writeln(" ");
                        }
}

execute {
      for (var p in Path)
            for (var t in Period)
                  if (x[p][t] > 0){
                        write("x[",p.Part,"][",t,"]=",x[p][t]);
                        writeln(" ");
                        write("Cell = ");
                        for (var j in Machine)
                              write (p.cell[j]," ");
                        writeln(" ");
                  }
      }
execute {

            for (var t in Period)
                  for (var k in Cell)
                        for (var j in Machine){
                              var sum= 0;
                              for (var p in Path)
                                    if(p.cell[j] == k)
                                          sum = sum + p.q[j] * x[p][t];
                              writeln("Sum[",t,"][",k,"][",j,"]   =   ",
sum);
```

```
            }
      }

execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + c[j] * a[j][k][t] * MachineCostFactor;
                        }
      writeln("Machine acquisition cost = ", sum);

      }

execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + c_prime[j] * b[j][k][t];
                        }
      writeln("Machine disposal cost = ", sum);

      }

execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + R[j] * u[j][k][t];
                        }
      writeln("Machine relocation cost = ", sum);

      }

execute {
      var sum = 0;
      for (var p in Path)
            for(var t in Period)
                  sum   =   sum   +   p.h   *   H[p.Part]   *   x[p][t]   *
IntercellHandlingFactor;
      writeln("Material handling cost = ", sum)

      }

execute {
      var sum = 0;
      for (var p in Path)
            for(var t in Period)
                  for (var j in Machine)
                        sum = sum + p.m[j]* x[p][t];
      writeln("Production cost = ", sum
      }
```

APPENDIX 4 OPL Robust Model (RMPCFP)

Using IBM ILOG CPLEX Optimization Studio

(Integer linear programming model for solving Cao and Chen illustrative example in Chapter 6). This model is used to run the (RMPCFP) model

```
/********************************************
 * OPL 6.1 Model
********************************************/
int   NbParts = ...;
int   NbMachine_types = ...;
int   NbCells = ...;
int   NbPeriods = ...;
int   NbOperations = ...;
int     NbScenarios = ...;

range Part = 1..NbParts;
range Machine = 1..NbMachine_types;
range Cell = 1..NbCells;
range Period = 1..NbPeriods;
range Operation = 1..NbOperations;
range Scenario = 1..NbScenarios;

float BigM = 10000;
float D[Part][Period][Scenario] = ...;
float prob[Scenario] = ...;
float C[Machine] = ...;
float c[Machine] = ...;
float c_prime[Machine] = ...;
float R[Machine] = ...;
float H1[Part] = ...; /* Intra-cell cost/unit */
float H2[Part] = ...; /* Inter-cell cost/unit */
float omega1 = ...; /*Penalty for surplus in machine capacity */
float omega2 = ...; /*Penalty for shortage in product demand */

int UM = ...;
int LM = ...;

/* Data to be generated outside OPL Studio using a script in Python or
any other language */
tuple path {
         int seq;
         int Part;
         int n_op;
       int machine[Operation];
       float m[Operation]; /*Machining cost*/
       float q[Operation];  /*Which operation */
       int cell[Operation]; /*Which cell */
       int h1;
       int h2;
}

{path} Path = ...;

dvar float+ x[Path][Period][Scenario];
```

```
dvar int+ n[Machine][Cell][Period];
dvar int+ u[Machine][Cell][Period];
dvar int+ v[Machine][Cell][Period];
dvar int+ a[Machine][Cell][Period];
dvar int+ b[Machine][Cell][Period];
dvar float+ tsum[Machine][Cell][Period][Scenario];
dvar float+ fplus[Machine][Cell][Period][Scenario];
dvar float+ fminus[Machine][Cell][Period][Scenario];
dvar float+ eplus[Part][Period][Scenario];
dvar float+ eminus[Part][Period][Scenario];

/* Objective function */

minimize
      sum (j in Machine, k in Cell, t in Period)
            (c[j] * a[j][k][t]
            + c_prime[j] * b[j][k][t]
            + R[j]*u[j][k][t])
      + sum (p in Path, t in Period, s in Scenario)
            (p.h1 * H1[p.Part] + p.h2 * H2[p.Part] + sum (l in Operation)
p.m[l])* prob[s]*x[p][t][s]
            + omega1*sum (j in Machine, k in Cell, t in Period, s in
Scenario) fplus[j][k][t][s]
            + omega2*sum (i in Part, t in Period, s in Scenario)
prob[s]*eminus[i][t][s];
subject to {

/* Constraint 2 */

forall (j in Machine)
      forall (k in Cell)
            forall (t in Period)
                  if (t == 1)
                        n[j][k][t] == a[j][k][t];
                  else
                        n[j][k][t] == n[j][k][t-1] + a[j][k][t] -
b[j][k][t] + u[j][k][t] - v[j][k][t];

/* Constraint 3 */

forall (j in Machine)
      forall (t in Period: t >= 2) {
            sum (k in Cell) u[j][k][t] == sum (k in Cell) v[j][k][t];
      }

/* Constraint 4 */

forall (k in Cell)
      forall (t in Period) {
            LM <= sum (j in Machine) n[j][k][t] <= UM ;
      }

/* Constraint 5 */
// Change to reflect operations

forall (j in Machine)
      forall (k in Cell)
```

```
            forall (t in Period)
                  forall (s in Scenario) {
                        tsum[j][k][t][s] == sum (i in Part, p in Path, l
in Operation: p.cell[l]==k && p.machine[l]==j && l <= p.n_op) p.q[l] *
x[p][t][s];
                        n[j][k][t]   *   C[j]   ==   tsum[j][k][t][s]   -
fplus[j][k][t][s] + fminus[j][k][t][s];
       }

/* Constraint 6 */

forall (i in Part)
      forall (t in Period)
            forall (s in Scenario) {
                        sum  (p  in  Path:  p.Part  ==  i)  x[p][t][s]  -
eplus[i][t][s] + eminus[i][t][s] == D[i][t][s];
      }

}

//execute {writeln("Epsilon is = ",epsilon);}

execute {

      for (var p in Path)
            for (var t in Period)
                  for(var s in Scenario)

                        if (x[p][t][s] > 0){

      write("x[",p.seq,"][",t,"][",s,"]=",x[p][t][s],   "   Part   =   ",
p.Part);
                              writeln(" ");
                              }
}

execute {

  writeln("Hello World [a].");

      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        if (a[j][k][t] > 0){

      write("a[",j,"][",k,"][",t,"]=",a[j][k][t]);
                              writeln(" ");
                              }
}

execute {

  writeln("Hello World [b].");

      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
```

```
                        if (b[j][k][t] > 0){

      write("b[",j,"][",k,"][",t,"]=",b[j][k][t]);
                              writeln(" ");
                              }
}

execute {

      writeln("Hello World [u].");

      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        if (u[j][k][t] > 0){

      write("u[",j,"][",k,"][",t,"]=",u[j][k][t]);
                              writeln(" ");
                              }
}

execute {

  writeln("Hello World [v].");

for (var j in Machine)
      for (var k in Cell)
            for (var t in Period)
                  if (v[j][k][t] > 0){
                        write("v[",j,"][",k,"][",t,"]=",v[j][k][t]);
                        writeln(" ");
                        }
}

execute {

  writeln("Hello World [n].");

for (var t in Period)
      for (var j in Machine)
            for (var k in Cell)
                  if (n[j][k][t] > 0){
                        write("n[",j,"][",k,"][",t,"]=",n[j][k][t]);
                        writeln(" ");
                        }
}

execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + c[j] * a[j][k][t];
                        }
      writeln("Machine acquisition cost = ", sum);

      }
```

```
execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + c_prime[j] * b[j][k][t];
                        }
      writeln("Machine disposal cost = ", sum);

      }

execute {
      var sum = 0;
      for (var j in Machine)
            for(var k in Cell)
                  for (var t in Period) {
                        sum = sum + R[j] * u[j][k][t];
                        }
      writeln("Machine relocation cost = ", sum);

      }

execute {
      var sum = 0;
            for (var p in Path)
                  for(var t in Period)
                        for(var s in Scenario)
                              sum = sum + p.h1 * H1[p.Part] * x[p][t][s]
+ p.h2 * H2[p.Part] * x[p][t][s];
      writeln("Material handling cost = ", sum)

      }

execute {
      var sum = 0;
            for (var p in Path)
                  for(var t in Period)
                        for(var s in Scenario)
                              for (var l in Operation)
                                    sum = sum + p.m[l]* x[p][t][s];
      writeln("Production cost = ", sum)

      }

execute {
            for (var p in Path)
                  for (var t in Period)
                        for(var s in Scenario)

                  if (x[p][t][s] > 0){

      write("x[",p.Part,"][",t,"][",s,"]=",x[p][t][s]);
                        writeln(" ");
                        write("Cell = ");
                        for (var l in Operation)
                              write (p.cell[l]," ");
```

```
                        writeln(" ");
                  }
      }

/*execute {
            for (var j in Machine)
                  for (var k in Cell)
                        for (var t in Period){
                              var sum= 0;
                              var total_avail_time;
                              //writeln("Capacity = ", C[j]);
                              //writeln("n = ", n[j][k][t]);
                              total_avail_time = C[j] *n[j][k][t];
                              for (var p in Path)
                                    for (var l in Operation)
                                          if   (p.cell[l]   ==   k   &&
p.machine[l]==j)
                                                sum  =  sum  +  p.q[l]  *
x[p][t];
                              writeln("Processing
Time[",j,"][",k,"][",t,"] = ", sum, " Capacity[",j,"][",k,"][",t,"] = ",
total_avail_time);
                        }*/
execute {
      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        writeln("Processing  Time[",j,"][",k,"][",t,"]  =
", tsum[j][k][t], " Capacity[",j,"][",k,"][",t,"] = ", n[j][k][t] * C[j]);

}

execute {
      var sum = 0;
      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        for (var s in Scenario)
                              sum = sum + fminus[j][k][t][s];

      writeln("Fminus Total is = ", sum);

   }

 execute {
      var sum1 = 0;
      for (var j in Machine)
            for (var k in Cell)
                  for (var t in Period)
                        for (var s in Scenario)
                              sum1 = sum1 + fplus[j][k][t][s];

      writeln("Fplus Total is = ", sum1);

   var sum2 = 0;
   for (var i in Part)
      for (var t in Period)
```

```
            for (var s in Scenario)
                  sum2 = sum2 + prob[s]*eminus[i][t][s];

      writeln("eminus Total is = ", sum2);
      sum3 = omega1*sum1 + omega2*sum2;
      writeln("Penalty cost is = ",sum3);
}

 execute {
   var sum = 0;
   for (var i in Part)
      for (var t in Period)
            for (var s in Scenario)
                  sum = sum + prob[s]*eplus[i][t][s];

      writeln("eplus Total is = ", sum);
}
```

APPENDIX 5 Python code

(Generating all possible routing sequences for Cao and Chen illustrative example in Chapter 6)

```
/*******************************************
    * Python (V.2.7)
*******************************************/
###Combinations code
##a = [[31,32],[11,12],[41,42]]
##
##r=[[]]
##for x in a:
##    t = []
##    for y in x:
##        for i in r:
##            t.append(i+[y])
##    r = t
##
##print r

#Combinations function
#This function picks all possible combinations from a list
def myCombinations(list):
    r=[[]]
    for x in list:
        t = []
        for y in x:
            for i in r:
                t.append(i+[y])
        r = t
    return r;

## Function to returns cell combinations based on number of operations and cells
```

```
def getCellCombinations(n_operations,n_cells):
    temp_string1 = [];
    for i in range(n_operations):
        temp_string2 = [];
        for j in range (n_cells):
            temp_string2.append(j+1);
        temp_string1.append(temp_string2);
    return myCombinations(temp_string1);

## Function to find the number of intra-cell material handling transfers based on cell sequence string
def getMHTransfers1(cell_list):
    num_transfers = 0;
    for i in range(len(cell_list)-1):
        if (cell_list[i] == cell_list[i+1]):
            num_transfers = num_transfers + 1;
    return num_transfers;

## Function to find the number of inter-cell material handling transfers based on cell sequence string
def getMHTransfers2(cell_list):
    num_transfers = 0;
    for i in range(len(cell_list)-1):
        if (cell_list[i] != cell_list[i+1]):
            num_transfers = num_transfers + 1;
    return num_transfers;

routings =
[[1,3,[1,2,3],[0,0,0],[0.020,0.024,0.010]],[2,2,[2,4],[0,0],[0.028,0.020]],[3,3,[1,2,3],[0,0,0],[0.014,0.016,0.01]],[4,2,[1,4],[0,0,0],[0.026,0.016]],[5,2,[2,3],[0,0],[0.024,0.01]]];
##print routings;
```

```
num_cells = 2;
##num_operations = routings[0][1];
##print 'Num operations = ', num_operations;
##print 'Num cells = ', num_cells;

##cell_string = getCellCombinations(num_operations,num_cells);
##print 'Cell String is: ', cell_string;
##print 'Routings[0] is', routings[0];

path = [];
for i in routings:
    cell_string = [];
    #print 'Num_operations for routing is', i[1];
    cell_string = getCellCombinations(i[1],num_cells);
    for j in cell_string:
        path.append(i+[j]);

for i in range(len(path)):
    path[i].append(getMHTransfers1(path[i][5]));

for i in range(len(path)):
    path[i].append(getMHTransfers2(path[i][5]));

print 'path[0] is:', path[0];

print('Path = {');
for i in range(len(path)):
    print '<', i+1, ",";
    print(str(path[i])[1 : -1]);
    print '>,';
print('};');
```