Comprehensive Models and Solution Procedures for Integrated Cellular Manufacturing Systems Design

Steve Ah kioon

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

August 2007

Your file Votre référence
ISBN: 978-0-494-31145-5
Our file Notre référence
ISBN: 978-0-494-31145-5

ABSTRACT

Comprehensive Models and Solution Procedures for Integrated Cellular Manufacturing Systems Design

Steve Ah kioon, Ph.D.

Concordia University, 2007

Cellular manufacturing (CM) has emerged because of the need for manufacturing organizations to supply products that require more customization and that have shorter product life cycles and times to market. This shift from mass production to demands for mid-volume and mid-variety product mixes has shown that traditional manufacturing systems relying on functional or line layouts are not efficient and flexible enough. CM is an alternate manufacturing system combining the high throughput rates of line layouts with the flexibility offered by functional layouts (job shops). The benefits include reduced set-up times, material handling, in-process inventory, better product quality and market response time. The benefits of CM can only be achieved by sufficiently incorporating the real-life structural and operational features of a manufacturing plant when creating the cellular layout. This research presents two integrated CM models, with an extensive coverage of important manufacturing structural and operational features.

The first CM model is initially formulated as a mixed integer non-linear program that incorporates multi-period production planning and dynamic system reconfiguration with deterministic production requirements. It consists of alternate process routings, the operation sequence of parts, machine duplication, machine procurement and machine capacity. Preliminary computational experiments show that only small-scale problems can be solved using the developed mixed integer non-linear program. This warrants the linearization procedures that are proposed to convert it into a linearized mixed integer programming formulation so as to solve problems of larger scale. This linearized mixed integer program is first solved using an exact solution (ES) procedure through the simplex-based branch and cut procedure of CPLEX software. Computational results are presented by solving some small-scale to large-scale numerical examples, extracted from existing literature. Although small to medium-sized problems can be solved using ES, larger ones (representing real-life sized problems) cannot be solved within reasonable computational times. A tabu search meta-heuristic is, therefore, developed to solve this mixed integer linear program. The results show that the tabu search procedure generates good quality solutions for real-life size problems within acceptable computational times.

The second model addresses the same attributes as the first one but an important extension is the introduction of routing flexibility in the system by the formation of additional alternate part process routings, called *contingency routings*. In addition to the alternate *main process routings* being created for each part type, the model forms *contingency routings*, which serve as backups so as to effectively address the reality of *part process routing* disruptions, thus allowing the cellular manufacturing system to operate in a continuous manner even in the event of any breakdowns. This enhanced model is formulated as a mixed integer linear program and is then solved using the simplex-based branch and cut procedure of CPLEX (ES). The results show that the *contingency* and *main routings* can be formed simultaneously, thus bringing enhanced system and routing flexibility. The trade-off between the increased system flexibility obtained versus the additional cost to be incurred through the formation of *contingency* routings for all parts is discussed by comparing the solutions obtained from the model without routing flexibility and the one with routing flexibility. It is found that the enhanced system flexibility enabled by *contingency routings* offsets the additional system investment costs.

Keywords: integrated cellular manufacturing systems design, mixed integer programming, linearization procedures, simplex-based branch and cut procedure, tabu search, routing flexibility, *contingency* routings, enhanced CM system flexibility.

ACKNOWLEDGEMENTS

I am grateful to have been surrounded, encouraged and helped by so many great people during this journey of intellectual stimulation and hard work.

I thank *QuanYin* for teaching me the lessons of patience and faith. I thank *God* for giving me the courage and will-power to overcome so many difficulties and challenges.

I thank my mum, my dad, Yony, Pin and Foun for being ever present with their love and constant encouragement. Thank you mum for helping me foster my spiritual development during this journey. Special thanks to Yony, for your help in computer programming.

I thank you, Karen Hung, for being the very special person that you are. Thank you so much for your enduring love, patience, encouragement and understanding. Your hard-work ethic and discipline are inspiring.

I thank my thesis supervisor, Dr. Akif Bulgak, for believing in me and giving me such a great opportunity within high-level academic research. Thank you for being a great guide and motivator, and for your ever-caring attitude. Through thick and thin, you have always given me your full support, for which I will always be grateful.

I thank you, Dr. Tolga Bektas, for the endless hours that you generously spent helping me in my research and teaching me about optimization software and tools. Through our little research team, we have succeeded in publishing papers in the academic community thanks to your incisive comments, insightful suggestions and swift professional reviews.

I thank my friends and colleagues at Concordia University for their help and support. Amongst these many people, I especially wish to thank Chaher Alzaman as we continually encouraged and helped each other throughout our studies. I thoroughly enjoyed our many conversations and am looking forward to further more. I also thank Dr. Fantahun Defersha and Dr. Asif Raza who have generously helped me during certain stages of my research.

My thanks also go to the CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation) who have provided me with the powerful computer facilities that I needed for my research.

Finally, I thank the members and chair of my examinations committee, namely Dr. Akif A. Bulgak, Dr. Mingyuan Chen, Dr. Ali Akgunduz, Dr. Maria Elektorowicz, Dr. Tamer Boyaci and Dr. Brigitte Jaumard

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Sets:

Index set of part types

$\mathcal{P} \quad = \{1, 2, 3, \ldots, P\}$

Index set of operations indices for part type p

$\mathcal{K}(p) \quad = \{1, 2, 3, \ldots, K_p\}$

Index set of machine types

$\mathcal{M} \quad = \{1, 2, 3, \ldots, M\}$

Index set of machine types that can perform operation k for part type p

$\mathcal{Q}(k,p) \quad = \{1,2,3,\ldots,Q_{kp}\}$

Index set of cells

$\mathcal{C} \quad = \{1,2,3,\ldots,C\}$

Index set of time periods

$\mathcal{T} \quad = \{1,2,3,\ldots,T\}$

Model Parameters:

δ_m	Relocation cost per machine type m per period;	$\forall m \in \mathcal{M}$
α_m	Maintenance and overhead costs per machine type m	$\forall m \in \mathcal{M}$
ε_m	Operating cost per unit time per machine type m	$\forall m \in \mathcal{M}$
μ_m	Procurement cost per machine type m	$\forall m \in \mathcal{M}$
O_p	Outsourcing cost per part type p	$\forall p \in \mathcal{P}$
H_p	Inventory holding cost per part type p per time period	$\forall p \in \mathcal{P}$
β_p	Production cost per part type p	$\forall p \in \mathcal{P}$
IE_p	Intercellular material handling cost per part type p	$\forall p \in \mathcal{P}$
IA_p	Intracellular material handling cost per part type p	$\forall p \in \mathcal{P}$
$D_p(t)$	Demand for part type p at time period t	$\forall p \in \mathcal{P}$, $\forall t \in \mathcal{T}$
$A_m(t)$	Number of units of machine type m available at time period t	$\forall m \in \mathcal{M}$, $\forall t \in \mathcal{T}$
T_m	Capacity of one unit of machine type m during one period	$\forall m \in \mathcal{M}$
B_U	Upper Cell Size Limit	
B_L	Lower Cell Size Limit	
e_{kpm}	Processing time of operation k on machine m per part p	$\forall k \in \mathcal{K}(p)$, $\forall p \in \mathcal{P}$, $\forall m \in \mathcal{Q}(k,p)$

Model Decision Variables

$O_p(t)$ Number of parts p to be outsourced at time t

$\forall p \in \mathcal{P}, \forall t \in \mathcal{T}$

$V_p(t)$ Quantity of inventory of part type p kept in period t and carried over to period $t + 1$

$\forall p \in \mathcal{P}, \forall t \in \mathcal{T}$

$N_{mc}(t)$ Number of machines of type m present at cell c at time t.

$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$

$N^{+}_{mc}(t)$ Number of machines of type m added to cell c at time t

$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$

$N^{-}_{mc}(t)$ Number of machines of type m removed from cell c at time t

$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$

$A^{*}_{m}(t)$ Number of units of machine type m available at time period t, after considering the extra machines brought into the system by machine procurement

$\forall m \in \mathcal{M}, \forall t \in \mathcal{T}$

$BN_m(t)$ Number of machines of type m procured at time t.

$\forall m \in \mathcal{M}, \forall t \in \mathcal{T}$

Decision variables for main routings

$NM_{mc}(t)$ Number of machines of type m present at cell c at time t and used for *main* routings.

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$X_{kpmc}(t)$ Number of parts of type p processed by operation k on machine type m in cell c at time t

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$Z_{kpmc}(t)$ 1 if operation k for part type p is carried out on machine type m in cell c at time t, and 0 otherwise

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

Decision variables for contingency routings

$NP_{mc}(t)$ Number of machines of type m present at cell c at time t and used for *contingency* routings.

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$XP_{kpmc}(t)$ The quantity of parts of type p that is processed by operation k on machine type m in cell c at time t. This is defined for backup routings.

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$ZP_{kpmc}(t)$ 1 if operation k for part type p is carried out on machine type m in cell c at time t, and 0 otherwise. This is defined for backup routings.

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

Linearization decision variables

$YP_{kpmc}(t)$ and $YM_{kpmc}(t)$ are non-negative continuous variables that are used to linearize the intercellular movement cost term. The expression $YP_{kpmc}(t)+YM_{kpmc}(t)$ gives the magnitude of the quantity of parts involved in intercellular movement.

$Z_{kpmnc}(t)$ and $W_{kpmnc}(t)$ are non-negative continuous variables used to linearize the intracellular movement cost term. The linearization variable $Z_{kpmnc}(t)$ defines whether there is intercellular movement. There is intracellular movement ($Z_{kpmnc}(t)>0$) when operation k of part p at time t is done on machine type n in cell c and this is followed by the next operation, $k+1$, being done on any machine type m within the same cell c. There is no intracellular movement ($Z_{kpmnc}(t)=0$) when the consecutive operations are done in different cells. $W_{kpmnc}(t)$ is the magnitude of the number of parts involved in the intracellular movement.

LIST OF ACRONYMS

AAA	-	Assignment Allocation Algorithm
AI	-	Artificial Intelligence
ALC	-	Average Linkage Clustering
ANN	-	Artificial Neural Network
BEA	-	Bond Energy Analysis
CF	-	Cell Formation
CFA	-	Component Flow Analysis
CFP	-	Cell Formation Problem
CI	-	Cluster Identification
CLC	-	Complete Linkage Clustering
CLS	-	Candidate List Size
CM	-	Cellular Manufacturing
CMS	-	Cellular Manufacturing Systems
DCA	-	Direct Clustering Algorithm
EA	-	Evolutionary Algorithm
FIFO	-	First-In-First-Out
GA	-	Genetic Algorithm
GAP	-	Generalized Assignment Problem
GT	-	Group Technology
ISNC	-	Ideal Seed Non-hierarchical Clustering
JIT	-	Just-In-Time

MFA-SA	-	Hybrid Mean Field Annealing – Simulated Annealing
MGI	-	Machine Group Identification
MODROC	-	Modified Rank Order Clustering
MST	-	Minimum Spanning Tree
NP	-	Non-Polynomial
PFA	-	Production Flow Analysis
PFI	-	Part Family Identification
PF/MC	-	Part Family/Machine Cell
ROC	-	Rank Order Clustering
SA	-	Simulated Annealing
SLC	-	Single Linkage Clustering
TLS	-	Tabu List Size
TS	-	Tabu Search
WIP	-	Work In Progress
ZODIAC	-	Zero-One Data Ideal seed Algorithm for Clustering

Chapter 1

Introduction

The increasingly competitive global market-place in the last few decades has led to manufacturing industries having to adapt their manufacturing strategies with more focus set upon enabling product customization on a large scale. Such firms need to operate in a flexible, efficient and diverse production environment to quickly manufacture products, meeting or even exceeding the customers' expectations of variety and customization, low cost and high quality. The increasing focus on creating better socio-technical systems (management-worker relations, team work and job satisfaction) in the workplace, changing marketing strategies (changes in product design and demand) and the introduction of new technology in the manufacturing environment form part of the challenges that manufacturers have to contend with in order to survive in the international market. Group Technology (GT) is a philosophy that has been adapted to address such manufacturing issues. Its tenet is that by grouping similar problems together, a single solution can be determined and used to solve them, thereby saving time and effort. Similarities are exploited in three ways: similar activities are performed together, similar tasks are standardized and information about recurring problems are efficiently stored and retrieved. The most important implementation of the GT concept in manufacturing is Cellular Manufacturing (CM), where the design and process similarities of parts are exploited.

1.1. Cellular Manufacturing

Cellular Manufacturing (CM) is an application of GT to manufacturing and has emerged because of the need for manufacturing organizations to cope with shorter product life-cycles, time-to-market and diverse customer needs. With increased global competition, there has been a shift from mass production to the production of large product mixes. CM involves the formation of relatively independent work cells (dedicated clusters of machines or manufacturing processes) that each specialize in the manufacture of a narrow line of products, or even part of a product, known as part families. CM has been implemented in the manufacturing industry for a number of products including machinery and machine tools, agricultural and construction equipment, electrical and electronic products and components, fluid handling and flow control devices, heating and cooling products and components, mechanical hand tools, woodcutting tools, seals, gaskets, engines, bearings, defense products, hospital and medical equipment. Traditional manufacturing systems involving functional or line layouts do not offer the required efficiency and flexibility to adapt to these product demands.

A functional or process layout (a job shop) copes well with high product variety but does not provide adequate throughput when high product volumes are required. In such a configuration, similar processes are located together by laying out all machines of the same type in the corresponding department of the manufacturing plant. For instance, the milling department would specialize in milling operations and would contain milling machines only (called process specialization). This type of manufacturing system is capable of producing an extensive variety of products through small lot sizes. A typical layout is illustrated in figure 1-1, showing that batches of parts have to be moved from one department to another in order to complete all of the required operations. Each one of the parts requires different operations and has different operation sequences with varied processing times needed for each operation Production is done by moving a batch of parts to another processing department only when the entire batch has completed all of its processing in the previous department. Such a system involves about only 5% of the time being spent on a machine in productive activity with the remaining 95% being spent moving and waiting (non-productive activity). Therefore, this system is characterized by lower throughput and longer production times, high levels of work-in-progress (WIP) and high production costs. Furthermore, it involves considerable material handling as batches of parts have to be moved between many different departments or even through the entire manufacturing facility.

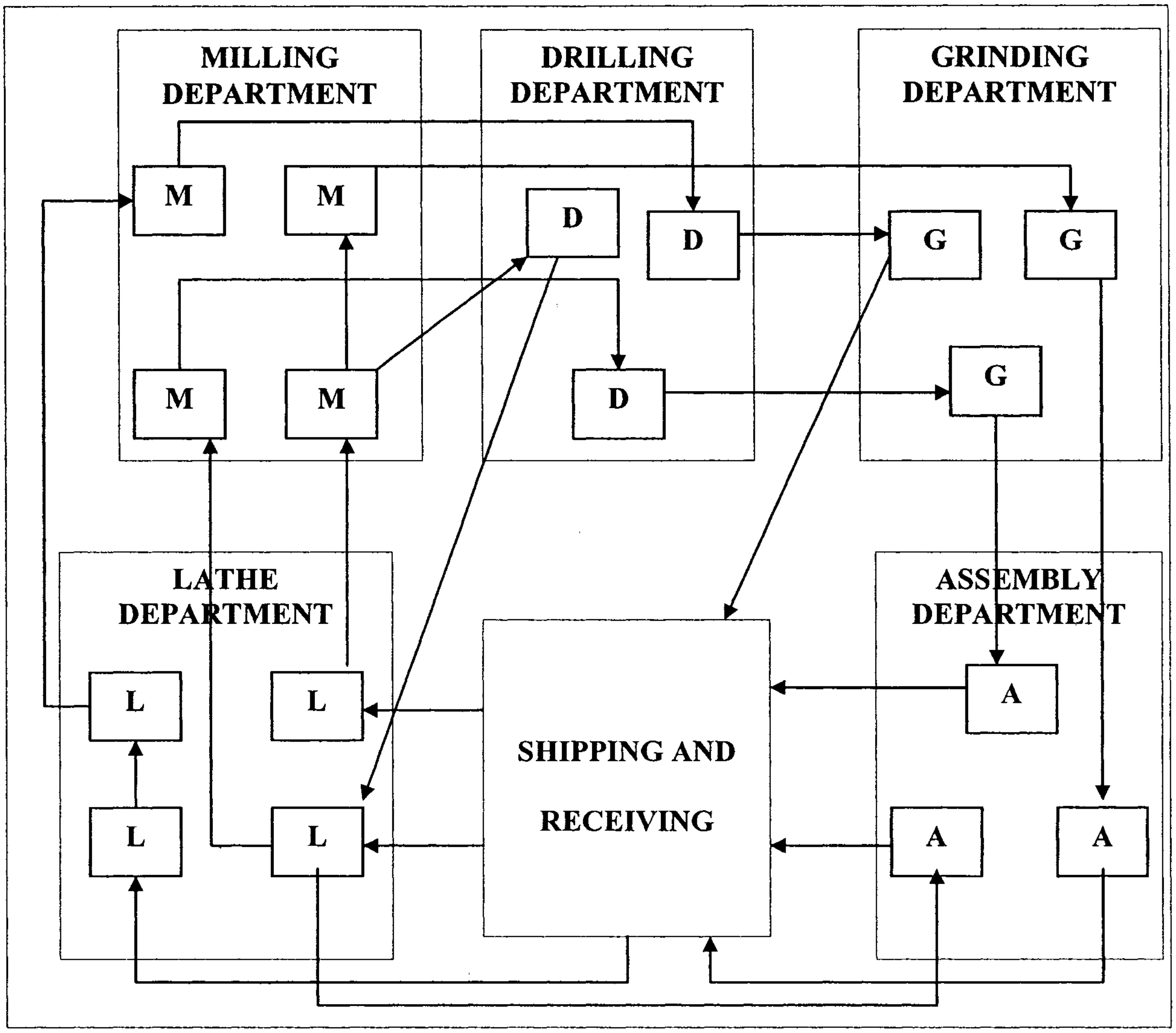

Figure 1-1 : A functional layout (job shop)

A line layout (a flow line) copes well with high production volumes through high production rates and low costs but cannot provide the flexibility required to cope with product customization. This configuration is typically used in simple process industries, in continuous assembly and for the mass production of a large quantity of parts. Each flow line is designed for the manufacture of a specific product, where specialized machines are arranged in such a way as to provide the exact operation sequence required by each product, as shown in figure 1-2. The high machine investment costs are justified by the production of large quantities of these products. The lack of manufacturing flexibility arises from the fact that these machines can only perform a limited range of operations and cannot be reconfigured frequently. Any change in product design has to be accompanied by a corresponding change in the design of the processes (changes in the operations or operation sequence). This invariably leads to a new configuration being required on the existing production lines for the new product design(s). This results in lost production times and lowered production rates, especially due to the increased change-over and setup times experienced during reconfiguration of the lines.

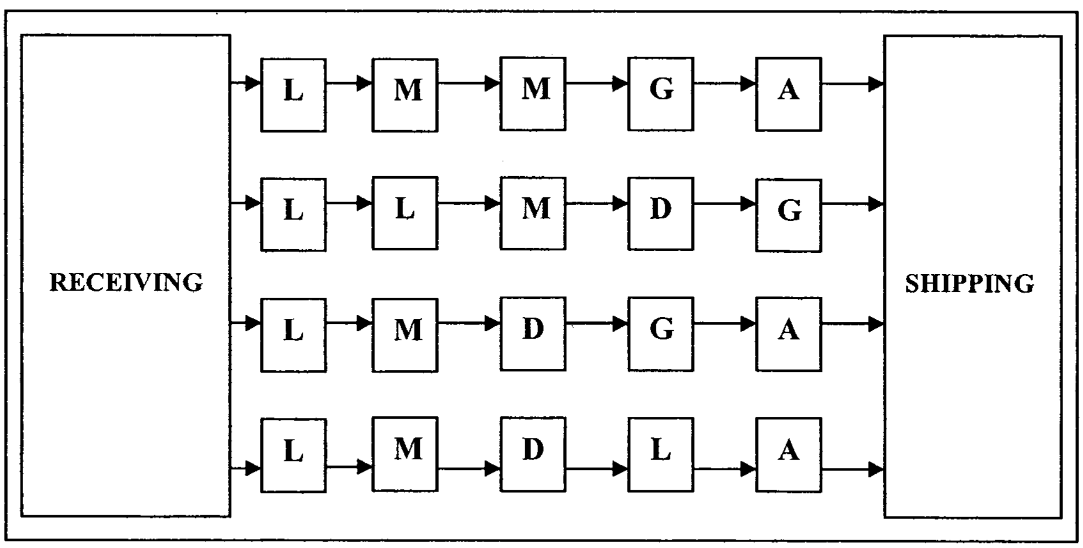

Figure 1-2 : A line layout (flow line)

Therefore, job shops and flow lines cannot combine high production flexibility with high production volumes, being unable to respond effectively to changes in product design and demand. Cellular manufacturing (CM) has emerged to cope with such production requirements since it is a hybrid manufacturing system that incorporates the flexibility of functional layouts and the high production rate of line layouts. CM involves the formation of part families based on part processing similarities and of machine cells to produce these part families. A part family is defined as a collection of parts that can be processed on the same group of machines because of geometric shape and size or similar processing steps required in their manufacture. A manufacturing cell consists of machines of different types (functionally different) that are dedicated to the manufacture of a part family, as depicted in figure 1-3. This type of system provides the high production rates akin to those obtained from flow lines. Cellular reconfiguration in CM systems is made possible by the use of general-purpose machines that can be changed in order to respond to new product designs or demands, thereby providing the production flexibility similar to job shops.

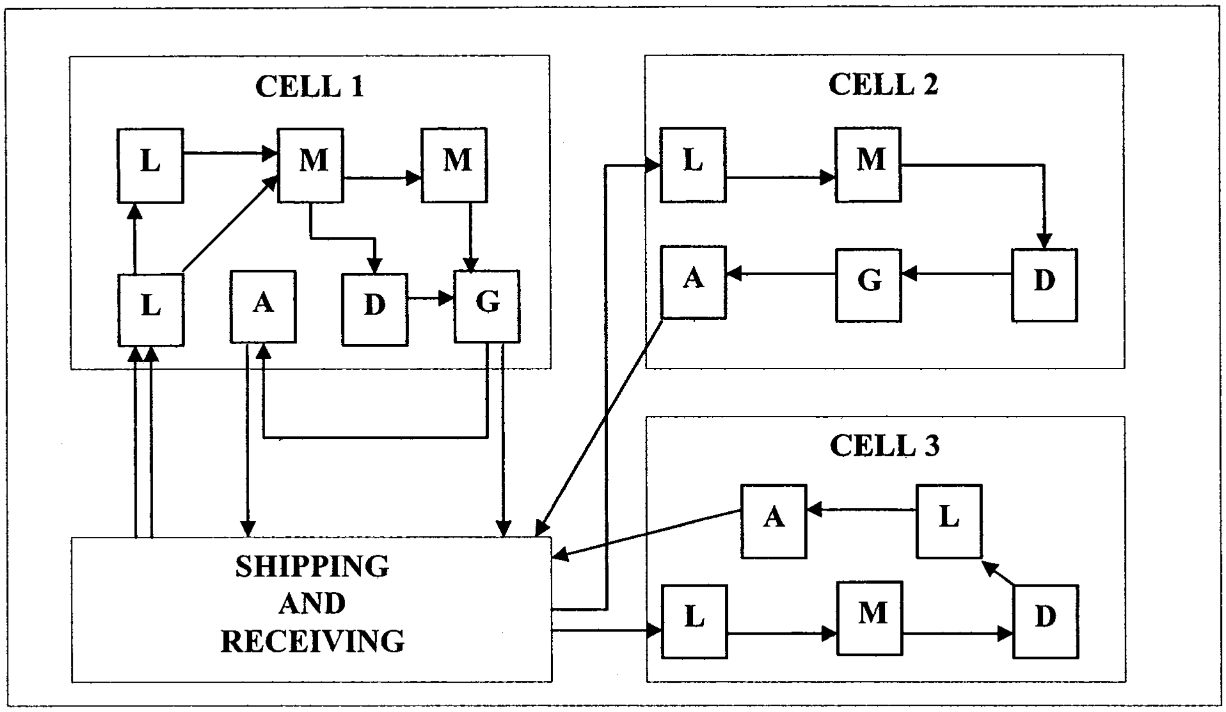

Figure 1-3 : A cellular (group) layout

Therefore, CM is a manufacturing system which can produce medium-volume and medium-variety part types more efficiently and economically than the other types of manufacturing systems. The product design and demand characteristics have to justify the implementation of any CM system since the latter is not a universal panacea to all of the challenges inherent to modern day manufacturing. For instance, products that have very large production volumes are better processed using pure flow lines. Moreover, products fetching small production volumes coupled with widely varying part processing operations do not warrant the need for CM systems.

1.2. Benefits of Cellular Manufacturing

Manufacturing plants with successful CM implementations have reported benefits such as reduced set-up times, reduced material flow, reduced inventory and work-in-process, better socio-technical systems (human work environment) and product quality (Wemmerlov and Hyer, 1989; Singh and Rajamani, 1996; Brandon, 1996; Wemmerlov and Johnson, 1997; Olorunniwo, 1997). The benefits that have been established in these papers are based on surveys of companies implementing CM systems, on company-specific case studies and other related reports. As such, the advantages derived from CM implementation can be summarized as follows:

1. **Reduction in throughput times**. In CM systems, parts are moved between cells in small batches and within each cell individual parts can be moved to the next machine after completing their operation on the previous machine. This contributes in significantly reducing the waiting times. Furthermore, CM leads to an easier identification of bottlenecks since the material flow within each cell can be better tracked. Therefore, this is conducive in enabling delivery due dates to be met and in providing better customer lead times.

2. **Reduction in setup times and lot sizes**. Since a manufacturing cell is designed to manufacture a part family (parts having the same processing requirements: required operations, tolerances, machine tool capacities, similar shapes and sizes), it can accommodate the standardization of equipment, tools, jigs and fixtures. The parts can, therefore, be quickly processed without the need to redesign tools for

that matter. Moreover, the use of adapters and generic fixtures significantly reduces the time required to change tools and fixtures. Owing to the considerable reduction in setup times, it also becomes more economical to operate using smaller lot (batch) sizes.

3. **Reduction in materials handling cost and time**. With most of the processing of a part family taking place inside a single work cell, there is less material transfer occurring over large distances (and times). This contributes in simplifying material flow within the whole cellular layout facility.

4. **Reduction in WIP and finished goods inventory**. Owing to the reductions in setup times, lot sizes, waiting times and material handling, parts can be produced on a just-in-time (JIT) basis. This enables reductions in WIP and finished goods inventory to be achieved.

5. **Reduction in space**. Owing to the standardization of equipment, tools, jigs and fixtures as well as reduced WIP and finished goods inventory, less factory and warehousing space is required.

6. **Better production scheduling and response to product design changes**. Reductions in setup times and lot sizes, simplified material flow and the ability of cells to be reconfigured quickly (through standardized equipment) effectively make production scheduling and product design changes significantly more

manageable. When a new part is introduced, the designer can use the database for existing part families which are similar in processing requirements. The aggregation of machines into cells reduces the number of work centers that have to be scheduled. Owing to the relative independence of the cells, changes in the production scheduling or design of a particular product line can be better addressed within one cell instead of the whole production facility.

7. **Better product quality**. Since parts are manufactured in small lot sizes, any quality defects can be immediately tracked and addressed within a cell, without having to stop production in other cells. Also, quality improvement circles within each cell can be more effective since the latter consists of a team of operators working together on a daily basis and who are aware of their increased job responsibility and ownership.

8. **A better socio-technical environment**. Better employee satisfaction, labor relations, worker motivation and reduced employee absenteeism and turnover have been achieved as a result of enhanced job enrichment and status (Brandon, 1996). In fact, within a cell, this is made possible thanks to higher levels of variety, identity, significance, autonomy and feedback.

CM is also one of the methods of implementing lean manufacturing as it paves the way to small manufacturing lot sizes (batch manufacturing) and improved machine changeover and setup times. A cellular layout is also a typical pre-requisite for achieving just-in time (JIT) production since it helps achieve a decrease in inventory, work-in-progress (WIP) and waste, simplifies the work-flow and enables the scaling down of equipment (for faster setup and changeover).

1.3. Research Objectives

The design of cellular manufacturing (CM) system is also known as the cell formation problem (CFP), part family/machine cell (PF/MC) formation, and manufacturing cell design. The design of a cellular manufacturing system consists of three main phases (Dimopoulos and Zalzala, 1998) as shown in figure 1-4.

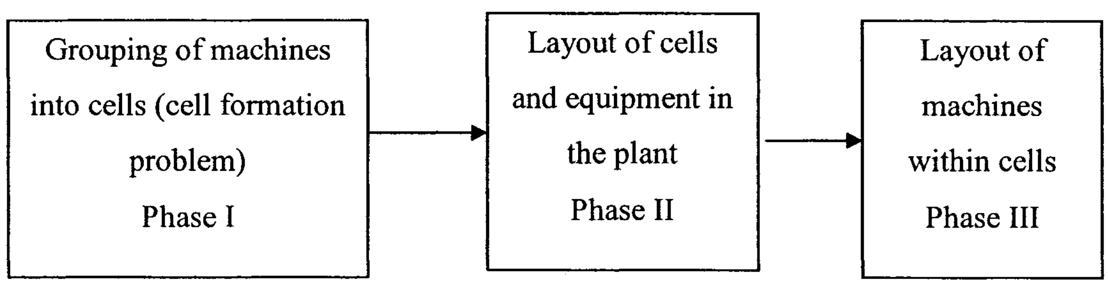

Figure 1-4 : Main stages in the design of a cellular manufacturing system

This research focuses on the first phase of CM design where part families and machine cells are formed, followed by the assignment of the former to the latter. The effectiveness of this phase is related to several decisions in the CM design process. These manufacturing decisions are related to both system structure and system operation, and affect the whole system cost and performance. As such, manufacturing attributes must be incorporated to address these structural (e.g. number of machines, machine availability, capacity) and operational (e.g. production planning aspects) issues, allowing the cell designer to evaluate cell design using cost-oriented or performance-oriented objectives. Therefore, with regards to CM modeling, the following research objectives are met in this thesis:

1. Identify the key manufacturing attributes, whose consideration within a CM system can lead to more effective and realistic CM models.
2. Integrate these manufacturing dimensions with the cell formation problem and develop a mathematical formulation for the CM model(s) under investigation.

It has been shown that the cell formation problem is NP-hard (Garey and Johnson, 1979). Since the models developed in this thesis integrate many manufacturing attributes and other manufacturing problems in addition to the cell formation problem, it is expected that high levels of computational difficulty will be experienced. Furthermore, Ballakur (1985) has shown that the problem of CM design is NP-hard, even under fairly restrictive conditions. Therefore, with regards to CM design solution methodology, the following research objectives are also met:

3. Develop efficient and exact solution procedures for solving the proposed integrated CM models.
4. Evaluate the ability of some selected off-the-shelf optimization softwares in solving various CM problem instances.

5. Develop an efficient meta-heuristic solution procedure using an appropriate programming language for solving the CM models.
6. Evaluate and compare the exact and meta-heuristic solution procedures in terms of solution quality and computational effort.
7. Evaluate the potential benefits of implementing CM systems that have enhanced system flexibility.

1.4. Research Approach

With regards to the development of integrated CM models, a number of steps are taken in order to select the important manufacturing aspects to be incorporated and to implement some possible solution approaches:

1. Through an extensive literature review of existing models and CM design-related work, the important manufacturing attributes are identified and selected.
2. The model is then formulated as a mathematical program that comprises all of the selected manufacturing attributes. This will be referred to as model **A**.
3. The mathematical program is converted into a format that can be recognized and solved by selected off-the-shelf optimization packages.
4. Various problem instances of varying size are selected and populated using existing data sets from literature and case studies.
5. These problems are solved to optimality using the selected optimization packages.
6. Analysis of the results is done on the basis of the computational effort required to solve problems of increasing sizes.

7. Develop a meta-heuristic solution approach to solve model **A**, with special emphasis on problems of larger sizes.
8. Select an appropriate programming language to implement the meta-heuristic approach and to solve the problems.
9. Validate the meta-heuristic solutions by comparing them with the corresponding optimal solutions that could be obtained for the same problem instances via the exact solution procedure and the selected off-the-shelf software package.
10. Compare the meta-heuristic and optimal solutions with respect to solution quality and computational effort.
11. The first step in evaluating the benefits of increased system flexibility is achieved by extending model **A** through the inclusion of *contingency (backup)* routings. This will be referred to as model **B**.
12. To assess the effect of increased system flexibility, the costs of the CM systems (with and without increased flexibility) are compared through the optimal solutions obtained for problems that could be solved.

1.5. Outline of Thesis

The remainder of this thesis is organized as follows. In chapter 2, a literature review is presented with regards to CM systems design methods, including the CM solution procedures and modeling approaches. Chapter 3 presents a comprehensive mathematical CM model (model **A**) that integrates the important manufacturing attributes that have been identified in chapter 2. The properties of model **A** are discussed followed by the implementation of some linearization procedures. The linearized model is then solved using an off-the-shelf optimization package for various numerical examples followed by a detailed discussion of the computational results. Chapter 4 presents the development, implementation and application of the meta-heuristic procedure (tabu search) that solves the linearized model **A** from chapter 3. The same numerical examples from chapter 3 are thus solved using the tabu search solution procedure. This enables comparisons and discussions to be made with respect to the results obtained through the exact and tabu search solution procedures. Chapter 5 presents the second mathematical model (model **B**) that includes all of the aspects present in model **A** but which is characterised by the inclusion of routing flexibility. The properties of model **B** are also discussed. In chapter 5, model **B** is solved using the same numerical examples, with an exact solution procedure being applied again. The effects and investment costs related to forming contingency routings are also investigated. Chapter 6 presents the summary, conclusions and future research directions. This chapter also highlights the research contributions brought to both CM modeling and solution methodologies. Future research directions are also discussed.

Chapter 2

Literature Review

2.1. Introduction

The cell formation problem has been extensively researched and the literature abounds with various solution methodologies that have been proposed. Comprehensive taxonomies of work devoted to the part-machine grouping problem have been carried out by many researchers (Greene and Sadowski, 1984; Wemmerlov and Hyer, 1986; Kusiak, 1987; Singh, 1993; Vakharia and Slim, 1994; Joines *et al.*, 1996; Selim *et al.*, 1998). The rest of this chapter consists of two main sections. Firstly, it presents a review of the relevant literature pertaining to the solution methods proposed to solve the cell formation problem. Secondly, recently published work and reviews (Mansouri *et al.*, 2000; Balakrishnan and Cheng, 2007) are considered with a view to identifying the key manufacturing attributes that need to be taken into account during the CM system design phase in addition to the cell formation problem.

2.2. Taxonomy of CM Design Methods

The cell formation problem deals with the identification of part families and machine groups on which to process these parts. To enable this, a basic relationship must be identified between a part and a set of machines, for example a part *process routing*, where the latter is defined as the machines or work centers visited by a part type according to the sequence and type of operations required. Part families can be formed such that all parts within a family are processed on the same machine group. Similarly, machines can be grouped into cells if they process the same set of parts. After establishing part and machine populations, the cell formation problem can be reduced to three major decisions: firstly, identification of part families (PFI), secondly identification of machine cells (MGI) and thirdly, the allocation of part families to machine cells or vice-versa (PF/MG). These three decisions are interrelated and are the sub-problems to the cell formation problem. This section reviews the different solution methodologies that have been developed for the cell formation problem. The methods used for part family/ machine cell formation are classified as either design-oriented or production-oriented. Design-oriented methods form part families according to the design features of the parts whilst the production-oriented methods form do so based on the processing requirements of the parts.

2.2.1. Descriptive Procedures

Descriptive procedures can be classified into three main classes (Selim *et al.*, 1998): part family identification (PFI), machine group identification (MGI) and part family/machine group identification (PF/MG). PFI-related methods are further classified as those utilizing informal systems (rules of thumb or visual examination) and those utilizing formal classification and coding systems. Through the use of the latter, parts can be sorted into part families that have similar part design attributes (part shape, size, surface integrity, material type, raw material state) and/or manufacturing attributes (operations and sequences, batch size, machine and cutting tools, processing times) for a specific purpose. The purpose of the family determines the attributes to be considered. For instance, if part design advantages are to be gained, the part families are formed by aggregating parts having identical shapes or sizes. This allows existing drawings to be retrieved when new parts are introduced into the system. The classification can be done with respect to equipment type, geometric shape, design and operations or by similarity of equipment tooling. When the part variety is low, a visual/manual analysis by part and drawing can be employed to find part families. With high part variety, it becomes preferable to code all the parts and classify them by a code similarity or distance measure. A code is a string of characters that stores information about a part and is used to either group similar parts or to separate dissimilar parts. It is used in three coding systems: monocode (hierarchical code), polycode (attribute code) and mixed code. Using polycodes, distance measures (Minkowski, weighted Minkowski and Hamming distance measures) can be found between each one of the parts according to some selected manufacturing attribute (Singh and Rajamani, 1996). Although some cluster analysis procedures can be subsequently applied to form the part families using these distance measures, the coding and classification systems do not identify the set of machines required to process the parts. This is the main drawback to the usefulness of coding and classification schemes in CM design. An overview of coding and classification systems is reported in Askin and Vakharia (1991). MGI-related descriptive procedures consider the cell formation problem as having two-phases. In the first stage machines need to be grouped taking into account the information obtained from the part routings. In the second stage, the parts are allocated to the formed machine groups. PF/MG- related descriptive approaches are those that group parts into part families and machines into machine cells simultaneously. The earliest approach was proposed by Burbidge (1963) and is known as Production Flow Analysis (PFA). The latter finds a complete division of all parts into families and also a complete division of all existing machines into associated cells by analyzing the information present in the process routes of parts. More methods of this type have been developed: Component Flow Analysis (CFA) by El-Essawy and Torrance (1972) and a manual method called Nuclear Synthesis.

2.2.2. Cluster Analysis Procedures

Cluster analysis consists of many diverse techniques that are used to recognize structure in a complex data set. The main objective of such a statistical tool is to group either objects or entities or their attributes into clusters such that individual elements within a cluster have a high degree of "natural association" among themselves and such that there is very little "natural association" between clusters. Clustering procedures can be classified as: 1) array-based clustering, 2) hierarchical clustering, and 3) non-hierarchical clustering.

Array-based Clustering. Array-based clustering techniques form part of the most cited methods employed in CM design, where the processing requirements of parts on machines are represented by an incidence matrix known as the machine-part matrix. The latter has zero and one entries, where a "1" entry in row i and column j ($a_{ij}=1$) of the matrix means that part j requires machine i for one of its operations, whilst a "0" entry means that it does not. The array-based methods attempt to allocate machines to groups (cells) and parts to families to find a block diagonal form of the $a_{ij}=1$ entries in the machine-part matrix. This can be done by iteratively rearranging the order of the rows and columns. Several array-based clustering algorithms have been proposed: Bond Energy Analysis (BEA) by McCormick *et al.* (1972), Rank Order Clustering (ROC) by King (1980a, 1980b), ROC2 by King and Nakornchai (1982), Modified Rank Order Clustering (MODROC) by Chandrasekharan and Rajagopalan (1986a), Direct Clustering Algorithm (DCA) by Chan and Milner (1982), Cluster Identification by Kusiak and Chow (1987), and the Hamiltonian Path Heuristic by Askin *et al.* (1991). The drawbacks of array-based clustering are as follows. Operation sequence of parts is ignored in the machine-part matrix. The part-machine matrix cannot be used to identify all the operations of a part if the latter requires more than one operation. The matrix only refers to the machine types that are required by the parts so that multiple copies of the same machine type cannot be represented. Also ignored are machine capacity limitations (the main assumption in array-based techniques is that the machine type within the cell to which the part has been assigned has enough capacity to process the parts completely), production requirements, machine costs, part production costs and cell size limits. In addition, application of the array-based clustering technique has to be followed by manual/subjective human intervention via visual inspection to determine the cell composition. This is unpractical for problems of real-life sizes where large numbers of columns and rows need to be represented and visualized.

Hierarchical Clustering. The basis of hierarchical clustering is to define a measure of similarity (or dissimilarity) or distance coefficient between each pair of individuals, for example, parts, machines, tools, design features. For instance, the larger the value of the similarity coefficient, the more similar the two parts/machines are. The smaller the value of a dissimilarity coefficient, the more similar the two parts/machines are. This similarity (dissimilarity) coefficient is used to form part families and machine groups. In hierarchical clustering, the data in the machine-part matrix are not partitioned into groups or cells in one step. Rather they are first separated into a few broad cells, each of which is further divided into smaller groups, and each of these further partitioned , and so on until terminal groups are generated which cannot be subdivided. The similarity coefficient or distance measure is used to guide each successive partition. Hierarchical techniques are further subdivided into agglomerative and divisive methods. These methods are illustrated in figure 2-1 through a dendogram, where 5 machines are considered. A dendogram is a two-dimensional diagram that illustrates the fusions or divisions which have been made at each successive stage of the analysis. Agglomerative techniques proceed by a series of successive fusions of the M machines or the P parts into groups, ultimately reducing the data to a single cluster containing all the machines (parts). Divisive techniques partition the set of M machines or P parts successively into finer groups and will finally split the entire set of machines or parts into M cells or P cells, each containing a single machine or part.

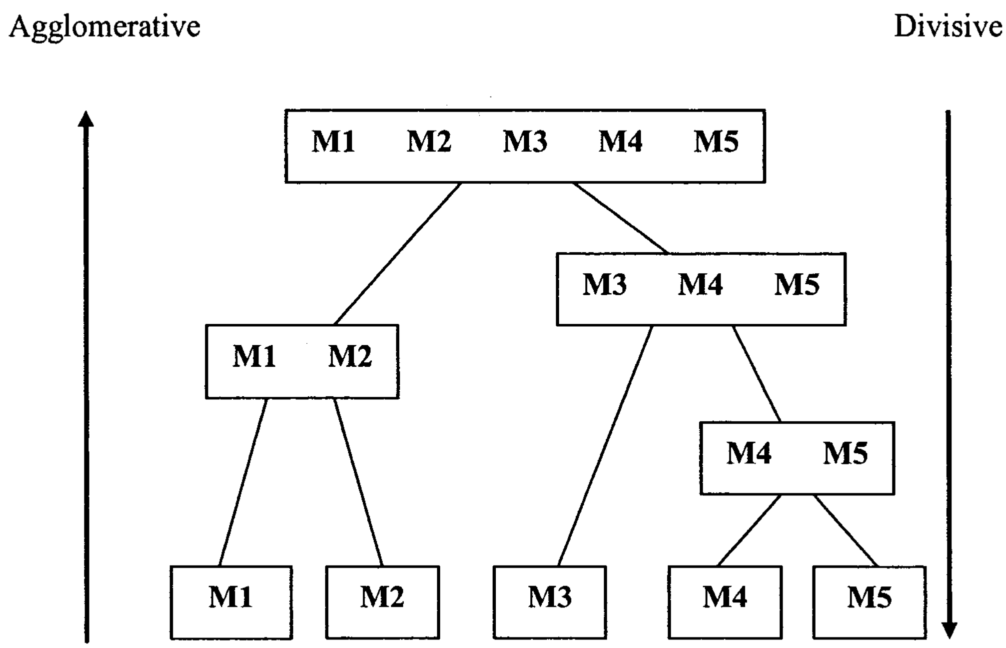

Figure 2-1 : A dendogram showing the hierarchical classifications

Agglomerative techniques are the mostly used methods for the cell formation problem. The most commonly used technique is Single Linkage Clustering (SLC) by McAuley (1972). A similarity coefficient is first defined between each pair of machines in terms of the number of parts that visit each machine. Since the matrix has binary attributes, four types of matches are possible: parts visit both of the machines, parts visit one machine (e.g. m) but not the other (e.g. n), parts visit one machine (n) but not the other (m) and parts do not visit any one of the two machines. A number of coefficients have been used as measure requiring the information provided in the part-machine matrix. Jaccard's similarity coefficient is the most commonly used one, defined as the ratio of the number of parts processed on both machines (m and n) to the sum of the total number of parts processed on machines m and n. The value of the Jaccard's coefficient ranges from 0.0

(maximum dissimilarity when the two machines do not process the same part types) to 1.0 (maximum similarity when two machines process the same part types). Other manufacturing features such as part volume, part operation sequence, tooling requirements, setup features, production volume and lead time can be considered while computing the similarity measure. The SLC algorithm proceeds where machines groups (two machines, a machine and a machine group or two machine groups) are joined. The similarity is computed at each partition so as to decide on the formation of new groups. This process continues until all machines form part of only one group. There are limitations to this technique within SLC; two groups are merged together merely because two machines (one in each group) have high similarity. If this process continues with machines that have yet to be clustered, a chaining problem occurs, known as the machine chaining problem. The latter happens when a bottleneck machine, which may have more common operations with machines in a group, has been assigned to another group. The reason why the bottleneck machine has been assigned to the latter is because the assignment has been based solely on the similarity it has with the other machines in the wrongly assigned group, ignoring the taking into account of any of its interactions with all the machines. Complete Linkage Clustering (CLC) is considered to be the other extreme of SLC in that it is the least likely to cause chaining. The algorithm for CLC is similar in most respects to SLC. However, the difference is that CLC combines two clusters at minimum similarity levels, rather than at maximum similarity levels as in SLC. Noting that SLC and CLC are clustering techniques based on extreme values, it is interesting to note the development of the Average Linkage Clustering technique (ALC), where clustering occurs by considering the average of all links within a cluster. However, ALC produces results that lie between the extremes of SLC and CLC, especially with regards to the machine chaining problem. Although the algorithms provide different sets of groups, they do not give the best way to group machines (e.g. machine chaining problem, no insight for the treatment of bottleneck machines). Furthermore, the part families have yet to be formed and get assigned to the machine groups.

Non-Hierarchical Clustering. In comparison to the hierarchical techniques, the non-hierarchical procedures allow objects to change group membership during the cell formation process. They are iterative methods that begin with either an initial partition of the data set or the choice of a few seed points, with the number of clusters (cells) decided a priori. However, arbitrary choice of seed points or of the initial data partition can lead to unsatisfactory results. For instance, the number of clusters has to be varied before each experimental run in order to evaluate the quality of the solutions obtained. Lemoine and Mutel (1983) developed a non-hierarchical technique that enables the automatic recognition of machine cells and part families. Chandrasekharan and Rajagopalan (1986b) developed the Ideal Seed Non-hierarchical Clustering (ISNC) algorithm for either machine cell or part family formation. It uses the group efficiency criterion, which measures intercellular movement and within-cell machine utilization. Chandrasekharan and Rajagopalan (1987) improved upon ISNC through the ZODIAC algorithm (Zero-One Data Ideal seed Algorithm for Clustering), which simultaneously identifies part families and machine cells. One limitation of ZODIAC is that improper selection of the initial seed solution can lead to the division of some good clusters and/or to cells that have single members. Another limitation is that the machine processing that is required by individual parts is not adequately represented by the basis used for clustering (minimum rectilinear distance). GRAFICS (Srinivasan and Narendran, 1991) has been developed to overcome some of the limitations of ZODIAC. It creates initial seed solutions by solving an assignment problem. The latter maximizes the similarity between machines. A non-hierarchical algorithm that uses the maximum density rule as the clustering criterion identifies sub-tours to obtain initial seeds. Further work includes those of Nair and Narendran (1998), where the algorithm form the cells and part families using sequence data, and of Ohta and Nakamura (2002), who improved the latter model with setup time reduction between machines in the same cell.

To conclude on clustering analysis procedures, it is first noted that the clustering algorithms are not affected by the definition of the similarity measure. Also, the availability of commercial software packages implementing the clustering algorithms makes the latter more attractive than the matrix manipulation algorithms. However, all of these methods are heuristics and are data dependent. The input data could be structured so that a pure diagonal form exists, or the data could be unstructured so that the latter cannot be decomposed to a pure diagonal form with non-overlapping elements. Therefore, use of any one of these heuristics warrants the importance of considering the nature of the input matrix.

2.2.3. Graph Theoretic Approaches

Graph theoretic approaches can be used to structure the cell formation problem in a number of forms through a number of methods including Graph Partitioning Algorithms, Bipartite Graphs, Minimum Spanning Tree, and Network Flow.

Graph Partitioning Approaches. The use of graph partitioning algorithms is characterized by machines/parts being represented as vertices (nodes) and the processing of parts as the arcs linking these nodes. Rajagopalan and Batra (1975) developed and applied a pure graph theoretic approach to the cell formation problem, where machines are treated as vertices and the similarity between the machines as the arcs joining such nodes (using the Jaccard's similarity coefficient). Using this graph partitioning approach machine cells can be formed by assembling cliques obtained from the graph. These cliques are obtained by determining the value of the Jaccard's coefficient between machine pairs; only arcs (edges) that have a Jaccard's coefficient value above a pre-specified threshold are accepted and used to form the cliques. The latter can then be merged to create cells in such a way that intercellular movement is minimized. Also, an upper limit on cell size can be set by limiting the number of machines that get assigned to cells. De Witte (1980) investigated the use of different similarity coefficients with the same approach. Askin and Chiu (1990) proposed a cost-based mathematical formulation and a heuristic graph partitioning solution approach. They adapted the Kernighan and Lin graph partitioning method and applied a two-phase partitioning algorithm. The first phase assigns parts to specific machines whilst the second phase groups machines into cells. Faber and Carter (1986) use a graph theoretic algorithm which converts the machine similarity matrix into a cluster network. This is used to group the machines and parts into cells. The cluster network is partitioned into cells by solving a minimum cost flow problem.

Bipartite Graphs. King and Nakornchai (1982) suggested the use of a bipartite graph where the parts and machines represent the two sets. The edges (arcs) between the two sets of nodes represent the requirement for machine *m* for part *p*. As such the problem is formulated as a *k*-decomposition problem in graph theoretic terms. A *k*-decomposition is obtained by deleting edges to obtain *k* disconnected graphs. The objective of the model is to determine optimally the edge(s) to be cut to make the graph into two disjoint sub-graphs.

Network Flow. Vohra *et al.* (1990) developed a network-based algorithm (a modified Gomory-Hu algorithm) that minimizes the amount of machining times performed outside the part primary cells (intercellular interaction). Wu and Salvendy (1993) developed a network model to partition the machine-machine graph into cells by considering the operation sequences. Lee and Garcia Diaz (1996) represented the clustering problem as a capacitated circulation network that measures the functional similarity between machines. The primal-dual algorithm developed by Bertsekas and Tseng (1988) was used to determine a complete loop as well as several sub-loops representing the machine cells. After determining the machine cells, parts or part families are assigned to the machine cells using other algorithms. Certain special properties of a network flow problem can be used to outperform certain mathematical programming approaches. In some cases, they can be used in conjunction with a mathematical programming model for better computational times. For instance, Nsakanda *et al.* (2006) integrated several other manufacturing issues to the cell formation problem. A model was mathematically formulated and solved using a hybrid genetic approach. The solution methodology involves partially solving the main model by applying mutation operators that determine the values for the decision variables related to the machine assignment to cells. The partially solved model is further modified by relaxing the machine capacity constraints. It is then observed that the fully modified model has a min-cost flow structure that readily lends itself to a Dantzig-Wolfe decomposition approach.

2.2.4. Artificial Intelligence-based Approaches

An increasing amount of research is being directed in the application of artificial intelligence (A.I) techniques to solve the cellular manufacturing problem. The techniques developed include syntactic pattern recognition, expert systems/knowledge base, fuzzy mathematics, artificial neural networks. Such approaches are relatively new in the design of CM systems.

Syntactic Pattern Recognition. This is a technique where items are presented as pattern structures that are able to consider more complex inter-relationships between features than the simple numerical feature vectors used in statistical recognition. Syntactic pattern recognition, also known as structural pattern recognition, can be used (as opposed to statistical recognition) when there is a clear structure within the patterns. The structure can be presented using the strings of a formal language. The differences in the structures of the classes are encoded as different grammars. Wu *et al.* (1986) applied syntactic pattern recognition to CM design, using the machine sequence data of the parts to be produced. By using analytical methods from formal language theory, the complex patterns (routing sequences of parts) are represented as strings of primitive characters (machine identifiers). The grammar of the language defines the rules to be used for constructing complex sub-patterns and patterns out of the primitives (simple primitives or prime sub-patterns). The analogy between manufacturing cells and the grammars used is as follows: each cell is able to speak a language, the language being the family of parts that it can produce. The assignment of new parts to cells means that a new set of complex patterns can be obtained from the routing sequences of these new parts. Using the new patterns and the defined set of rules, a recognition process takes place as the primitives are parsed, before assigning the new parts to the appropriate cells. Advantages include the fact that the cell formation process can consider material flow patterns and the operation sequence of parts. There is also the advantage of assigning a non-uniform importance to machines, which can be useful when dealing with bottleneck machines.

Expert Systems. Also known as knowledge based systems, they are the most successful and developed branch of artificial intelligence. Expert systems are computer programs that contain some of the subject-specific knowledge of one or more human experts. The most common type of expert systems is a program made up of a set of rules that analyze information (usually supplied by the user of the system) about a specific class of problems. The program also provides mathematical analysis of the problems and can recommend a course of user action to implement corrections. Such a technique has been applied to the cell formation problem by Kusiak (1988), Elmaghraby and Gu (1989), Luong. (1993). Kusiak (1988) implemented an approach where the machine-part incidence matrix is modified by replacing the binary data (whether part p needs machine m for processing) with data that represents the processing time of part p on machine m. The procedure involves the use of both a knowledge-based and a clustering algorithm. A heuristic that takes into account various production factors determines the machine cells and part families. A number of constraints are checked for the given cellular configuration. When any of these constraints is violated, certain rules from the knowledge-base are activated so as to change the assignments and make the configuration feasible. Actions may include the inclusion of new machines to increase production capacity or the use of other process plans. Elmaghraby and Gu (1989) presented an approach where domain specific knowledge rules are used in a prototype feature based modeling system. The latter automates the process of identifying part attributes and assigning the parts to the most appropriate manufacturing cell. Such a system is based on the geometric features of the parts, characteristics of the formed manufacturing cells, parts functional characteristics and attributes, as well as domain-specific manufacturing knowledge. Luong *et al.* (2002) proposed a knowledge-based system which makes recommendations during the CM conceptual design phase. These recommendations try to ensure system feasibility (based on the production quantity and product variety ratio), give options for the cell formation techniques to be used (classification and coding or production flow analysis) and give information on the types of cells to form (product-focused, process-focused, general-purpose or hybrid).

Fuzzy Logic. Fuzzy logic allows for set membership values to vary between and include 0 and 1. Specifically, it allows partial membership in a set. Most clustering methods assume that each part can belong to only one part family, that is, part families are mutually exclusive and collectively exhaustive. Some of the parts can be found to belong to certain unique part families but it is not always clear which part family is the most appropriate for other parts. Xu and Wang (1989) applied fuzzy mathematics to the cell formation problem where part features are transformed into fuzzy numbers through the use of membership functions. The latter treat part features in such a way that the resulting fuzzy numbers enable each part to be differentiated according to its processing requirements. A similarity coefficient matrix is formed using the fuzzy numbers and a part can belong to a family as long as long as the value of its similarity coefficient exceeds a certain level. Chu and Hayya (1991) applied a fuzzy *c*-means non-hierarchical clustering algorithm to production data. It has the same limitations as the other non-hierarchical clustering algorithms. The number of part families, *c*, has to be specified a priori. However, if the latter is underestimated, the result is far from optimal. A poor stopping criterion in the algorithm leads to inferior clusters but the technique is unaffected by exceptional elements. Chu and Hayya (1991) found that their approach is superior in terms of execution time and solution quality in comparison to the optimal 0-1 integer programming model. Although their algorithm was not as efficient as the heuristic that they used in their comparison, it provided more information than is available from the crisp values, (defining the cells and part families formed) obtained from the heuristic. Further work reporting the use of fuzzy mathematics in CM design can be found in Josien and Liao (2002), Lozano *et al.* (2002). Gungor and Arikan (2000) used fuzzy set theory for CM design through an algorithm that treats design and manufacturing attributes and operation sequences as input parameters when formulating the cell formation problem. Membership functions are used to fuzzify such parameters so that they can be used together with some IF-THEN decision rules with a view to determining the part relationships s fuzzy sets. The defuzzification step follows where crisp values for the part relationships are obtained. A traditional cell formation procedure, e.g. Single Linkage Clustering (SLC), can thus be applied by using the defuzzified part relationships chart as input.

Neural Networks. When a problem is well defined and self-contained, traditional serial computing is superior to the biological brain in finding a solution. However, biological brains are superior in dealing with problems involving a lot of uncertain and noisy data where it is important to extract the relevant items quickly. Neural networks models are used to mimic the way biological brain neurons generate intelligent decisions. A wide variety of applications are present, including speech recognition and diagnosis. Artificial neural networks (ANN) have also been successfully applied to many manufacturing areas including CM design because of their ability to learn through a training process by recognizing patterns and memorizing special features associated with input data. The ANN techniques can be divided into supervised and unsupervised learning approaches. The success of the applications is attributed to the fact that these techniques can learn by recognizing patterns and memorizing special features related to the input data. ANN has been applied through a supervised learning approach to the classification and coding problem based on the back-propagation learning algorithm (design-oriented approach).

This technique can also be applied to a production-oriented method to determine part families and machine cells. By taking a design-oriented approach Kaparthi and Suresh (1992) have applied artificial neural networks for classification and coding of rotational parts, using a three-digit part description whilst Liao and Lee (1994) have developed an automated GT coding and part-based computer-aided design (CAD) system. Malave and Ramachandran (1991) applied a modified version of the Hebbian learning rule to the cell formation problem. This technique belongs to the supervised learning category but is also a production-oriented method for forming part families and machine cells. Unsupervised learning approaches have also been applied to the cell formation problem and the methods include adaptive resonance theory (ART) and its numerous variants, fuzzy ART, competitive learning (Chu, 1993; Venugopal and Narendran, 1992a) and Kohenen nets (Venugopal and Narendran, 1992a). Unsupervised learning techniques are more suitable for the general clustering problem as neither the number of clusters (cells) nor the representative members of these clusters need to be specified at the onset. Once part families and machine cells are determined by the unsupervised model, a supervised model can be developed and trained to assign new parts to the existing cells. CM design research has also made use of a neural network classifier based on an unsupervised learning model (known as ART) developed by Carpenter and Grossberg (1987). ART and its variants (e.g. ART1) can be classified as non-hierarchical clustering methods and cluster the input vectors into separate groups based on similarities (Kusiak and Chung, 1991; Dagli and Sen, 1992; Kaparthi and Suresh, 1992; Rao and Gu, 1992, 1994; Liao and Chen, 1993; Chen and Cheng, 1995). Another variant of ART, called fuzzy ART, implements fuzzy logic into ART's pattern recognition and enhances generalization. It also handles both analogue and binary inputs while assimilating and utilizing new learning laws (Suresh and Kaparthi, 1994; Burke and Kamal, 1992).

2.2.5. Mathematical Programming

Mathematical programming approaches for the cell formation problem consist of the formulation of linear or non-linear integer programming problems. They present the distinct advantage of being able to incorporate some design logic within the objective function and constraints, for example, ordered sequences of operations, alternative process plans, non-consecutive part operations on the same machine, setup and processing times, the use of multiple identical machines as well as outsourcing of parts. Selim *et al.* (1998) classified these techniques as linear programming (LP), linear and quadratic programming (LQP), dynamic programming (DP) and goal programming (GP). The classical clustering p-median model is the first model to form part families using mathematical programming. It is used to identify f part families (machine cells) optimally, such that the distance between parts (machines) in each family (cell) is minimized with respect to the median of the family (cell). Each part (machine) can only belong to one part family (machine cell). The number of f families (cells) has to be specified a priori so that the model selects f parts (machines) as medians and assigns the remaining parts (machines) to these medians such that the sum of distances in each part family (machine cell) is minimized. Unlike the hierarchical clustering algorithm, the p-median model allows parts (machines) to be moved between families (cells) when searching for optimality. As observed, the formation of part families and machine cells cannot be done simultaneously. A sequential approach is needed when using the p-median model. Also, it is not known how to identify values of f that result in the formation of good diagonal blocks. Finally, the p-median model relies on the assumption that each part has a single process plan. Kusiak (1987) modified the model into a generalized p-median model where alternate process routings can be taken into consideration. Subsequently, several authors further modified the generalized p-median model and reported successful applications (Ribeiro and Pradin, 1993; Viswanathan, 1996; Lee and Garcia-Diaz, 1996; Wang and Roze, 1997; Deutsch *et al.*, 1998). However, only small-scale problems were solved. Medium-sized problems required prohibitive computational time while there has been little reported about large-scale problems that represent real-life industrial instances. Also, off-the-shelf optimization softwares do not have enough solving capabilities to solve the cell formation problem since the latter is NP-hard. To avoid the problem of having to determine the optimal value of f, Srinivasan *et al.* (1990) proposed an assignment model where a sequential procedure identifies machine cells followed by the identification of part families. The objective of the assignment model is to maximize the similarity (same as that used in the generalized p-median model). Sub-tours (closed loops) are identified when solving the model and form the basis for grouping parts and machines. The algorithm consists of two steps. If the matrix is mutually separable (parts are assigned to machine cells such that they are disjoint), the procedure stops after the first stage. If the solution results in the formation of exceptional elements, the second stage is activated and another assignment problem is solved to determine the part families (the part grouping model from stage 1 is used). These part families are then assigned to machine cells while minimizing the number of exceptional elements and voids. Shtub (1989) developed the cell formation problem as a generalized assignment problem (GAP) and proved that the proposed model is equivalent to the general formulation of the GT problem (p-median problem) and to the generalized GT problem (the generalized p-median model). The proposed model was solved by a branch and bound procedure developed by Ross and Soland (1977). Purcheck (1974; 1975) was among the first researchers to apply linear programming to group technology. This has been achieved since cluster analysis represents a linear programming problem where the objective is to maximize the total sum of similarities between each pair of individuals (parts or machines) or to minimize the distances between each pair. According to Kusiak and Chow (1988), the distance between each pair can be an asymmetric function, and the Minkowski, the weighted Minkowski and the Hamming distance measures are the ones that are the most commonly used in the cell formation problem.

The discussion so far on CM design methods considers indirect measures, for example similarity/dissimilarity, bond energy, ranking, distance, when solving the cell formation problem to obtain a block diagonal form. Part families and machine cells are identified while minimizing the number of exceptional elements and voids. However, one needs to better consider the costs related to voids and exceptional elements, as such costs vary for different part/machine combinations. The procedures so far also decouple the cell formation process and any cell evaluation procedure. Boctor (1991) developed a CM model which simultaneously assigns machines and parts to cells. The model objective function minimizes the number of exceptional elements. The model, however, ignores machine costs, capacity and duplication. Choobineh (1988) developed a linear programming model that uses a similarity measure based on part operations and sequence, taking into account machine cost and capacity but ignoring the presence of alternate process routings. Adil *et al.* (1993) presented a CM model that incorporates certain machine operational aspects during the cell formation: machine investment and operational costs. Rajamani *et al.* (1996) developed a mixed integer programming model that considers multiple process plans for part production. The objective function of the model minimizes the sum of investment, process and material handling costs as a weighted sum of the three different cost functions. Song and Hitomi (1996) have formulated a mixed-integer programming (MIP) model that allows inventory and layout adjusting (system reconfiguration) for multiple periods (with varying dynamic deterministic product demand). Their proposed MIP model is solved to optimality using the Bender's decomposition approach. Heragu and Chen (1998) also applied the Bender's decomposition approach to optimally solve their proposed CM model. Steudel and Ballakur (1987) developed a two-stage heuristic to solve the cell formation problem, where the first stage uses a dynamic programming (DP) approach to determine the sequence or chain or machines that maximizes machine similarity. The second stage then partitions the maximum machine chain into individual cells.

Goal programming (GP) has been applied mostly for multi-criteria CM design. Sankaran (1990) considers multiple goals in the cell formation procedure by first solving a single model which has the sum of five distinct cost functions as its objective function. The optimal total cost is broken into two cost aspiration levels, namely operating and capital investment cost. These two costs, along with five other goals (minimum similarity of parts based on their required machines and tools, available machine capacity, minimum and maximum number of total parts movement), are then combined within a linear integer goal programming model. Shafer and Rogers (1991) present CM goal programming models that are applicable in three types of situations, namely, when setting up a new CM system and purchasing all the required equipment, reorganizing an existing layout without new equipment procurement, reorganizing the layout with both existing and new equipment. The criteria considered are to minimize setup times (through parts sequencing), minimize intercellular movement, minimize investment in new equipment and maintain acceptable machine utilization levels. Thus, the GP models combine the p-median (for identifying part families) and the traveling salesman problem (for determining the optimal sequence of parts). Realistically-sized problems are solved by using a heuristic.

However, because of the way that the CM models are formulated, certain limitations apply to the mathematical programming approaches. Since the objective function can be of a nonlinear form, most approaches do not concurrently group machines into cells and parts into part families. Also, the number of machine cells has to be specified a priori and this has an affect on the grouping process and can potentially obscure natural cell formations in the data. Since some or all of the variables are constrained to integer values, most of these models are computationally intractable for realistically sized problems. It has been shown that large scale problems typically require that the model be subject to model-specific linearization procedures and/or be solved using approximate methods such as Lagrangian relaxation with subgradient optimization, simulated annealing, genetic algorithms, tabu search or a hybrid of any of these methods. Therefore, the major drawback of mathematical programming approaches in CM design is the adverse computational time and effort required to solve real-life sized problems.

2.2.6. Meta-heuristic Approaches

Traditional heuristic procedures were seen as relying on either some clever rule of thumb or on an iterative rule that terminates the search as soon as a new solution does not improve the last one that was found. These iterative heuristics are often referred to as descent or ascent methods. Furthermore, optimization algorithms may yield a global optimal solution in a possibly prohibitive computation time. An approach is needed that may be fast and assures a good solution to the problem, especially for typical industrial sized problems. A growing number of researchers have adopted the use of meta-heuristics ('smart heuristics') for large combinatorial problems. Meta-heuristics draw on ideas and concepts from another discipline to help solve the artificial system being modeled; they incorporate concepts based on biological evolution, intelligent problem solving, mathematical and physical sciences, nervous systems, and statistical mechanics. They are able to search large regions of the solution space without being trapped in local optima. Most meta-heuristics are naturally discrete as opposed to conventional methods which are most naturally continuous. This means that meta-heuristics can handle models with integer variables, discrete variables and/or logical (binary or zero-one) variables well. Another advantage is that of flexibility: the range of models (in particular those with the complicating factors mentioned above) that can be solved by meta-heuristics is far greater than by conventional methods. Problem-specific knowledge can be more easily integrated into the solution process – e.g. non-standard goals, constraints, objectives and conditions. The popularity of meta-heuristics is further explained by the fact that they have been successfully used to solve a wide range of optimization problems, especially combinatorial problems, whilst yielding an approximate solution in an acceptable computational time. Jones *et al.* (2002) reviewed 115 articles concerned with the theory and application of meta-heuristics. They concluded that theoretical papers account for only 20.9% of the articles concerned. This healthy ratio indicates that these techniques have a lot of real-world usage rather than just theoretical value. 70% of the work surveyed utilize GA as the primary meta-heuristic, 24 % use SA whilst only 6% draw on TS. TS is more frequently used in conjunction with either GA or SA as a secondary meta-heuristic refinement in order to strengthen the avoidance of convergence at local optima (enhancing global optimization). A possible disadvantage is that there are a larger number of parameters to be set by the modeler in meta-heuristics. The solution is sensitive to these parameters; so a number of executions of the meta-heuristics (with different parameter settings) might be required before a good solution is produced. In other words, meta-heuristics can be considered to be "poor black boxes"; they become more difficult to apply when only a single run is allowed due to time or other pressures. It has to be also noted that these meta-heuristics might be sensitive to the initial solution (size and diversity of the initial population), the 'groupability' of the input machine-part matrix and the number of cells specified. The rest of this section elaborates on the use of meta-heuristics in general for the CM design problem and provides a review of the main meta-heuristics that have been applied.

Ballakur (1985) has shown that the problem of CM design is NP-hard, even under fairly restrictive conditions, as it cannot be solved within a reasonable amount of time using a polynomial-time algorithm. The combinatorial nature of the CM design problem exists because the latter involves a complex, multi-criteria and multi-step process. The NP-hardness and combinatorial complexity of the CM design problem is widely reported in literature (Singh and Rajamani, 1996; Selim *et al.*, 1998). In spite of the NP-completeness of the grouping problem and the existence of local minima, meta-heuristics present promising solution techniques for large-scale problems: they do not make strong assumptions about the form of the objective function as do many other optimization techniques. Also, the objective function is independent of the algorithm, that is, the stochastic decision rules. Multiple criteria objective functions including setup time requirements, tooling and crewing requirements, alternative routings, cost of machines, inter-cell transfers and reduced machine utilization are needed in order to obtain satisfactory algorithmic results. The only objective function requirement is that the meta-heuristic maps the solutions into a partially ordered set. This presents the flexibility of interchanging various objective functions and of utilizing multi-criteria objective functions. Convenient substitution of various evaluation functions would then allow the system designer to generate and review alternative cell designs quickly. One can verify in various surveys (Wemmerlov and Hyer, 1987; Wemmerlov and Hyer, 1989) that there is a tendency to group both machines and parts simultaneously. This practice is conceptually appealing but typically results in over-complicated models and contradicts with the staged approach suggested by Burbidge (1975) for clustering problems in GT. However, a staged approach is not compatible with models that have alternative routings being available a priori. As a result, most of the existing optimization methods for the cell formation problem oversimplify the problem (e.g. with network flow models, p-median models and minimum spanning tree formulations), and these cannot deal with instances of practical size using reasonable computing resources, especially with the large variety of parts present in modern production environments. All the complexity in real life cannot be formulated through mathematical models. So a set of reasonable solutions is much more useful as this allows the decision-maker to select the most appropriate one for some specific situation (Solimanpur *et al.*, 2004). These justify the heightened use of meta-heuristics to address integrated and increasingly elaborate/complex forms of the CM design problem. This also explains the motivation with respect to applying meta-heuristics to an exceptionally complex proposed model. The meta-heuristics that have been extensively adapted to solve the CM design problem are, namely, genetic algorithm (Venugopal and Narendran 1992b; Gupta *et al.* 1995; Joines *et al.* 1996; Nsakanda *et al.* 2006), simulated annealing (Venugopal and Narendran 1992c; Chen *et al.* 1995; Mungwattana 2000) and tabu search (Logendran *et al.* 1994; Dake *et al.* 1995; Vakharia and Chang 1997; Lozano *et al.* 1999). Genetic algorithms (GA), simulated annealing (SA) and tabu search (TS) have been increasingly used to solve CM design models, again with the majority of papers dealing with GA and SA implementations. In comparison, few papers deal with the full use of all the features of TS for CM design. Therefore, this warrants further investigation of the full capabilities of TS in performing CM design.

Genetic Algorithm. Genetic algorithms (GA) emulate the way species breed and adapt in the field of genetics. Genetic algorithms have the property of implicit parallelism; they do not evaluate and improve a single solution but analyze and modify a population of solutions simultaneously. The ability of a GA to operate on many solutions simultaneously and to gather information from all the current points to direct the search mitigates the problem of local optima. These two features enable a GA to tackle even NP-hard problems successfully. Genetic algorithms have proved to be effective and flexible as optimization tools that can produce optimal or near-optimal solutions. The objective function is independent of the algorithm. This offers the flexibility to interchange various objective functions and to utilize multi-criteria objective functions. Convenient substitution of various evaluation functions allows the system designer to generate and review alternative cell designs quickly. To move toward a satisfactory algorithmic result, multiple criteria objective functions that include set up time requirements, tooling and crewing requirements, alternative process routings, cost of machines, intercellular transfers and machine utilization levels are needed. The inclusion of functions, either constraints or objectives, of difficult forms (e.g. non-convex functions), is possible because these functions will not be directly manipulated in the production of each new generation. The advent of parallel computing techniques has made genetic procedures even more attractive because their structures are particularly appropriate for such implementations. Industrial data sets are often too large for visual methods to associate machine cells and part families effectively. Genetic algorithms can form machine cells and part families simultaneously and avoid visual inspection of the data. Further exploration of genetic algorithm capabilities makes practical solutions to industrial scale problems more realistic. Venugopal and Narendran (1992b) used a genetic algorithm to solve a bi-objective integer programming cell formation problem. The uniform mutation and simple crossover operators are used to minimize the total number of intercellular moves and the total intracellular workload variation. A different population of solutions was employed for each of those objectives. To avoid trivial solutions, the constraint of having at least two machines per cell has been included in their model. Their solution representation is such that each machine in the plant corresponds to a gene in the chromosome. The value of the gene defines the owing cell of the respective machine. The total number of cells in the plant is pre-determined, but the formulation of the problem considers the processing time of parts, which is a significant improvement in comparison to the traditional cell formation methods. Gupta *et al.* (1995) developed a genetic algorithm to minimize the weighted sum of intercellular and intracellular moves. An acceptable level of machine utilization is considered to assign parts to manufacturing cells. Gupta *et al.* (1996) also considered different cell layouts and used a genetic algorithm to minimize the total number of intra-cell and inter-cell moves and cell load variation. Dimpoulos and Zalzala (1998) proposed an evolutionary algorithm for the cell formation problem of a pharmaceutical company. Both the representation of the solution and the genetic operators were purpose-based. Different multi-objective optimization methods were compared on the solution of the problem. Moon and Kim (1999) used a genetic algorithm to maximize the total number of parts flowing between the machines within the same cell. They considered different manufacturing data such as production volume, cell size and the capacity of the material handling device. Zhao and Wu (2000) used a genetic algorithm to solve a multi-objective cell formation problem. User-defined weights are used to convert multiple objectives into a single objective. The objectives in their research are: total number of exceptional elements, the total within-cell workload variation and the total intra-cell/ inter-cell movements, with alternative process routings.

These factors complicate the machine grouping problem but also generate better solutions as they result in more paths being chosen to achieve the desired objectives.

Simulated Annealing. Simulated annealing (SA) emulates the way in which a material cools down to its steady state in the field of physics. Since its introduction by Kirkpatrick *et al.* (1983), simulated annealing has gained popularity in solving hard combinatorial problems. Simulated annealing is not only a highly effective and general random search method to obtain near-global optimal solutions for optimization problems, but is also quite an appropriate method for solving the machine-part cell formation problem which is an NP-hard complex problem. The procedure is extremely adaptive, flexible and efficient and can be used to solve real machine-part cell formation problems in factories by providing a good manufacturing cell formation in a short execution time. Chen *et al.* (1995) applied an SA-based heuristic to minimize the number of intercellular moves to form manufacturing cells. Boctor (1996) explicitly considered the main elements of manufacturing cost in designing a CM system and developed an algorithm based on SA to solve the machine-part cell formation problem. Both of these researches consider some different factors including machine capacity, part demand, machine duplication cost, material handling cost and operation time. However, some other issues like operation sequences of parts, intercellular machine loading balance and the impact of cell layout were not addressed, thereby limiting the implementation of their approaches. Su and Hsu (1998) used a modified simulated annealing approach with the merits of a genetic algorithm (called a parallel simulated annealing) to solve a multi-objective cell formation problem. They considered the following objectives: machine investment cost, intercellular and intracellular material handling costs and inter-cell and intra-cell machine imbalances. They first construct the machine-part cell formation problem based on production flow analysis under realistic conditions. A PSA-based procedure is then introduced to solve the machine part cell formation (MPCF) problem. User-defined weights are used to convert the multiple objectives into a single objective. Safaei *et al.* (2007) developed a CM model that incorporates multi-period planning, dynamic system reconfiguration and alternate process plans. The proposed model was solved using a hybrid mean field annealing-simulated annealing (MFA-SA) approach.

Tabu Search. Tabu search (TS) is a powerful optimization technique used in a variety of settings including telecommunications, logistics, financial planning, transportation and production (Pirlot, 1996; Glover and Laguna, 1997). Its basic form originates from the work of Glover (1990) and its popularity is due to the fact that it enables the search for solutions to escape from local optimality especially in combinatorial problems. The neighborhood of an actual solution is explored through move operators so as to obtain better solutions. The next considered neighborhood solution can be accepted even if its objective cost is worse than the current solution. Only neighbors that are classified as tabu are forbidden. A neighborhood solution can be classified as tabu if it belongs to a *tabu list*. The *tabu list* stores certain attributes of recently visited solutions, which determine whether new solutions are accepted or not. Thus moves that reverse an actual solution to an already visited state (that is cycling) are prevented, that is, the move is classified as tabu active. The idea is that solutions of a higher cost are still considered as long as they lie in unexplored regions of the solution space. However, there is also the aspiration criterion which over-rides the tabu active status of a move when this move generates a new solution cost that is better than the best solution cost found so far. Effective use of this meta-heuristic is reported in many papers and such a review shows that good tabu search implementations require good definitions for the neighborhood structure, move operators (used to generate neighborhood solutions from a current solution) and aspiration criterion (Lozano *et al.*, 1999; Salhi, 2002; Raza *et al.*, 2006). Other tabu search parameters must also be set: the size and nature of the *tabu lists* as well as the size of the *candidate list* (Glover and Laguna 1997). The use of a varying *tabu list* size (Salhi 2002) and/or of a long term tabu search scheme (Glover and Laguna 1997) is seen as a good research avenue to solve such CM models. Most of the tabu search schemes used to solve the CM problem are short term in nature, with fixed-size *tabu lists*, and make limited use (or even no use) of long-term memory components.

Dake *et al.* (1995) compared the use of their tabu search approach to that of using a greedy local search procedure. Their CM model involved the minimization of intercellular material movement between cells, with upper bounds on the number of cells and cell size limits. By modeling their CM model as such, they showed that their problem is equivalent to a graph partition problem, which is known to be NP-hard. By the equivalency of the two problems, their CM model is also NP-hard. They developed an algorithm that creates an initial feasible cell configuration, under the constraints of cell size and numbers – this is the construction phase. Their approach is to carry out improvements on this initial cell configuration by using a tabu search technique – this is the improvement phase. This tabu search procedure includes two move operators – a single move and a double move. A single move involved moving a machine from one cell (source) to another (destination) while respecting the constraints of cell size and numbers. A double move occurs when a single move is followed by another single move which involves moving a machine from the destination cell to the source cell. This is done when the upper bound on the destination cell size is exceeded. A tabu list is implemented as a circular queue on a first-in-first-out (FIFO) basis. The tabu list is updated with information on the machine(s) moved, source cells, destination cells and the move made. This prevents the search from applying moves that generate already visited solutions. In other words, the tabu search does not reverse recent moves and prevents cycling. Their computational results shows the superiority of tabu search in solving problem sizes of up to sixty machines and six cells for a single planning period.

Lozano *et al.* (1999) developed a CM model to minimize a weighted sum of intercellular moves and intracellular voids to keep part families homogeneous. A short term tabu search variant is used, where the basic tabu search is enhanced by using a look-ahead scheme. Instead of selecting the single best non-tabu move as the next move, several candidate moves are tried. This is done to check whether the corresponding next move improves the objective function value. Using this tabu search procedure, they solve problems of up to thirty part types and sixteen machine types. They compare the quality of their results and computational times against two simulated annealing algorithms, one other tabu search approach (Dake *et al.* 1995) and three heuristics, namely ZODIAC and GRAFICS (non hierarchical clustering approaches) and MST (maximum spanning tree heuristic – a greedy heuristic capable of handling cell size constraints). Their approach consists in using tabu search to generate and explore feasible machine cell configurations, that is, the neighborhood of a possible solution. The corresponding part families are then found using a linear network flow model before obtaining an objective function evaluation of each cell configuration. Swap moves consist in changing the cell configurations into other feasible ones. These are done in four ways: two machines belonging to different cells are exchanged; a machine is deleted from a cell and placed into another provided that the maximum size of the destination cell is not exceeded; two separate cells with their machines are joined together to form one new cell; one cell is split into two separate ones. They use three tabu lists to record these moves, with each list having its own aspiration criterion. Experimental design is used to determine the best candidate and tabu list sizes.

Cao and Chen (2004) have used an embedded optimization technique to transform the CM problem they proposed into a parametric linear programming problem. A tabu search algorithm was used to solve the problem, where, under certain constraints, an initial feasible solution is generated to start the search. Swap operations are employed to search the solution neighborhood. Three types of swap moves are made. The first one involves changing the cell in which an operation of a part is made, keeping other decision variables constant. The second one involves changing the decision of whether to start part production, keeping other decisions constant. The third move is to change the number of machines assigned to a cell while keeping all other decision variables constant. The third type of move is performed via three actions: add one more machine of a type to a cell, remove one machine (keeping the number of machine assigned to the cell at non-negative values), remove all machines of this type from the cell concerned. The neighborhood is defined as the ensemble of these swap moves. The stopping criterion is activated after a predetermined number of iterations or non-improving iterations. Fixed *tabu list sizes* are used, the number of maximum non-improving iterations is pre-set as is the maximum number of total iterations. They reported that the quality of their initial solution highly affected the quality of the best solution found. Initial solutions that are far from the optimal or sub-optimal solutions required larger tabu list sizes to reach good quality final solutions. The largest model that was solved consisted of five machine types, four possible cells, five parts with each one having two to four operations. Further work on the use of tabu search to solve a CM model that integrates system configuration with probabilistic product demands was later presented by Cao and Chen (2005). A two-stage tabu search was used to solve a mixed integer programming model. Their work showed that as problem size increases, the efficiency of a short term tabu search decreases. With larger problems, more iterations are required and this leads to solution deterioration. The *tabu list sizes* had to be adjusted to fine-tune the tabu search.

Therefore, it is seen that each one of the three meta-heuristics can be adapted to solve CM models. However, they each have distinguishing features that need to be discussed and considered prior to implementation. These key features include whether adaptive memory is used, the type of neighborhood exploration used and the number of current solutions that are carried after each iteration. The use of genetic algorithms can consist of a "memoryless" and a population-based approach. To improve a given solution, GA approaches can either employ some systematic neighborhood search or rely on random sampling. Simulated annealing procedures are also "memoryless" with the possible use of both systematic neighborhood search and random sampling. However, SA implementations differ from the GA ones in that the method moves from one current solution to the next after each iteration and does not keep a population of solutions. Tabu search is the only one of the three meta-heuristics that employs adaptive memory. It can keep track of a list of solutions that has been explored together with their attributes, thus preventing the search from cycling back to previously visited solutions. Also, it employs a systematic neighborhood search scheme, with the possibility of either moving from one current solution to the next one or of taking a population-based approach. However, the full potential of TS has yet to be explored with respect to CM design. For example, most of the TS work done with regards to the CM design problem tends to only use the short-term aspects of the search (with *tabu list* and *candidate list sizes*). TS can become considerable more powerful if long-term memory and its associated strategies are included when solving CM design problems. Furthermore, the use of variable *tabu list* and *candidate list sizes* (as opposed to those of fixed sizes), search intensification and diversification schemes, as well as the strategic use of elite solutions and better stopping criteria remains to be explored. These aspects of tabu search form part of the contributions of this research in CM design (specifically in terms of CM solution methodology) as a tabu search framework is developed that consists of the aforementioned features.

2.3. Manufacturing Attributes considered in CM Design

This section presents the issue of considering real-life manufacturing attributes within the cell formation problem when performing CM design. Then a review of recent literature on CM models that integrate various production aspects is presented, allowing the identification of several important manufacturing attributes. The design of CM systems is a multi-criteria and multi-step process. Before implementing a CM system, a company needs to invest adequate effort during the planning and design phase because the benefits of CM are only possible by sufficiently incorporating the structural and operational features of a manufacturing plant within the CM design decisions. Therefore, integrated models can be used by designers to evaluate cellular layouts considering various aspects of the manufacturing operations. The models developed in this research are briefly presented in this section so as to demonstrate the manufacturing attributes that are covered herein. Such attributes are used to represent the important structural and operational aspects that the cell designer has to take into consideration when forming machine cells and part families. As already mentioned, the design of a cellular manufacturing system consists of three main stages (Dimopoulos and Zalzala 1998). This research deals with the first phase of the CM design problem, namely, the cell formation problem. Wu and Salvendy (1993) draw attention upon the fact that the design of CM systems must address the need for many important production factors to be considered when the cells are created: machine requirements, machine setup times, machine utilization, machine workload, alternative part process routings, machine capacities, part operation sequences, setup cost and cell layout. Mansouri *et al.* (2000) reviewed the literature with respect to the manufacturing criteria that need to be considered during CM systems design. They reported that pressures from the market provide a number of conflicting criteria on which performance is evaluated. Thus, the design of any CM system is critical to the efficient performance of the business. The objectives that are usually considered in the cell formation problems are either performance or cost-oriented. They put emphasis on the following objectives and constraints under which a typical CM system should be designed:

- Minimize the intercellular material handling cost (or maximize cell independence). The primary focus of GT is to identify cells where the interaction between cells is restricted.
- Minimize investment in equipment. In reorganizing a job shop into a GT cell system, there is typically an increase in the number of machines required. Hence, this objective focuses on minimizing additional equipment investment.
- Maintain acceptable equipment utilization levels. A cell system design is feasible if the equipment utilization in each cell is less than the maximum acceptable level.
- Identify cells of a "reasonable" size. The size of the GT cell will have an impact on how easily the cell can be managed and controlled. Hence, cells identified should not contain more than a specified number of machines.

Evidently, there is some inherent conflict between the objectives. For instance, cells can always be created without intercellular parts movement, simply by adding machines as and when required. However, this would create high machine and equipment investment costs. Most procedures developed for the cell formation problem do not have explicit objectives tied to them. Some cell formation techniques have explicit or implicit objectives, such as the minimization of intercellular movements that do not necessarily produce the best overall cell performance or satisfy application-specific objectives.

Ballakur and Steudel (1987) also provided a list of the important objectives to be considered in CM design, namely minimization of intercellular material handling costs, minimization of setup times, maximization/minimization of a similarity/ dissimilarity measure, minimization of total production cost, minimizing the number of exceptional elements, maximizing the utilization of machines and minimizing machine idle times. In addition to these, several real-life production factors have to be considered, for instance the presence of multifunctional machines, alternate part process routings, machine capacities and workload. Table 2-1 shows a survey of 20 recently published articles together with the manufacturing attributes (referenced by the caption) that are incorporated within each one of the developed models. The models developed in this research (models **A** and **B**) are also presented in table 2-1, showing that the proposed models in this research cover an extensive amount of the manufacturing attributes that are important in CM design and integrate more of these than previous models. A review of a few key models is now presented in more detail to not only describe the manufacturing criteria that were considered but to also discuss the solution approaches used.

Table 2-1 : Review of manufacturing attributes used in CM design

Models/Manufacturing Attributes	1		2	3	4	5		6	7		8						9		10	11	12	13
	a	b				a	b		a	b	a	b	c	d	e	f	a	b				
Proposed model **A**	X	X	X	X	X		X	X		X	X	X	X	X	X			X		X	X	X
Proposed model **B**	X	X	X	X	X		X	X		X	X	X	X	X	X	X		X	X	X	X	X
Caux *et al.* (2000)	X										X	X	X				X			X	X	
Chen (2001)	X				X		X						X									
Cao and Chen (2005)	X					X					X	X	X								X	
Chen and Cao (2004)	X				X		X				X	X	X								X	
Das *et al.* (2006)	X						X	X				X	X							X	X	
Defersha and Chen (2006)	X			X			X	X		X	X	X	X	X	X	X	X			X	X	X
Jayaswal and Adil(2004)	X										X	X	X				X			X		
Gupta *et al.* (1996)	X	X										X										
Mungwattana (2000)	X				X	X	X	X		X	X	X	X		X		X			X	X	
Nsakanda *et al.* (2006)	X	X		X							X	X					X			X	X	
Safaei *et al.* (2007)	X	X					X	X		X	X	X	X	X	X		X			X	X	
Selim *et al.* (1998)	X			X							X	X	X							X		
Solimanpur *et al.* (2004)											X	X	X				X			X	X	
Song and Hitomi (1996)	X	X			X		X	X		X		X										
Spiliopoulos and Sofianopoulos (2003)	X			X							X						X			X		
Su and Hsu (1998)	X	X						X			X	X	X							X	X	
Uddin and Shanker (2002)	X											X								X		
Vakharia and Chang (1997)	X										X	X										
Yin and Yasuda (2002)											X	X								X		
Zhao and Wu (2000)	X	X									X	X								X		

Caption to Table 2-1: Important design attributes for CMS design

1a. Intercellular material handling cost 1b. Intracellular material handling cost	4. Inventory holding in production planning	7a. Robust cell configuration 7b. Agile cell configuration	10. Formation of part *contingency* routings.
2. Part internal production cost	5a. Stochastic demand requirements 5b. Deterministic demand requirements	8a. Machines with multiple copies; 8b. Machine with limited capacities; 8c. Machine operating cost; 8d. Machine maintenance and overhead cost; 8e. Machine relocation and cost; 8f. Machine procurement	11. Part operation sequence and processing times
3. Subcontracting cost	6. Multi-period planning	9. Alternate routings: a. Chosen from user-specified routings; b. Chosen from all possible options based on operation and machine type	12. Cell size limits – upper and lower bound
13. Machine Procurement			

Throughout the literature review and mathematical model development, emphasis has been put upon the works of Song and Hitomi (1996), Mungwattana (2000), Chen and Cao (2004), Cao and Chen (2004, 2005), Nsakanda *et al.* (2006), and Defersha and Chen (2006) as they discuss the important issues of CM design that are further addressed in this research.

Song and Hitomi (1996) have integrated production planning within CM design. A mixed integer programming (MIP) model is formulated that allows inventory and layout adjusting (system reconfiguration) for multiple periods (with varying dynamic deterministic product demands), thereby addressing the issue of flexibility to meet diversified market needs. Therefore, the costs considered include: inventory holding, group setup, material handling (intercellular and intracellular) and layout adjustment. Machine capacity as well as part demand satisfaction (taking into account inventory held) constraints are included and their proposed MIP model is solved using the Benders decomposition approach. The model does not consider alternative part process routings and the outsourcing option.

Mungwattana (2000) focused on dynamic and stochastic production requirements with alternate part process routings. Emphasis is put on the fact that with shorter product life cycles and time-to-market, the product mix and part demand are likely to change frequently. Owing to this, two models working in a dynamic environment are developed for multi-period planning: one where the demands are uncertain (stochastic) and the other where the demands are deterministic. These models bring various benefits with respect to

CM design, in particular the consideration of cellular reconfiguration from period to period. The models also consider the machine allocation problem, machine operating and amortized costs together with machines that have multifunctional capabilities. However, the models ignore intracellular material handling, production planning (inventory holding) and outsourcing costs. The dynamic deterministic version of the model is solved via an optimal solution procedure. Following this, a dynamic stochastic version is formulated so that a small-scale problem is solved via Extended Lingo, followed by large problems being solved by simulated annealing.

Chen and Cao (2004) have included the aspect of production planning within CM design. An integrated approach was taken to study a production planning problem over a certain planning horizon in the designed CM system, with fixed charge cost. Their model features quadratic terms for intercellular material handling, decisions for manufacturing cell construction and production planning decisions such as times to start part processing and levels of finished part inventory. The objective is to minimize the total cost including intercellular material handling cost, fixed cell set-up cost, production set-up cost and product inventory cost in the system. By assuming that there are single process plans, limited machine capacities and multiple machines of the same type, their model decides whether cells are to be formed and how many of these, which machine types and how many to be allocated to the formed cells, the best time periods for part processing in a production planning horizon of multiple time periods. The model is solved by tabu search.

Nsakanda *et al.* (2006) developed a model that considers all of intercellular, intracellular and outsourcing costs. The following manufacturing attributes are included in their model: part demands, machine capacity limits, multiple process plans and alternative process routings for each part type, the processing sequence of parts, the trade-off between intercellular and intracellular costs and the option of outsourcing. They do not take into account the presence of multiple machines of the same type being available and production planning aspects. Using a genetic algorithm with a local optimizer (hybrid GA) to solve their model, their model finds the optimal assignment of machines to cells, the number of parts to be manufactured with each possible process plan and process routing and the sizes of each cell, and the number of parts of each type to be outsourced. The solution methodology involves partially solving the main model by applying mutation operators that determine the values for the decision variables related to the machine assignment to cells. This partially solved model is further modified by relaxing the machine capacity constraints. It is observed that the fully modified model has a min-cost flow structure that readily lends itself to a Dantzig-Wolfe decomposition approach.

Defersha and Chen (2006) presented a CM design model that incorporates many aspects of manufacturing in CM design. They take into consideration dynamic cell reconfiguration, lot splitting, operation sequence, multiple units of identical machines, machine capacity, workload balancing among cells, operation cost, outsourcing, setup cost, cell size limits and machine adjacency constraints. They have developed a non-linear mixed integer model and have followed some linearization steps in order to obtain a mixed-integer linear problem to solve various problem scenarios using Lingo. Defersha

(2006) also solved the proposed model using both parallel genetic algorithm and parallel simulated annealing.

Model A in this research deals with multi-period planning where the part demand and production requirements vary from one period to the next. The CM model is designed so as to achieve an agile cell configuration, that is, machine relocation can occur to change the overall CM layout and new part process routings can be chosen. At the same time, various features about the machines are integrated namely: machines can have identical copies, have limited capacities, are multifunctional, can be procured and can be relocated. Machine operating, maintenance and overhead costs, and overhead costs are minimized in the objective function. Other costs that are minimized include both intercellular and intracellular material handling, internal part production, subcontracting and inventory holding. Model A establishes the existence of alternate process routings for each part, where part routings can be chosen by selecting any of the machines that are able to carry out the required set of operations. The literature on CM design shows that there are various ways of defining part process routings. A majority of the models have the process routings defined in terms of a user-specified sequence of machines (or machine types), in an attempt to make the problem computationally tractable. Within these models, the production of a certain part type can be done through one or more process routings, where specific machines or machine types are manually specified to carry out the required sequence of operations in each of these part process routings. In the proposed model, part process routings are not manually configured but are automatically created. To that effect, given a required sequence of operations for a part type, the model takes into account all of the possible machines that can be used for each operation. Therefore, the proposed model does not limit itself to only one machine for each operation but is able to choose from any machine that has the required operation capability and capacity. This allows the model to be flexible when forming process routings for each part type. The use of such a flexible routing definition was only reported in Uddin and Shanker (2002). As such, in this model operation sequence of parts is an important aspect, as are the upper and lower limits on cell size. The manufacturing attributes covered by model **A** can be found in table 2-1.

Model **B** brings further extensions to model **A** as it considers some additional manufacturing aspects that can bring benefits in terms of system flexibility. Model **B** covers all of the important attributes of model **A** but also includes the formation of *contingency* routings for all parts. In model **B**, *main* and *contingency* routings are simultaneously formed for each and every part type using the approach suggested in Uddin and Shanker (2002). The formation of *contingency* part process routings can lead to a more reliable CM system that has more routing flexibility. This is achieved by forming *contingency* routings that can be activated whenever any *main* part routing gets disrupted (e.g. by machine breakdowns). There was no model reported to date that addresses the simultaneous formation of *main* and *contingency* routings for all parts. The manufacturing attributes covered by model **B** can be found in table 2-1.

2.4. Chapter Summary

A taxonomy of CM design approaches has been presented, revealing that a wide variety of solution methodologies have been researched and implemented with each one of them having its own strengths and flaws. Before implementing CM, a company needs to invest adequate effort during the planning and design phase. The benefits of CM are only possible by sufficiently incorporating the structural and operational features of a manufacturing plant within the design decisions. Because of this, there is a growing interest in the development of integrated models that can be used by designers to evaluate cellular layouts considering various aspects of the manufacturing operations. The literature review has revealed that mathematical programming is extensively used when developing CM models as it allows the integration of the cell formation problem with several important manufacturing attributes and other manufacturing problems. Owing to the computational difficulty of solving such a CM model, exact and meta-heuristic solution procedures can be developed and contrasted. As such, the rest of this thesis presents two integrated models, with an extensive coverage of the important identified manufacturing structural and operational features, along with some efficient solution approaches.

Chapter 3

Integrated Model with Production Planning and Dynamic System Reconfiguration

3.1. Introduction

This chapter presents the CM model **A.** Such a CM model is formulated based on the operation requirements of parts and the operational capability of machines. The operation sequence of parts is considered along with the presence of alternate part routings. Model **A** incorporates dynamic system reconfiguration (machine relocation and selection of new part process routings), machine procurement and production planning with the options of internal production of parts, inventory holding and subcontracting. The CM model is first formulated as a non-linear mixed integer program (MIP), referred to as model **F**. The high computational difficulty of solving the non-linear model **F** is demonstrated through some small-scale problem instances. Some linearization procedures are then proposed and implemented on model **F**, resulting in a linear mixed integer formulation of the model, referred to as **A**. Model **A** is solved through small, medium and large problem instances, using an exact solution procedure (ES). The computational results are then discussed.

3.2. The Proposed Mathematical Model

This section presents the problem definition for model **F**, the initial non-linear mixed integer mathematical program, along with objective function and constraints, the manufacturing system attributes incorporated and a discussion on the properties of the model.

3.2.1. Model Assumptions

When formulating the proposed CM model **A**, certain assumptions have been made and are as follows:

- Lot sizes are equal to one for all produced parts during material handling and inventory holding.
- No backlogging is allowed.
- Setup time and cost are assumed to be zero.
- Parts that are kept in inventory are fully processed. In other words, only fully completed parts are held as inventory.
- The demand in each period is deterministic, hence leading to dynamic deterministic production requirements.

3.2.2. Problem Definition

This integrated CM model **F** consists of the classical cell formation problem, bridged with the machine allocation, multi-period production planning and dynamic system reconfiguration problems, along with alternate part process routings. The classical cell formation problem (CFP), depicted in figure 3-1, is to group machines into cells, parts into part families and to assign part families to cells to form relatively independent cells. Within a manufacturing environment there are machine types which have different operational capabilities, limited capacities and multiple copies. There are also different part types, each of which requires a certain sequence of specific operations and processing capacity to complete production. The overall strategy of forming a CM layout is to group parts that require similar operations into the same cell whilst assigning machines to these cells so that they are capable of performing these operations.

	Part Family I		**Part Family II**				
Part Type	**1**	**3**	**2**	**4**	**5**	**Machine Type**	
	1	1				**2**	**Machine Cell I**
	1	1				**4**	
		X	1	1	1	**1**	**Machine Cell II**
			1	1	O	**3**	

Figure 3-1 : The classical cell formation problem

Figure 3-1 shows a representation of a solution to the classical CFP where four machines have been grouped into two cells, e.g. machines 2 and 4 are in cell I, and where five part types are grouped into two part families in such a way that each of these part families are assigned to a cell containing the machines required for the operations. Part type 1 and 3 (belonging to part family I) are thus assigned to machine cell I as they require machines 2 and 4 for processing, The '**X**', termed an exceptional element, represents intercellular movement of part type 3 between cells I and II. This happens because machine cell I does not contain machine type 1 required for an operation on part 3. So the latter has to be transferred to machine cell II as the latter contains a copy of machine type 1. Too many exceptional elements give rise to increased intercellular movement and relatively dependent cells, which can increase the coordination effort required between cells. It must be noted that this is a very simple situation as machine capacities, availabilities, multiple routings and other manufacturing aspects are ignored. The '**O**' represents a void and this occurs because machine type 3 is only required to process parts 2 and 4 in cell II and is not needed for part type 5. Too many voids can lead to the formation of large inefficient cells and can give rise to scheduling problems. A *process routing* is defined as a sequence of specific machines or work centers that a part type has to go through in order to complete all of its required operations. This implies that a part type can have alternate *process routings* when machines have multiple capacities, capabilities and availabilities. Therefore, there may be numerous feasible cellular configurations because of these alternate part *process routings*. The optimal configuration for this classical cell formation problem is the one that minimizes the total cost of material handling in terms of intercellular (from cell to cell) and intracellular (within the same cell). With regards to the multi-period production planning problem, it is assumed that the demands for parts vary in a deterministic way. This allows the model to consider producing more in a period so that inventory can be used in future periods or to subcontract parts when internal production is not feasible either due to insufficient machine capacity or uneconomical repercussions. Simultaneously, the CM environment can respond by undergoing system reconfiguration where machines are relocated from one cell to another and/or where new process routings are selected for the part types. Additionally, machines can be procured to increase the internal production capacity. The notations used for the model are presented followed by the objective function, constraints and model properties.

Sets:

$\mathcal{P}$	$= \{1, 2, 3, \ldots, P\}$	Index set of part types
$\mathcal{K}(p)$	$= \{1, 2, 3, \ldots, K_p\}$	Index set of operations indices for part type p
$\mathcal{M}$	$= \{1, 2, 3, \ldots, M\}$	Index set of machine types
$\mathcal{Q}(k,p)$	$= \{1,2,3,\ldots,Q_{kp}\}$	Index set of machine types that can perform operation k for part type p
$\mathcal{C}$	$= \{1,2,3,\ldots,C\}$	Index set of cells
$\mathcal{T}$	$= \{1,2,3,\ldots,T\}$	Index set of time periods

Model Parameters:

δ_m	Relocation cost per machine type m per period;	$\forall m \in \mathcal{M}$
α_m	Maintenance and overhead costs per machine type m	$\forall m \in \mathcal{M}$
μ_m	Procurement cost per machine type m	$\forall m \in \mathcal{M}$

ε_m	Operating cost per unit time per machine type m	$\forall m \in \mathcal{M}$
o_p	Outsourcing cost per part type p	$\forall p \in \mathcal{P}$
H_p	Inventory holding cost per part type p per time period	$\forall p \in \mathcal{P}$
β_p	Internal production cost per part type p	$\forall p \in \mathcal{P}$
IE_p	Intercellular material handling cost per part type p	$\forall p \in \mathcal{P}$
IA_p	Intracellular material handling cost per part type p	$\forall p \in \mathcal{P}$
$D_p(t)$	Demand for part type p at time period t	$\forall p \in \mathcal{P}, \forall t \in \mathcal{T}$
$A_m(t)$	Quantity of machine type m available at time period t	$\forall m \in \mathcal{M}, \forall t \in \mathcal{T}$
T_m	Capacity of one unit of machine type m during one period	$\forall m \in \mathcal{M}$
B_U	Upper Cell Size Limit	
B_L	Lower Cell Size Limit	
e_{kpm}	Processing time of operation k on machine m per part p $\forall k \in \mathcal{K}(p),\ \forall p \in \mathcal{P},\ \forall m \in Q(k,p)$	

Model Decision Variables

$X_{kpmc}(t)$	Number of parts of type p processed by operation k on machine type m in cell c at time t $\forall k \in \mathcal{K}(p),\ \forall p \in \mathcal{P},\ \forall m \in Q(k,p),\ \forall c \in C,\ \forall t \in \mathcal{T}$
$O_p(t)$	Number of parts p to be outsourced at time t $\forall p \in \mathcal{P}, \forall t \in \mathcal{T}$

$V_p(t)$ Quantity of inventory of part type p kept in period t and carried over to period $t+1$

$\forall p \in \mathcal{P},\ \forall t \in \mathcal{T}$

$N_{mc}(t)$ Number of machines of type m present in cell c at time t

$\forall m \in \mathcal{M}, \forall c \in C,\ \forall t \in \mathcal{T}$

$N^{+}_{mc}(t)$ Number of machines of type m added to cell c at time t

$\forall m \in \mathcal{M}, \forall c \in C,\ \forall t \in \mathcal{T}$

$N^{-}_{mc}(t)$ Number of machines of type m removed from cell c at time t

$\forall m \in \mathcal{M}, \forall c \in C,\ \forall t \in \mathcal{T}$

$BN_m(t)$ Number of machines of type m procured at time t

$\forall m \in \mathcal{M},\ \forall t \in \mathcal{T}$

$A^{*}_{m}(t)$ Quantity of machine type m available at time period t after accounting for machines that have been procured

$\forall m \in \mathcal{M},\ \forall t \in \mathcal{T}$

$Z_{kpmc}(t)$ 1 if operation k for part type p is carried out on machine type m in cell c at time t, and 0 otherwise

$\forall k \in \mathcal{K}(p),\ \forall p \in \mathcal{P},\ \forall m \in Q(k,p),\ \forall c \in C,\ \forall t \in \mathcal{T}$

3.2.3. Objective Function and Constraints

The developed CM model is first formulated as a non-linear mixed integer program and is referred to as model **F**:

$$\text{Minimize} \quad \sum_{t\in\mathcal{T}} \sum_{c\in C} \sum_{m\in\mathcal{M}} \delta_m \cdot \left(N^{+}_{mc}(t) + N^{-}_{mc}(t)\right) \tag{1.1}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{c\in C} \sum_{m\in\mathcal{M}} \alpha_m \cdot N_{mc}(t) \tag{1.2}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}(p)} \sum_{m\in\mathcal{Q}(k,p)} \sum_{c\in C} \varepsilon_m \, e_{kpm} \cdot X_{kpmc}(t) \tag{1. 3}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} o_p \cdot O_p(t) \tag{1.4}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} H_p \cdot V_p(t) \tag{1.5}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}(p)} \sum_{m\in\mathcal{Q}(k,p)} \sum_{c\in C} \beta_p \cdot X_{kpmc}(t) \tag{1.6}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}(p)\setminus\{K_p\}} \sum_{c\in C} \left| IE_p \cdot \left(\sum_{m\in\mathcal{Q}(k+1,p)} X_{k+1,pmc}(t) - \sum_{m\in\mathcal{Q}(k,p)} X_{kpmc}(t) \right) \right| \tag{1.7}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}(p)\setminus\{K_p\}} \sum_{m\in\mathcal{Q}(k,p)} \sum_{c\in C} \left(IA_p \cdot Z_{k+1,pmc}(t) \cdot \sum_{n\in\mathcal{Q}(k,p)\setminus\{n=m\}} \left(Z_{kpnc}(t) \cdot X_{kpnc}(t)\right) \right) \tag{1.8}$$

$$+ \quad \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \mu_m \cdot BN_m(t) \tag{1.9}$$

The objective function consists of nine cost components. Term (1.1) is the machine relocation (reconfiguration) cost, where the latter is incurred when machines are added and/or removed from cells. Term (1.2) denotes the machine maintenance and overhead costs. Term (1.3) represents the machine operating cost. Term (1.4) represents the outsourcing cost. Term (1.5) corresponds to the inventory holding cost. Term (1.6) is the internal part production cost. Term (1.7) denotes the intercellular material handling cost. This cost is incurred whenever consecutive operations of the same part type are carried out in different cells. For instance, assume that operation k of part type p is done on machine m in cell c at time t. If the next operation, k+1, of part p is done *on any machine but in another cell*, then there is *intercellular cost.* The latter is directly proportional to the number of parts moved between these two cells. In this model, it is assumed that the distances between each cell are equal so that the unit intercellular is expressed only as a function of the part type being handled. For instance, larger part types are more expensive to handle but the cost of moving equal quantities of this part type from, say cell 1 to cell 2, is the same as from cell 4 to cell 5. Term (1.8) is the intracellular material handling cost. This cost is incurred whenever consecutive operations of the same part type are done in the same cell but on different machines. For instance, say that operation k of part type p is done on machine m in cell c at time t. If the next operation, k+1, of part p is done *on any other machine but within the same cell*, then there is *intracellular cost.* The latter is directly proportional to the number of parts moved between the different machines in the same cell. Again, it is assumed that the unit intra-cellular cost only depends on the part type being handled. Within a cell, the distances between any two machines are considered to be the same. So, larger part types are more expensive to handle but would fetch the same intra-cellular cost were they moved from machine type 1 to 2 or from machine type 4 to 5 within the same cell. Term (1.9) is the cost incurred when machines are procured. The constraints of the problem, along with relevant explanations, are given below.

$$X_{kpmc}(t) \leq M \cdot Z_{kpmc}(t) \qquad \forall k \in \mathcal{K}(p),\ \forall p \in \mathcal{P},\ \forall m \in \mathcal{Q}(k,p),\ \forall c \in \mathcal{C},\ \forall t \in \mathcal{T} \qquad (2)$$

In model **F**, constraint (2) shows that the number of parts produced internally can be positive only if $Z_{kpmc}(t)=1$, that is, it has been decided that part p would be produced internally by operation k on machine m in cell c

$$\sum_{m\in Q(k,p)} \sum_{c\in C} X_{kpmc}(t) + V_p(t-1) - V_p(t) + O_p(t) = D_p(t)$$

$$\forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall t \in \mathcal{T} \qquad (3)$$

Constraint (3) shows that demand is satisfied in each period t through internal production, and/or outsourcing, and/or inventory carried over from the previous period.

$$A^*_m(t=1) = A_m(t=1) + BN_m(t=1) \qquad \forall m \in \mathcal{M} \qquad (4)$$

Constraint (4) relates to the machine availability constraint for period 1, taking into consideration the extra machines introduced through the machine procurement option. In period 1, the total number of machine of each type available is equal to the machine availability (before procurement) plus the number of machines procured in the same period 1. Therefore, if $A_m(t=1)=0$, there are no machine present in the system initially, meaning that a CM system is being designed and implemented from no existing manufacturing layout. If $A_m(t=1)>0$, there are machines already available in the system, meaning that the existing manufacturing layout is being reconfigured to form a CM layout.

$$A^*_m(t+1) = A^*_m(t) + BN_m(t+1) \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (5)$$

Constraint (5) relates to the machine availability constraint for the subsequent time periods. It takes into consideration the extra machines introduced through the machine procurement option in the period under consideration as well as those procured in all of the previous periods. In each period, the total number of machines of each type available is equal to the machine availability (with procured machines already accounted for) in the previous period added to the number of machines procured in the period under consideration.

$$\sum_{c \in C} N_{mc}(t) \leq A_m^*(t) \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (6)$$

Constraint (6) also relates to the machine availability constraint, where the total number of machines of each type assigned to all cells is less than or equal to the machine availability in the same period (where, through constraints (4) and (5), the machine availability has been updated in each period considering the total number of machines procured).

$$\sum_{m \in \mathcal{M}} N_{mc}(t) \leq B_U \qquad \forall c \in C, \forall t \in \mathcal{T} \qquad (7)$$

$$\sum_{m \in \mathcal{M}} N_{mc}(t) \geq B_L \qquad \forall c \in C, \forall t \in \mathcal{T} \qquad (8)$$

The cell size is user-defined through constraints (7) and (8), where the cell size lies within lower and upper bounds.

$$N_{mc}(t) = N_{mc}(t-1) + N_{mc}^{+}(t) - N_{mc}^{-}(t) \qquad \forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T} \qquad (9)$$

Constraint (9) accounts for machine relocation, where the number of machines of a type assigned to a cell during a period is equal to the number of machines in the previous period, plus the number of machines added and minus the number of machines removed.

$$\sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}(p)} e_{kpm} \cdot X_{kpmc}(t) \leq T_m \cdot N_{mc}(t) \qquad \forall m \in Q(k,p),\ \forall c \in C,\ \forall t \in \mathcal{T} \qquad (10)$$

Constraint (10) shows the machine capacity constraint. It ensures that the required internal production capacities respect the available machine capacities, taking into account the required production capacities according to the quantity of all parts processed and the available production capacities according to the total number of machines assigned to all cells.

$$\sum_{c \in C} \sum_{m \in Q(k+1,p)} X_{k+1,pmc}(t) = \sum_{c \in C} \sum_{m \in Q(k,p)} X_{kpmc}(t) \qquad \forall k \in \mathcal{K}(p) / \{K_p\},\ \forall p \in \mathcal{P},\ \forall t \in \mathcal{T} \qquad (11)$$

Constraint (11) ensures material flow conservation of the parts under production. It ensures that the total quantity of parts processed in an operation is equal to the total quantity of parts processed in the next and/or preceding operation. With such a constraint, each operation of a certain part type can be split in different cells and on different machines that have the required capabilities. Since a trade-off is being sought in the objective function with regards to the amount of intercellular and intracellular movement (minimization of intercellular and intracellular material handling cost), constraint (11) allows for more possible part routing allocations to be explored.

$X_{kpmc}(t) \geq 0$ and integer	$\forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p),$	
	$\forall c \in C, \forall t \in \mathcal{T}$	(12)
$O_p(t) \geq 0$ and integer	$\forall p \in \mathcal{P}, \forall t \in \mathcal{T}$	(13)
$V_p(t) \geq 0$ and integer	$\forall p \in \mathcal{P}, \forall t \in \mathcal{T} \setminus \{T\}$	(14)
$N_{mc}(t) \geq 0$ and integer	$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$	(15)
$N^{+}_{mc}(t) \geq 0$ and integer	$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$	(16)
$N^{-}_{mc}(t) \geq 0$ and integer	$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T}$	(17)
$BN_m(t) \geq 0$ and integer	$\forall m \in \mathcal{M}, \forall t \in \mathcal{T}$	(18)
$A^{*}_{m}(t) \geq 0$	$\forall m \in \mathcal{M}, \forall t \in \mathcal{T}$	(19)
$Z_{kpmc}(t) \in \{0,1\}$	$\forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p),$	
	$\forall c \in C, \forall t \in \mathcal{T}$	(20)

Constraints (12) to (20) are the logical binary and non-negativity integer requirements on the decision variables.

3.3. Properties of the Model

This section draws attention to and explains the important manufacturing design aspects that have been integrated within the proposed CM model. Such a model takes a holistic approach in CM design considering more manufacturing aspects than most other models as it introduces many other extensions to a number of well-integrated models reported in the literature (Safaei et al 2007; Defersha and Chen, 2006; Mungwattana, 2000). These extensions can be classified as production planning with internal production, inventory holding and part subcontracting, dynamic system reconfiguration, cell formation, machine procurement and the presence of alternate process routings with the consideration of part operation sequence.

Machine procurement. One of the key strengths of this CM model **A** resides is that the system is able to respond to variations in part mix demands by the fact that new machines can be brought in, at a cost, through machine procurement to increase the internal production capacity.

Production Planning. Various benefits and more flexibility in the production planning activity can be achieved for a multiple period planning horizon as the proposed model enables the CM system to set optimal levels of internal production, inventory holding and outsourcing to meet the part demand in each period. Depending on the demand and total cost of meeting that demand, the system could produce some surplus parts in a time period which can be carried over as inventory and used to supply part or all of the demand for the parts in future planning periods. It is assumed that there is no initial inventory and that no inventory is left over at the end of the whole planning horizon. Outsourcing is an option where some of the required parts are procured when internal production cannot satisfy the part demand, either due to insufficient machine capacity or adverse economical repercussions.

Dynamic System Reconfiguration. As a result of dynamic deterministic demands within the planning horizon, a CM configuration for a period might not be optimal or even feasible for other planning periods. Therefore, the model allows the user to obtain the best configuration within each planning period in terms of the types and number of machines assigned to cells, part types assigned to cells and part routings. This is achieved when machines are added and/or removed (relocation) and when new process routings for parts are chosen from one period to the next. The presence of such a dynamic system reconfiguration feature enhances the flexibility of the CM system to respond to variations in part mix demands and machine availabilities.

Cell formation. The cell formation in the proposed model occurs in such a way that any part operation can be split onto one or more machines within the same cell or even in different cells. This approach to cell formation allows a trade-off to be made between intercellular and intracellular material handling since the costs associated with those activities are minimized in the model objective function. This allows for a control on material handling and can also enable better part process routings to be found by the model.

Alternate process routings with operation sequence The availability of multifunctional machines and the presence of multiple copies of each machine type are taken into account when forming machine cells, assigning parts to cells and selecting part process routings. A p*rocess routing* is defined as the set of machines and cells that are visited by a part type when it undergoes each one of its required operations. Multiple possible routings are possible for each part type because of these multi-functional machines and the multiple copies of each of these. This creates more flexibility in forming CM configurations. In most literature, routes are pre-specified taking into account the operations required by a part and the machine capabilities. In the same literature and the relevant models, the approaches taken to form part routings are as follows. In certain models, the presence of single process routings is assumed, where process routings are structured in terms of a selected (and user-specified) sequence of either specific machines or machine types. In more advanced models, multiple part process routings can be formed by specifying additional sequences of machines or machine types that can process the part type under consideration. With such approaches, only a discrete number of process routings are explored. In this research, model **F** is designed in such a way that the system optimally decides on the best part routing(s) instead of the user having to specify pre-determined routes. This is further explained as follows. The set of operations $\mathcal{K}(p)$ required for part type p to complete production is shown below, followed by the set of machine types, $Q(k,p)$, that can perform operation k for part type p.

$\mathcal{K}(p) = \{K1, K2, K3\}$	set of operations for part type p
$Q(K1,p) = \{M1\}$	set of machine types that can perform operation K1 for part type p
$Q(K2,p) = \{M2, M3\}$	set of machine types that can perform operation K2 for part type p
$Q(K3,p) = \{M3, M4\}$	set of machine types that can perform operation K3 for part type p

The alternate process routings for part type p can be chosen from the following four possible machine sequences: M1→M2→M3, and/or M1→M2→M4, and/or

M1→M3→M3, and/or M1→M3→M4. The definition of part process routings used in the proposed model brings more flexibility and variety into the system as all of the machines are taken into account during the formation of each part process routing. Throughout the proposed model the system automatically decides on the process routings for each part type instead of the user having to specify a number of pre-determined routings. *Operation sequence* is also incorporated because it allows the correct calculation of intercellular and intracellular material handling cost. In comparison, most models define the part process routings in terms of one or more (but not all) of these machine sequences, thereby limiting themselves to only a subset of all the possible process routings that can be formed. In a majority of previous models considering multiple part process routings, the possible process routings for part type *p* are defined in terms of at most two possible machine sequences, for instance M1→M2→M3 and M1→M2→M4 only, disregarding the two other possibilities M1→M3→M3 and M1→M3→M4.

It is widely reported in CM literature that the whole problem of designing a CM system, taking into account the numerous phases and criteria involved, belongs to the class of NP-complete problems (King and Nakornchai, 1982; Ballakur, 1985). The proposed model is expected to be very difficult to solve since it includes the machine-part grouping (cell formation) problem extended with other manufacturing problems pertaining to multi-period production planning and dynamic system reconfiguration. The other various manufacturing attributes considered in the proposed model make the problem especially more complex to solve, in addition to it already being a combinatorial complex problem. For instance, the allocation of parts to process routings (whilst respecting machine capacities and considering duplicates of machine types) is more challenging when alternate process routings are considered than when there are only single process routings. One phase of the proposed model is to determine the number and types of machines to assign for each part type; then to determine the process routings from multiple possible ones (machines can perform more than one operation and are available as multiple copies); Logendran *et al.* (1994) have shown that the problem involved in this phase is NP-hard. The consideration of cellular reconfiguration for a planning horizon is in itself a problem that belongs to the class of NP-complete problems. Chen (1998) has developed a model that considers system reconfiguration in terms of machine relocation, showing that the model is NP-hard. The proposed model **F** is, therefore, expected to be very difficult to solve since it integrates the problem of cell formation, dynamic system reconfiguration, and production planning along with the consideration of alternate process routings, machine capacities and availabilities.

3.4. Preliminary Computational Experience

Four small-scale problems were solved in order to validate the proposed mathematical formulation that represents the integrated CM model of this research. The problems were solved using an off-the-shelf optimization software, namely Extended Lingo (LINDO Systems Inc, 2005), with the results shown in table 3-1.

Table 3-1 : Results of the implementation of model F to four small scale scenarios in terms of solution time and solution status

No of part types P	No of m/c types M	No of time periods T	No of cells C	No of operations per part K_p	Time taken to reach solution (s)	Solution Status
1	3	2	2	2	4	Local optimum
1	3	5	2	2	1,290	Local optimum
2	3	5	2	2-3	1,955	Local optimum
3	3	5	3	2-3	N/A	N/A

For the first three scenarios, local optimum solutions were obtained after the shown computational times. No solution was obtained for scenario four (three part types), even after fifty hours of computation. The results shown in table 3-1 demonstrate that it is not possible to guarantee getting a global optimum solution when solving model **F** with Extended Lingo. Furthermore, as the size of the problems grows, it will be more arduous to solve this model. Therefore, in the next section, some linearization steps are proposed to transform this mixed integer non-linear model **F** into a mixed integer linear programming formulation **A**.

3.5. Linearization of the Model

In this section, the linearization procedure of model **F** is presented through two propositions followed by the respective proofs. The resulting linear mixed integer linear programming formulation **A** for the CM model is then presented.

3.5.1. Linearization Procedure

The linearization procedure that is proposed here consists of two steps that are given by the two propositions stated below. The non-linearity results from terms (1.7) and (1.8) in the objective function of model **F**, therefore, these two terms will be linearized using the following auxiliary continuous variables $YP_{kpmc}(t)$, $YM_{kpmc}(t)$, $Z_{kpmnc}(t)$ and $W_{kpmnc}(t)$. Each proposition for linearization is followed by a proof and a numerical example that illustrates the meaning of each auxiliary (linearization) variable and the expressions where they are used.

Proposition 1. The non-linear component of the objective function (1.7) in problem **F** can be linearized by the following transformation $\left|X_{k+1,pmc}(t)-X_{kpmc}(t)\right|=YP_{kpmc}(t)+YM_{kpmc}(t)$, under the following set of constraints:

$$X_{k+1,pmc}(t)-X_{kpmc}(t)=YP_{kpmc}(t)-YM_{kpmc}(t) \qquad \forall k\in\mathcal{K}(p)/\{K_p\}, \forall p\in\mathcal{P}, \tag{21}$$

$$\forall m\in\mathcal{Q}(k,p), \forall c\in\mathbf{C}, \forall t\in\mathcal{T}$$

Proof. Consider the following three cases:

i. $X_{k+1,pmc}(t) > X_{kpmc}(t)$. By (21), $YP_{kpmc}(t) - YM_{kpmc}(t) > 0$. Since this is a minimization problem and the objective function cost coefficients are strictly positive, $YM_{kpmc}(t) = 0$ and $YP_{kpmc}(t) = X_{k+1,pmc}(t) - X_{kpmc}(t)$ will hold in the optimal solution.

ii. $X_{k+1,pmc}(t) < X_{kpmc}(t)$. By (21), $YP_{kpmc}(t) - YM_{kpmc}(t) < 0$. In this case, again with the coefficients of $YP_{kpmc}(t)$ and $YM_{kpmc}(t)$ being strictly positive, the objective function will ensure that $YP_{kpmc}(t) = 0$ and thus $YM_{kpmc}(t) = X_{kpmc}(t) - X_{k+1,pmc}(t)$ will hold in the optimal solution.

iii. $X_{k+1,pmc}(t) = X_{kpmc}(t)$. By (21), $YP_{kpmc}(t) - YM_{kpmc}(t) = 0$. In this case, both $YP_{kpmc}(t) = 0$ and $YM_{kpmc}(t) = 0$, will hold in the optimal solution since their coefficients in the objective function are strictly positive. ■

$YP_{kpmc}(t)$ and $YM_{kpmc}(t)$ are both non-negative continuous variables that are used to linearize an expression of the type given in term (1.7), one where the absolute term creates non-linearity. When considered together in the expression $YP_{kpmc}(t) + YM_{kpmc}(t)$, under constraint (21), they give the magnitude of the quantity of parts involved in intercellular movement. This indicates the quantity of parts moved from operation k (performed on machine type m in cell c at time t) to another cell where the next operation $k+1$ in the process routing takes place (performed on any machine type m capable of doing operation $k+1$).

The linearization in proposition 1 is explained in more detail through the two numerical examples shown in table 3-2. $X_{kpmc}(t)$ and $X_{k+1,pmc}(t)$ can take any positive general integer value. The non-negative variables ($YP_{kpmc}(t)$ and $YM_{kpmc}(t)$) take values in such a way that the computations for intercellular cost using the linear terms generate the same final results as would be obtained by the original variables and non-linear terms.

Table 3-2 : Examples to the linearization stated in proposition 1

	Example 1	Example 2
$X_{kpmc}(t)$	5	10
$X_{k+1,pmc}(t)$	10	5
$X_{k+1,pmc}(t)-X_{kpmc}(t)$	5	-5
$\lvert X_{k+1,pmc}(t)-X_{kpmc}(t)\rvert$	5	5
$YP_{kpmc}(t)+YM_{kpmc}(t)$	5	5
$YP_{kpmc}(t)-YM_{kpmc}(t)$	5	-5
$YP_{kpmc}(t)$	5	0
$YM_{kpmc}(t)$	0	5

In table 3-2, consider example 1 where $X_{kpmc}(t)=5$ and $X_{k+1,pmc}(t)=10$. This means that 5 units of part *p* were processed by operation *k* on machine type *m* in cell *c* at time *t*. The next operation for *p*, that is *k+1*, involves the processing of 10 units on the same machine *m* in the same cell *c* at time *t*. This implies that 5 additional parts *p* have been transferred from another cell (where they underwent operation *k*) to destination cell *c* for the current operation *k+1*. The term $\lvert X_{k+1,pmc}(t)-X_{kpmc}(t)\rvert$ gives the magnitude of the quantity of parts transferred from different cells between two successive operations of the same part p in the same period t. Thus $YP_{kpmc}(t)$ and $YM_{kpmc}(t)$ take non-negative values in such a way that $YP_{kpmc}(t)+YM_{kpmc}(t)$ also gives the correct magnitude for the same intercellular material flow; in this case 5 units of parts p. Upon considering example 2 in table 3-2, it is seen that $X_{kpmc}(t)=10$ and $X_{k+1,pmc}(t)=5$. Out of the 10 units of part p processed by operation k on machine type m in cell c, only 5 undergo the next operation $k+1$ within the same cell. Thus, the other 5 units of part p have been transferred to another cell to undergo the next operation $k+1$. $\left|X_{k+1,pmc}(t)-X_{kpmc}(t)\right|=5$ gives the correct magnitude for the intercellular flow involved, as does $YP_{kpmc}(t)+YM_{kpmc}(t)=5$, where both $YP_{kpmc}(t)$ and $YM_{kpmc}(t)$ have taken the required non-negative values.

Proposition 2. The non-linear intracellular cost terms in the objective function (1.8) of problem **F** can be linearized with $W_{kpmnc}(t)=Z_{kpmnc}(t)\cdot X_{kpnc}(t)$, where $Z_{kpmnc}(t)=Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)$, under the following sets of constraints:

$$Z_{kpmnc}(t)\geq Z_{k+1,pmc}(t)+Z_{kpnc}(t)-1 \qquad \forall k\in\mathcal{K}(p)/\{K_p\},\ \forall p\in\mathcal{P},\ \forall m\in Q(k,p),\ \forall n\in Q(k,p)/\{n=m\},\ \forall c\in C,\ \forall t\in\mathcal{T} \tag{22}$$

$$W_{kpmnc}(t)\leq M\cdot Z_{kpmnc}(t) \qquad \forall k\in\mathcal{K}(p),\ \forall p\in\mathcal{P},\ \forall m\in Q(k,p),\ \forall n\in Q(k,p)/\{n=m\},\ \forall c\in C,\ \forall t\in\mathcal{T} \tag{23}$$

where M is a large positive value.

Proof. This can be shown for each of the two possible cases that can arise.

i. $Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)=0$. Such a situation arises under one of the following three sub-cases:

a. $Z_{k+1,pmc}(t)=1$ and $Z_{kpnc}(t)=0$. $\forall k \in \mathcal{K}(p)/\{K_p\}$, $\forall p \in \mathcal{P}$, $\forall m \in Q(k,p)$, $\forall n \in Q(k,p)/\{n=m\}$, $\forall c \in C$, $\forall t \in \mathcal{T}$

b. $Z_{k+1,pmc}(t)=0$ and $Z_{kpnc}(t)=1$. $\forall k \in \mathcal{K}(p)/\{K_p\}$, $\forall p \in \mathcal{P}$, $\forall m \in Q(k,p)$, $\forall n \in Q(k,p)/\{n=m\}$, $\forall c \in C$, $\forall t \in \mathcal{T}$

c. $Z_{k+1,pmc}(t)=0$ and $Z_{kpnc}(t)=0$. $\forall k \in \mathcal{K}(p)/\{K_p\}$, $\forall p \in \mathcal{P}$, $\forall m \in Q(k,p)$, $\forall n \in Q(k,p)/\{n=m\}$, $\forall c \in C$, $\forall t \in \mathcal{T}$

In all of the three sub-cases given above, the value of the non-linear term (1.8) is given by $Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)\cdot X_{kpnc}(t)=0$, under any value of $X_{kpnc}(t)$. In this case, constraint (22) implies $Z_{kpmnc}(t)\geq -1$ and since $Z_{kpmnc}(t)$ has a strictly positive cost coefficient, the minimizing objective function ensures that $Z_{kpmnc}(t)=0$. Thus, under constraint (23), $W_{kpmnc}(t)=0$. Hence, the value of the non-linear term (1.8) and $W_{kpmnc}(t)$ have equivalent values for this case.

ii. $Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)=1$. $\forall k \in \mathcal{K}(p)/\{K_p\}$, $\forall p \in \mathcal{P}$, $\forall m \in Q(k,p)$, $\forall n \in Q(k,p)/\{n=m\}$, $\forall c \in C$, $\forall t \in \mathcal{T}$

Such a situation arises when $Z_{k+1,pmc}(t) = Z_{kpnc}(t) = 1$ so that the non-linear term (1.8) can take any non-negative value since $Z_{k+1,pmc}(t) \cdot Z_{kpnc}(t) \cdot X_{kpnc}(t) = X_{kpnc}(t)$. In this case, constraint (22) implies $Z_{kpmnc}(t) \geq 1$ and since $Z_{kpmnc}(t)$ has a strictly positive cost coefficient, the minimizing objective function ensures that $Z_{kpmnc}(t) = 1$. Then under constraint (23), it follows that $W_{kpmnc}(t) \leq M$, showing that $W_{kpmnc}(t)$ can also take any non-negative value. Therefore, the non-linear term (1.8) and $W_{kpmnc}(t)$ are equivalent for this case. ■

$Z_{kpmnc}(t)$ and $W_{kpmnc}(t)$ are non-negative continuous variables used to linearize the term (1.8), where the non-linearity is caused by the multiplication of binary variables $Z_{kpnc}(t)$ and $Z_{k+1,pmc}(t)$ as well as integer variables $X_{kpnc}(t)$. After linearization, the linearization variable $Z_{kpmnc}(t)$ shows whether there is intracellular movement. When operation k of part p at time t is done on machine type n in cell c and this is followed by the next operation, $k+1$, being done on any other machine type within the same cell, there is intracellular movement involved so that $Z_{kpmnc}(t) > 0$. Otherwise, when $Z_{kpmnc}(t) = 0$, there is no intracellular movement as the consecutive operations are done in different cells. $W_{kpmnc}(t)$ gives the magnitude of the number of parts involved in the intracellular movement.

Proposition 2 is illustrated through the four numerical examples shown in table 3-3, where the possible values for $Z_{kpnc}(t)$, $Z_{k+1,pmc}(t)$ and $X_{kpnc}(t)$ are analyzed. $W_{kpmnc}(t)$ can, therefore, be calculated and compared with the non-linear term (1.8).

Table 3-3 : Examples to the linearization stated in proposition 2

	$Z_{kpnc}(t)$	$Z_{k+1,pmc}(t)$	$Z_{kpmnc}(t)$	$X_{kpnc}(t)$	$W_{kpmnc}(t)$	$Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)\cdot X_{kpnc}(t)$
Example 1	1	1	1	20	20	20
Example 2	1	0	0	Any value	0	0
Example 3	0	1	0	Any value	0	0
Example 4	0	0	0	Any value	0	0

Example 1 of table 3-3 deals with two consecutive operations on part *p*. $Z_{kpnc}(t)=1$ shows that operation *k* is performed on machine type *n* in cell *c* whilst $Z_{k+1,pmc}(t)=1$ shows that the next operation (for the same part type *p*) is done within the same cell *c* on another machine type *m*. Therefore, $Z_{kpmnc}(t)=1$ implies that there is intracellular movement, that is transfer of parts between different machines in the same cell *c*. $W_{kpmnc}(t)=20$ shows that 20 units of part *p* are involved in the intracellular transfer between machines *m* and *n* in cell *c*, when the operation sequence is *k* and *k+1* for part type *p* in time period *t*. Compared with $Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t).X_{kpnc}(t)=20$, which is the number of parts involved in the intracellular transfer, it is seen that the linear term, $W_{kpmnc}(t)$, gives the same correct value of 20. In examples 2, 3 and 4, $Z_{kpmnc}(t)=0$ indicates that the consecutive operations k and $k+1$ are performed in different cells, so that there is no intracellular movement. Therefore, $W_{kpmnc}(t)=0$, under any value of $X_{kpnc}(t)$. It is observed that $Z_{k+1,pmc}(t)\cdot Z_{kpnc}(t)\cdot X_{kpnc}(t)=0$, which also indicates that there is no intracellular material flow, showing that the non - linear term (1.8) and $W_{kpmnc}(t)$ are equivalent.

3.5.2. The Linearized Mixed Integer Programming Formulation

The transformed mixed integer linear programming formulation is referred to as model **A** and is presented below.

Minimize (1.1) – (1.6), (1.9)

$$+ \quad \sum_{t\in\mathcal{T}}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}(p)}\sum_{c\in C}\sum_{m\in Q(k,p)} IE_p\cdot\left(YP_{kpmc}(t)+YM_{kpmc}(t)\right) \tag{1.10}$$

$$+ \quad \sum_{t\in\mathcal{T}}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}(p)}\sum_{m\in Q(k,p)}\sum_{c\in C}\sum_{n\in Q(k,p)\setminus\{n=m\}} IA_p\cdot W_{kpmnc}(t) \tag{1.11}$$

subject to (2) – (20)

$$X_{k+1,pmc}(t)-X_{kpmc}(t)=YP_{kpmc}(t)-YM_{kpmc}(t) \qquad \forall k\in\mathcal{K}(p)/\{K_p\}, \forall p\in\mathcal{P}, \forall m\in Q(k,p), \forall c\in C, \forall t\in\mathcal{T} \tag{21}$$

$$Z_{kpmnc}(t)\geq Z_{k+1,pmc}(t)+Z_{kpnc}(t)-1 \qquad \forall k\in\mathcal{K}(p)/\{K_p\}, \forall p\in\mathcal{P}, \forall m\in Q(k,p), \forall n\in Q(k,p)/\{n=m\}, \forall c\in C, \forall t\in\mathcal{T} \tag{22}$$

$$W_{kpmnc}(t) \leq M \cdot Z_{kpmnc}(t) \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p), \forall n \in Q(k,p)/\{n=m\}, \forall c \in C, \forall t \in \mathcal{T} \tag{23}$$

$$YP_{kpmc}(t),\ YM_{kpmc}(t) \geq 0 \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p), \forall c \in C, \forall t \in \mathcal{T} \tag{24}$$

$$Z_{kpmnc}(t) \geq 0 \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p), \forall n \in Q(k,p)/\{n=m\}, \forall c \in C, \forall t \in \mathcal{T} \tag{25}$$

$$W_{kpmnc}(t) \geq 0 \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p), \forall n \in Q(k,p)/\{n=m\}, \forall c \in C, \forall t \in \mathcal{T} \tag{26}$$

3.6. Exact Solution (ES) Procedure and Numerical Examples

This section presents the use of 15 numerical examples extracted from existing CM literature to solve the proposed CM model **A**. All the computational experiments were performed on a 3 GHz Pentium 4 workstation running Linux. The models were solved using CPLEX 9.0.1 (ILOG Inc., 2006). The exact solution procedure (ES) used is the simplex-based branch-and-cut algorithm of CPLEX 9.0.1. The small-scale problems used in section (3.4) for the preliminary computational experience on the non-linear model **F** are now tested using the linear model **A**. In addition, data sets of larger sizes are used from Mungwattana (2000), Cao and Chen (2004), Chen and Cao (2004), Nsakanda (2006), Defersha and Chen (2006). Since the models developed and used by these authors are different from the proposed model **A**, some additional cost parameters were added pertaining to the features not addressed in their data sets and model (e.g. inventory holding cost). The unknown cost parameters, which proved more difficult to get, were extracted by cross-referencing between the data sets containing them and then incorporated within the other data sets that are missing this information. Therefore, all of the data sets used in each solved numerical example contain values within the same range in terms of unit costs. Such data can be found in table 3-4.

Table 3-4 : Cellular manufacturing data sets

Parameter	Parameter setting/range (units)
Demand for part type p at time period t	30,000-31,000
Processing time of operation k on machine m per part p	0.2
Internal production cost per part type p	0.5
Inventory holding cost per part type p per time period	5-10
Outsourcing cost per part type p	250-1000
Intercellular material handling cost per part type p	5
Intracellular material handling cost per part type p	2
Quantity of machine type m available at time period t	2-5
Capacity of one unit of machine type m during one period	10,000-30,000
Operating cost per unit time per machine type m	10
Maintenance and overhead costs per machine type m	20
Procurement cost per machine type m	200 -1,000
Relocation cost per machine type m per period	10
Upper Cell Size Limit	20
Lower Cell Size Limit	3

To better illustrate this, such 'grafting' enabled numerical values of certain parameters from some problems (e.g. unit inventory costs from the data sets of problems 5, 6, 7) to be exported to other ones (problems 1 to 4 and 8 to 15). To ensure that the data are as close as possible to reality, real-life raw data was used, collected from a company running in a CM environment. Emphasis was put on the number of part types, machine types, operations and number of cells. The summarized data from the surveyed company are shown in table 3-5.

Table 3-5 : Summary of design data collected from company implementing CM

No of products (part types p)	No of operations per product type	Number of cells within company	Number of machine types
12	5 - 9	5	42

Each one of the numerical examples used is solved as an integrated model and the solving ability of CPLEX is being tested as the problem size increases (number of variables and constraints). The computational times taken are compared with respect to the various problem sizes. The results and discussion are in the next section.

3.7. Computational Results and Discussion

The computational results, using model **A**, are presented in table 3-6, where elapsed time and final objective value are shown for each problem instance. Also shown are the number of part types p, the number of machine types m, the number of time periods t for which the design is performed, the number of cells c used in each problem and the number of operations K_p required for each part type. In this research, small-scale problems are defined as consisting of at most 3 part types, 5 time periods and 3 machine types. Therefore, problems 1 to 4 are small-scale ones. Large-scale problems consist of at least 15 part types, 5 time periods and 10 machine types. Thus, problems 12 to 15 are large-scale ones. Medium-scale problems are those whose number of part types, machine types and time periods lie between the small-scale and large-scale limits. So, problems 5 to 11 are the medium-scale ones. Real-life sized problem instances are represented by the problems which are large-scale or which lie in the upper end of the medium-scale limit, so that problems 9 to 15 are representative of real-life problems. The small-scale problems (problems 1 to 3) were successfully solved within 25 seconds. Small-scale problem 4 required more time than the other small-scale ones as 1,314 seconds (21.9 minutes) were required. The medium-sized problems (problems 5 to 8) required more computational times but were solved within up to 3,292 seconds (54.9 minutes). Problems 9 to 11 are considered to be a medium to large-scale problems. From these three problems, problem 11 required the most time as the optimal solution was found in 2,412 seconds (40.2 minutes). Therefore, the small to medium-scale problems (1 to 8) were solved within reasonable computational times, as were the larger ones (problems 9 to 11). All of the large-scale problem scenarios (12 to 15) proved to be too difficult for the simplex-based branch and cut algorithm of CPLEX to solve since no optimal solution was obtained after more than 6 hours of computation. In fact, CPLEX stopped solving problems 12 to 15 due to insufficient memory. The objective function values shown for such problems are the best feasible integer solutions obtained. This clearly indicates that the branch and cut algorithm of CPLEX is unable to produce good quality solutions in reasonable computational times for large-scale problems of the CM model, even when using the linear model **A**.

Table 3-6 : Summary of computational results for model A using the exact solution (ES) procedure

Problem scenario	Source	(p)	(t)	(m)	(K_p)	(c)	Objective Function Value ($OBJA_{ES}$)	Computational Time (seconds)
1	Small-scale	1	2	3	5-9	2	1,233,860	0.01
2	Small-scale	1	5	3	5-9	2	3,075,620	0.20
3	Small-scale	2	5	3	5-9	2	5,346,645	24.69
4	Small-scale	3	5	3	5-9	3	8,018,250	1,314
5	Cao and Chen (2004)	4	3	4	5-9	2	5,948,922	1,023
6	Chen and Cao (2004)	5	6	5	5-9	4	15,520,923	2,292
7	Mungwattanna(2000)	11	2	10	5-9	3	8,396,577	1,843
8	Nsakanda *et al.* (2006)	15	1	15	5-9	3	5,730,765	1,368
9	Defersha and Chen (2006)	25	2	10	5-9	5	19,041,767	1,904
10	Nsakanda *et al.* (2006)	40	1	20	5-9	5	15,224,592	2,201
11	Nsakanda *et al.* (2006)	20	1	20	5-9	5	7,634,805	2,412
12	Large-scale	15	5	15	5-9	3	28,592,575	11,790*
13	Large-scale	25	5	10	5-9	5	47,637,300	8,090*
14	Large-scale	40	5	20	5-9	5	76,243,715	22,578*
15	Large-scale	20	5	20	5-9	5	38,110,450	19,321*

* Search was stopped due to memory limitations and the objective function value reports the value of the best feasible integer solution found so far.

Problem 6 from Chen and Cao (2004) is selected in the rest of the discussion to illustrate the various features of the solution obtained for the developed CM model **A**. The objective function values obtained with model **A** and in Chen and Cao (2004) cannot be compared because of the different objective costs involved; this also applies to the other problems selected from existing literature.

Thus, a comparison of these different objective costs would not yield any meaningful information. However, a meaningful comparison can be made in terms of the CM decision aspects provided by the model **A**. The approach with model **A** is comparably more general since several additional manufacturing aspects are considered: production planning in terms of inventory holding, the outsourcing option, internal part production cost and intra-cellular material handling cost. The solution obtained with model **A** for problem 6 is detailed out in the rest of this section to illustrate the part-machine cell allocation, some sample part process routings, and production plans for some part types.

Table 3-7 shows the part-machine cell allocation for period t=1 using model **A**. In the configuration shown for the two periods, parts are produced in multiple cells and/or different machines. The table shows the different cells, the machine types (and number of machines) that they each contain, as well as the allocation of parts to cells and machines for internal production. The presence of alternate part process routings is reflected by the fact that similar operations of a part type are carried out on different machines and/or machine types and/or cells. For instance, operations 1 to 4 of part type P1 occur in two different cells, whereby in cell C2, three units of machine type M1 are used. They are also carried out in cell C3 where one unit of machine type M2 is being used.

Table 3-7 : Part process outings with corresponding part-machine cell allocation for period t=1 of Model A

Cell	Machines / Part types	P1	P2	P3	P4	P5
C1	M1=3			**K1 K2 K3**		**K1 K2 K3 K4**
	M3=3	**K5 K6 K7 K8**				
	M4=4	**K5 K6 K7 K8**	**K3 K4 K5 K6**			**K5 K6 K7 K8**
C2	M1=3	**K1 K2 K3 K4**				**K1 K2 K3 K4**
	M3=1				**K2 K3 K4 K5**	
	M4=1				**K2 K3 K4 K5**	
	M5=2		**K3 K4 K5 K6**	**K4 K5 K6 K7**	**K2 K3 K4 K5**	
C3	M2=1	**K1 K2 K3 K4**				
	M5=1				**K2 K3 K4 K5**	
C4	M2=2		**K1 K2**		**K1**	**K1 K2 K3 K4**
	M3=2		**K3 K4 K5 K6**			
	M5=1			**K4 K5 K6 K7**		

The process routings for part type P3 in period 1 are shown in table 3-8 to demonstrate the machines and cells visited by part P3 when undergoing each one of its required operations. Operations 1 to 3 all take place in cell C1 on machine type M1 with 30,426 parts processed. Operations 4 to 7 are carried out in two different cells: 19,533 parts undergo all of these operations in cell C2 on machine type M5 whilst 10,893 are processed in cell C4 on machine type M5.

Table 3-8 : Process routings for part type P3 in period 1 of problem 6 for model A

Operation	1	2	3	4	5	6	7
	30426	30426	30426	19533	19533	19533	19533
	M1/C1	M1/C1	M1/C1	M5/C2	M5/C2	M5/C2	M5/C2
				10893	10893	10893	10893
				M5/C4	M5/C4	M5/C4	M5/C4

Table 3-9 shows how part demands are met for sample part types P3 and P5 through internal production, inventory holding and outsourcing for the whole planning horizon. Within model A, since the option of keeping inventory is considered, the system leverages the excess capacity of capable machines to produce surplus quantities of parts P3 and P5 in certain periods to meet demands in future periods. Although the outsourcing option is also viable, the solution demonstrates that parts should not be sub-contracted in any time period.

Table 3-9 : Comparison of production plans for parts P3 and P5 over the whole planning horizon for problem 6

Time period/ Parts		P3	P5
T1	Internal production	30,426	30,927
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,426	30,927
T2	Internal production	30,444	38,590
	Outsourced	0	0
	Inventory held	0	8,023
	Demand	30,444	30,567
T3	Internal production	56,894	28,371
	Outsourced	0	0
	Inventory held	26,275	5,538
	Demand	30,619	30,856
T4	Internal production	4,164	35,552
	Outsourced	0	0
	Inventory held	0	10,594
	Demand	30,439	30,496
T5	Internal production	30,994	24,793
	Outsourced	0	0
	Inventory held	265	5,033
	Demand	30,729	30,354
T6	Internal production	29,851	25,552
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,116	30,585

To illustrate the information presented in table 3-9, consider the production plan for P3 in period 5. Although the part demand is 30,729 internal production of P3 results in 30,994 being produced. The surplus parts (265) are carried over to period 6 and contribute to meet the demand of 30,116 parts. Therefore, in period 6, an internal production of 29,851 parts of P3 is required. Since model **A** also considers the minimization of the amount of intracellular cost involved in consecutive operations, the cellular layout is more compact. In fact, a trade-off is made between intercellular and intracellular movement by simultaneously minimizing both of these costs within the objective function. This is important since, on the one hand, high intra-cellular costs imply that the cells are quite large, reducing the effectiveness of the CM system. On the other hand, high amounts of intercellular movement increase the dependence of cells on one another, increasing the coordination effort required between cells and adversely affecting the benefits of CM systems. In many of the models, the trade-off between intracellular and intercellular material handling is not sufficiently addressed as intracellular movement is ignored. This type of trade-off is only considered in Nsakanda *et al.* (2006) but the model in the latter does not consider several important manufacturing aspects including inventory holding, internal production cost, machine maintenance and overhead costs, multi-period planning and dynamic cell reconfiguration.

3.8. Chapter Summary

This chapter has presented CM model **A** which integrates multi-period production planning, dynamic system reconfiguration, alternate part process routings and several important manufacturing aspects. Having first formulated the model as a mixed integer non-linear program (**F**), some small-scale problems were solved up to a certain problem size. Motivated by the computational difficulty of the non-linear formulation **F**, some linearization techniques have been proposed to transform the model into a mixed integer linear programming formulation **A**. The small-scale models were again solved using the new linearized model **A**, resulting in better computational times, and larger problems were also successfully solved. Some more experiments were performed with this linearized model using problem data sets from literature to find the size limits of solving the linearized model using CPLEX. Small to medium-sized problems could be solved within up to 1 hour of computational time. The large-scale problems proved to be more difficult to solve using the proposed exact solution approach with model **A** as no optimal solution was obtained after up to 6 hours of computation. The best integer solutions obtained for these large-scale problems displayed optimality gaps of less than 1%. However, the solutions from this integrated model **A** show that additional CM structural and operational design decisions that were not considered in previous research can be addressed with the proposed model **A**. The next step in research is the investigation of the use of meta-heuristics, namely tabu search, to solve problems of larger scale for this integrated CM model. This is dealt with in chapter 4.

Chapter 4

Tabu Search Solution Procedure for Integrated Model with Production Planning and Dynamic System Reconfiguration

Chapter 4 describes the tabu search (TS) procedure to solve CM model **A.** Short-term memory aspects and long-term memory strategies (through search intensification and diversification) are applied within the tabu search procedure. An overview of the tabu search implementation is presented, followed by a tabu search framework explaining how the meta-heuristic is adapted to model **A**. Explanations for the various TS aspects are then given, followed by the inclusion of the pseudo-code used. The same numerical examples used in chapter 3 for model **A** are then solved using TS. The computational results obtained from the tabu search procedure are then compared with those from the exact solution (ES) procedure (simplex-based branch and cut algorithm solving a linearized mixed integer program **A**).

4.1. Implementation of Tabu Search to the CM Design Problem

The tabu search approach used makes use of the short-term aspects of tabu search by using recency-based memory and the long-term aspects through the use of residency-based frequency measures. The tabu search starts with a feasible integer solution, known as the seed solution. The latter is obtained by first selecting a feasible cellular configuration (in terms of machines assigned to cells in all time periods) for model **A** and by then using this partial integer solution to solve for the overall feasible solution for the overall problem **A** (see figure 4-1).

During the tabu search procedure, the neighborhood of a current solution is explored to obtain a candidate list of feasible neighborhood solutions. Within this neighborhood, a best candidate is selected (upon certain conditions) and used to initiate the search at the start of the next iteration. By choosing a best candidate solution from a list of neighborhood solutions, the tabu search does not limit itself to a local solution N_L but chooses the best possible solution N_B within a list of neighborhood candidates. Since non-improving moves can be accepted when the move is not tabu, various neighborhoods within the search space can be explored. The list of neighborhood solutions represents the size of the neighborhood explored and is called the *candidate list size* (*CLS*). Within each explored neighborhood, a *local aspiration criterion* is established to bias the search for neighborhood solutions in such a way as to select the move operator and move attributes that bring improvements in the neighborhood solution objective cost. The *local aspiration criterion* is set to the value of the best candidate solution found so far and is updated firstly, as better candidates are found within a specific neighborhood, and secondly, as other neighborhoods are explored.

At each iteration and after each explored neighborhood, the attributes of the move that lead to the chosen best candidate are compared to those recorded in one of two tabu lists. This forms part of the recency-based memory structure used within the short term component of the tabu search. These tabu lists are used to determine whether a move leading to a best candidate solution is to be classified as tabu or not. They are updated on a First-In-First-Out basis. The first tabu list, *TL1*, records move attributes belonging to move operator 1, when machines are added to cells. The second tabu list, *TL2*, records those associated with move operator 2, where machines are removed from cells. Once the attributes of a move are added to the corresponding tabu list, they remain tabu active (they have tabu tenure) for a given number of iterations, whose value is equal to the *tabu list size*. Move attributes of move operator 1 are compared to those recorded in tabu list *TL2* to determine the tabu status of the move. For instance, consider that the move attributes of operator 1 consist of 2 machines of type M1 added to cell C1 at time T1, and that within tabu list *TL2*, the following attributes of operator 2 have been recorded: 2 machines of type M1 removed from cell C1 at time T1. In this case, the move operator 1 is classified as tabu because applying it would lead the search to cycle back to a previously visited solution. Therefore, this prevents the search from accepting move attributes that lead to previously visited solutions. However, if the move passes the *global aspiration criterion* test (the best candidate obtained is better than the overall best solution found thus far), the tabu active status is over-ridden and the move is accepted. The *global aspiration criterion* used is the cost of the best solution found up to that iteration and gets updated whenever the best solution is improved.

Long term memory, within the integration of the intensification and diversification schemes, is used to strategically guide the search whenever there is no more improvement in the search. Upon a certain number of non-improving iterations, intensification starts so that attractive regions are revisited more thoroughly (this is achieved by increasing the size of the neighborhood explored by increasing *CLS* and by allowing for a larger number of solutions to be accepted by decreasing *TLS*). Reverting to these attractive regions means that certain elite solutions must be recorded and that an elite selection strategy must be applied to select a new starting point for the search. In this case, explicit long term memory is used whereby complete best candidate solutions are recorded on a First-In-First-Out basis. A list of three complete elite best candidate solutions is maintained throughout the search. When intensification occurs, the elite solution with the lowest cost is selected as the restarting point and erased from the long-term memory.

The diversification scheme is triggered at a higher value of non-improving iterations than the intensification scheme. Diversification enables the search to explore unexplored regions of the solution space after having determined that the neighborhoods that have already been explored do not contain any better solutions than the best one found so far. This scheme has the advantage of preventing the search from getting trapped around any local optima. The tabu lists are emptied, the local and global aspiration flags are re-initialized, and the tabu flags are de-activated during the diversification phase. Before presenting a step-by-step procedure for the tabu search, the overall tabu search procedure is explained in more detail through the framework shown in figure 4-1 and the important concepts and features of the meta-heuristic are presented.

The framework for the tabu search process is illustrated as a schematic in figure 4-1. Given the mixed integer linear programming formulation **A** the tabu search procedure starts by feeding a feasible partial integer seed solution **S1** into problem **A** to obtain a partially solved problem (named **A2**). The seed solution **S1** is a CM system where a feasible CM layout has been established in terms of machines assigned to cells in all time periods. Problem **A2** is, therefore, problem **A** having been partially solved, but with other decision variables still to be determined, namely internal production quantities, outsourced quantities, inventory to be held and part process routings. Solving problem **A2** through CPLEX gives a feasible solution for problem **A**, and under a CM configuration specified by seed solution **S1**. The essence of the TS adaptation is to use move operators SW1 and SW2 in order to modify each one of these pre-specified CM configurations. By applying the move operators to obtain various neighborhood solutions, problem **A2** is iteratively solved under various CM configurations. In effect, various solutions to the main problem **A** are being explored.

At the first iteration, problem **A2** is solved through CPLEX to obtain a feasible seed solution (which becomes the current solution). For this current solution, the move operators are applied to generate partial feasible neighborhood solutions **S2**, which are in turn fed into **A**. The various partially solved problems **A2** are solved, so that a set of complete neighborhood solutions is obtained according to the *candidate list size*. From this set of neighborhood solutions, the best candidate solution is selected in accordance with the *tabu list* and *global aspiration criterion* conditions on the move attributes used. Since this is the first iteration, the value of the best solution cost is equal to that of the selected best candidate cost. Depending on the number of consecutive non-improving iterations, the intensification and diversification schemes can be triggered.

At the end of each iteration the *stopping criterion* is tested. The *stopping criterion* is reached whenever there are 30 consecutive non-improving iterations. As long as the *stopping criterion* is not reached, the best candidate solution found is used to be the current solution whose neighborhood will be searched at the next iteration. The exceptions are when either the intensification or diversification schemes are triggered. Then, the current solution would be the new restarting solutions determined by the intensification and diversification schemes.

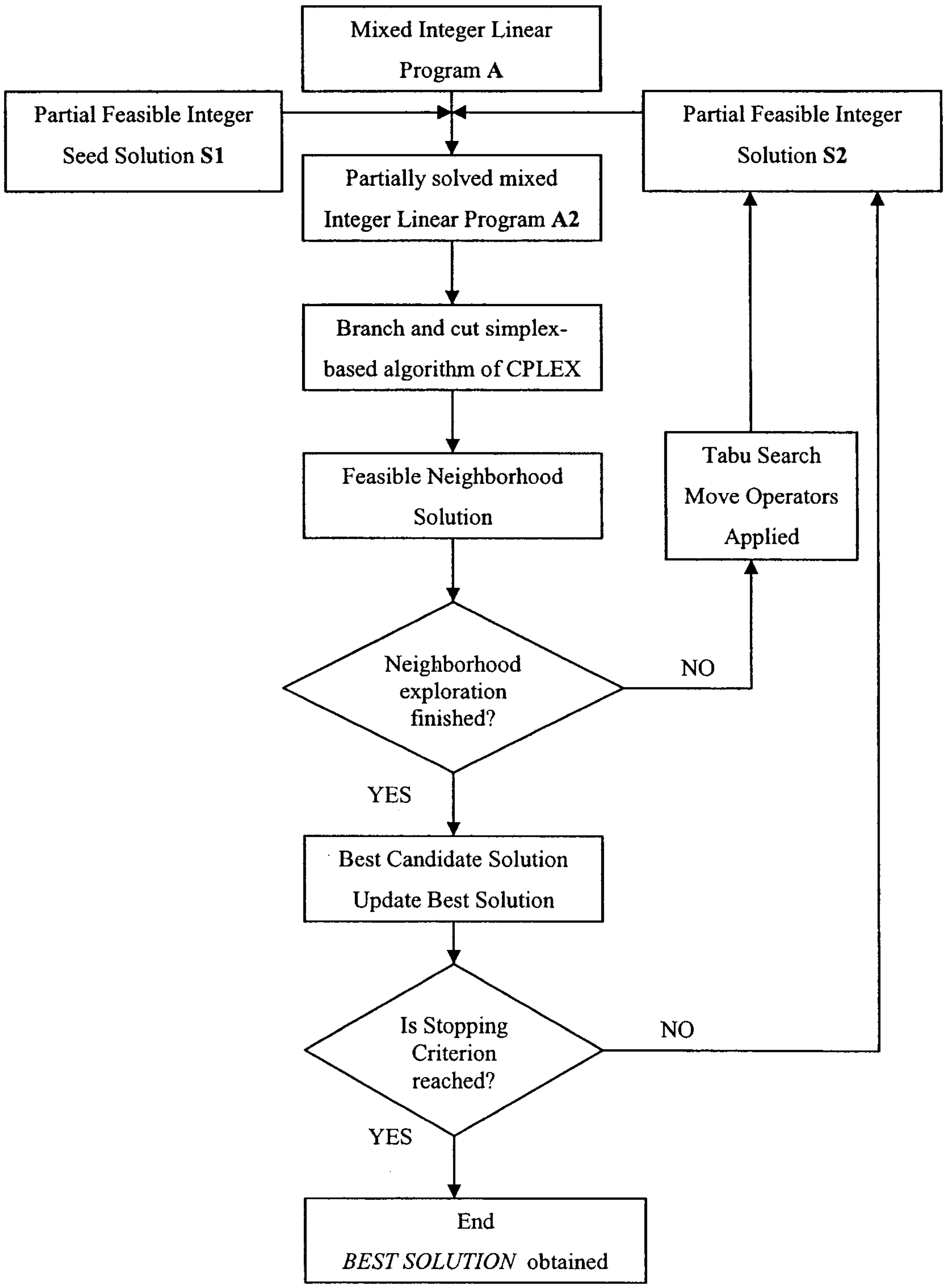

Figure 4-1 : Schematic representation of tabu search implementation for model A

4.2. Tabu Search Features

This section explains the different important individual aspects of the TS procedure in more detail. Consider the partial feasible integer solution **S2** representing a feasible cellular configuration obtained after move operators have been applied to a current solution. An example of such a partial neighborhood solution is given in figure 4-2, where 7 machines have been assigned to 3 cells. In specific, one unit of machine type M2 is assigned to cell 1 and one unit of machine type M6 to cell 3. Cell 2 contains one unit of machine types M3, M4 and M5, as well as 2 units of machine M1.

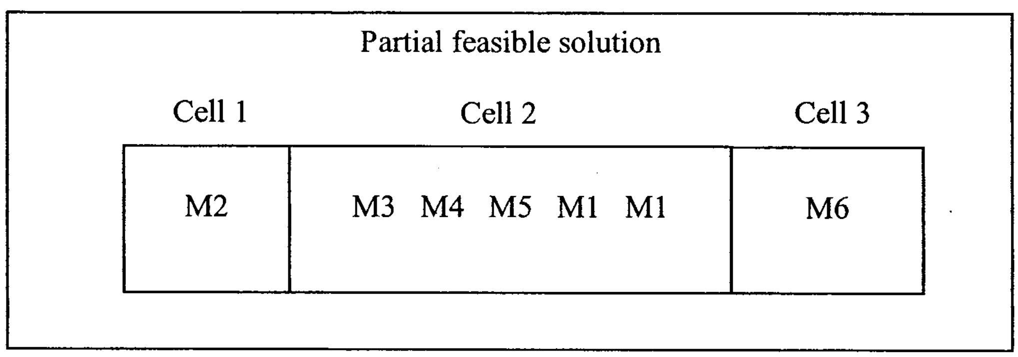

Figure 4-2: Representation of a feasible partial integer solution S2

It is through the use of such a cellular configuration that decisions will be made on several other variables used in the overall design of the CM system. For instance, values for the decision variables for internal production, outsourcing and inventory holding quantities, as well as the best process routings for all parts, are obtained by feeding this partial integer solution **S2** (feasible CM configuration) into the main problem **A** before solving the resulting modified linear mixed integer program **A2** (see figure 4-1). In fact, model **A** is being partially solved to obtain problem **A2** by incorporating the partial solution **S2** into it. **S2** represents a feasible cell configuration (machine allocation to cells), which is decided upon while respecting machine and cell model constraints 7 (upper limit on cell size), and 8 (lower limit on cell size) when the TS move operators are applied. The allocation of machines to cells in a certain time period is reflected through the decision variable $N_{mc}(t)$ and the latter takes fixed values in any feasible solution **S2.** This means that decision variable $N_{mc}(t)$ becomes a parameter in problem **A2**.

The problem to be solved **(A2)** is now shown below, with the cell formation decision variable $N_{mc}(t)$ being replaced by the cell formation parameter $\hat{N}_{mc}(t)$ so that the terms where such a variable occurs can be eliminated from the problem. Thus objective term (1.2), and constraints (7) and (8) can be eliminated from the model. Constraint (27) is the modified form of machine availability constraint (6). Constraint (28) represents the machine relocation constraint in problem **A2**, having been modified from constraint (9) in model **A**. Constraint (29) represents the machine capacity constraint in model **A2**, having been modified from constraint (10) in model **A**.

Model **A2:**

Minimize (1.1), (1. 3) to (1.6), (1.9), (1.10), (1.11)

subject to: (2) - (5), (11) - (14), (16) – (26)

$$\sum_{c \in C} \hat{N}_{mc}(t) \leq A^{*}_{m}(t) \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (27)$$

$$\hat{N}_{mc}(t) = \hat{N}_{mc}(t-1) + N^{+}_{mc}(t) - N^{-}_{mc}(t) \qquad \forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T} \qquad (28)$$

$$\sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}(p)} X_{kpmc}(t) \cdot e_{kpm} \le T_m \cdot \hat{N}_{mc}(t) \qquad \forall m \in \mathcal{Q}(k,p), \forall c \in \mathcal{C}, \forall t \in \mathcal{T} \tag{29}$$

Seed solution. The partial seed solution is denoted as **S1**. This is a generated by creating a feasible cellular configuration where machines of any type are assigned to cells in different time periods. This is shown in figure 4-3 below, where machine type M1 has been assigned to cell 1, M4 assigned to cell 2 and M5 assigned to cell 3. The neighborhood of the partial seed solution is explored to obtain other partial feasible solutions. Move operator SW1 (machine addition) is being applied to generate three possible neighborhood solutions. When generating these partial feasible solutions, it is important that constraints (7) and (8) are satisfied, so that machines are added to cells whilst respecting lower and upper cell size constraints. These partial feasible solutions will in turn be used to generate the relevant partially solved linear integer programs (**A2**), which will be solved to obtain feasible neighborhood solutions to problem **A**.

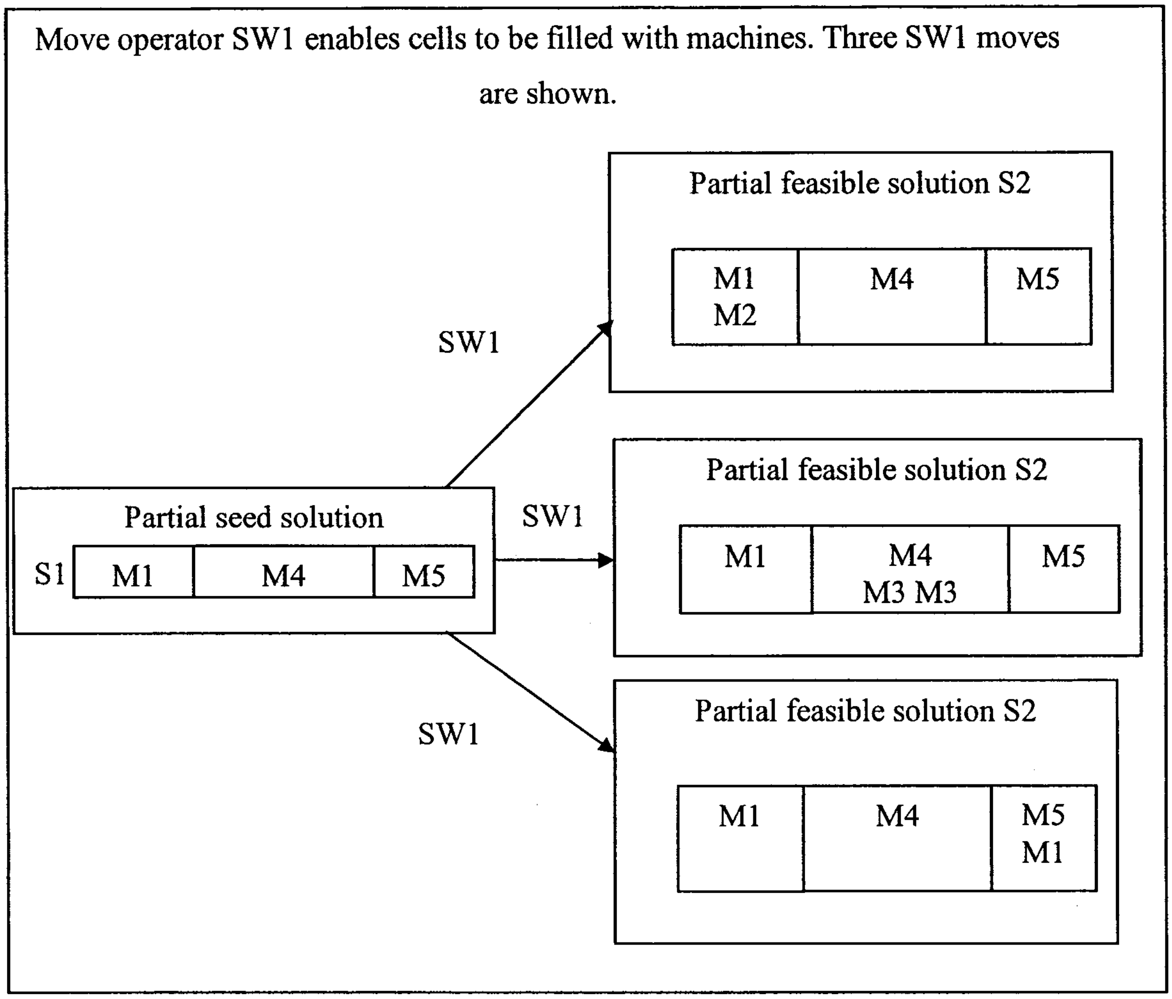

Figure 4-3: Move operator SW1 generating three possible neighborhood solutions during the construction phase of TS

Move operators: The moves involve changing a cell configuration in various ways with a view to obtaining other feasible CM configuration. The two types of moves (SW1 and SW2) used are illustrated in figures 4-3 and 4-4 and they define the different types of solutions explored in the neighborhood N(S) of S, where S is a current feasible partial solution. Both types of move operators are applied while cell size constraints. The first type of move, SW1, is used to gradually assign machines to cells. In figure 4-3, the first move operator assigns one unit of machine M2 to cell 1. The second move incorporates two units of machine M3 in cell 2 and the third move adds one unit of machine type M1 to cell 3. This type of move is favored in the early stages of the tabu search since the partial feasible solution consists of few machines having been assigned to cells. Therefore, this is the construction phase of the tabu search process. This type of move generates a neighborhood $N_{SW1}(S)$. The second type of move, SW2, enables only machine removal from cells. In figure 4-4, the SW2, when first applied, removes one unit of machine M2 from cell 1. Another feasible solution is when move operator SW2 removes two units of machine M3 from cell 2. This generates a neighborhood $N_{SW2}(S)$. By exploring various feasible configurations through these two sets of moves, and using these partial solutions to solve the associated partially solved problem **A2**, various neighborhood solutions are explored by tabu search.

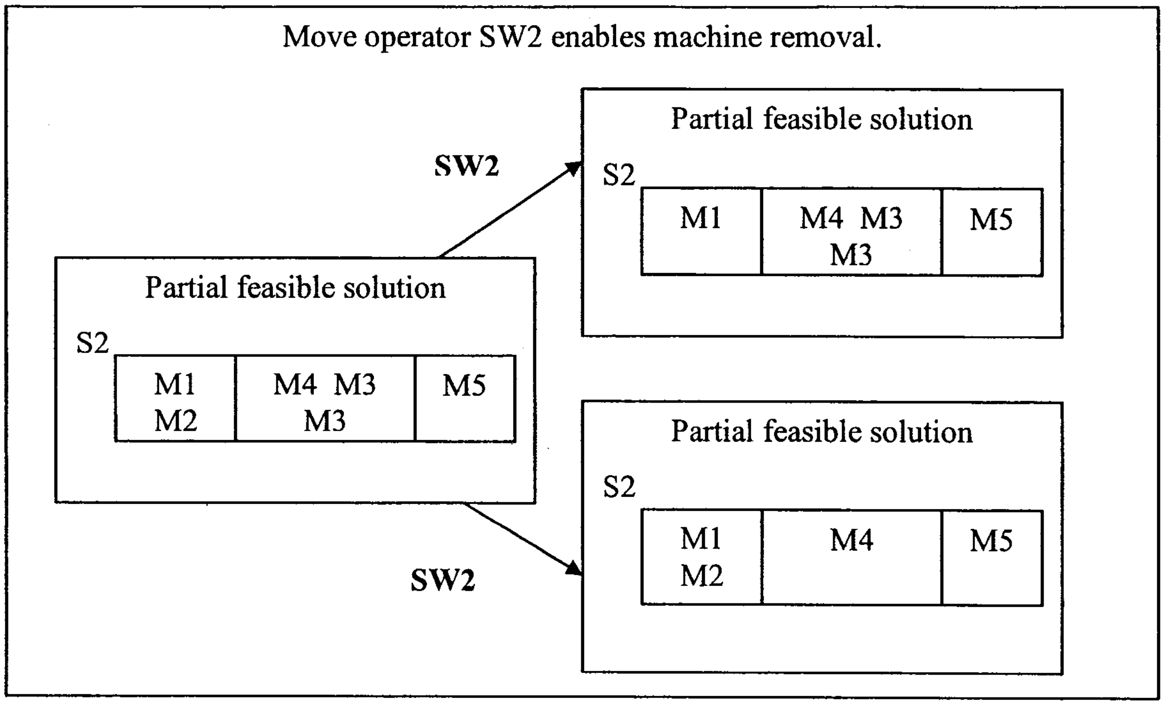

Figure 4-4: Move operator SW2 generating two possible neighborhood solutions during the machine removal phase of TS

Solution neighborhood. The neighborhood of a current solution is the ensemble of partial feasible solutions found by applying the two sets of move operators to obtain **A2** and solving the latter with CPLEX. These move operators define the different types of solutions explored in the neighborhood N(S) of S. The neighborhood N(S) is defined as the union of these two types of neighborhood, $N(S) = N_{SW1}(S) \cup N_{SW2}(S)$. Its size is the number of candidates that are explored and is governed by specifying the *candidate list size (CLS*). A *candidate list size* of 20 to 30 is found to work well in converging to good quality solutions in good computational times.

Tabu lists. The tabu search procedure uses recency-based memory and stores attributes of each chosen move operator (leading to a best candidate solution) within two tabu lists. Two lists are kept and updated. The first tabu list *TL1* records the move attributes that are associated with move operator SW1. It records the machine type being added, the cell to which it is added, the associated time period, the final number of machines of the same type in the cell as well as the number of machines being added. Whenever move operator SW2 is applied, its move attributes are checked against those recorded in *TL1*. If the move attributes are found in the list, then SW2 is classified as tabu. For example, consider if one of the records in tabu list *TL1* contains the following attributes when move operator SW1 was previously applied: 2 machines of type M1 added to cell C1 at time T1, with the final number of machines of type M1 in C1 at time T1 being 4 (after considering the two additional machines). Then, consider if move operator SW2 has the following attributes: remove 2 machines of type M1 from cell C1 at time T1, with four machines of type M1 being present in the same cell initially. As such, move operator SW2 is tabu since its attributes are tabu active. Therefore, the tabu list *TL1* prevents the move operator SW2 from cycling back to a previously visited solution. The second tabu list *TL2* works on the same principle, except that it records move attributes associated with move operator SW2. It records the number of machines removed, the machine type, cell and time period, and the final number of machines of the same type remaining in the cell. If the attributes of move operator SW1 are found to belong to tabu list *TL2*, then SW1 is classified as tabu. After some experimental runs, it is found that a *tabu list size* of 10 to 20 allows the TS procedure to converge to good quality solutions. Also, both tabu of the lists are of equal size and get updated on a First-In-First-Out basis.

Global aspiration criterion. The *global aspiration criterion* is set to be the value of the best solution cost found so far in the search and gets updated whenever the best solution cost is improved. It is used to over-ride the tabu active status of a move if the latter passes the *global aspiration criterion* test. If the attributes of a move are found to have tenure (that is, they are present in the tabu list and tabu active), the move has to be classified as tabu. However the *global aspiration criterion* test is applied to enable the tabu move to be accepted if the latter leads to a solution whose objective cost is lower than that of the *global aspiration criterion.* The *global aspiration criterion* test is applied for both of the move operators SW1 and SW2 to test moves that have been classified as tabu. This ensures that elite solutions are not discarded during the search and are sufficiently explored as they may produce better solutions.

Local aspiration criterion. The *local aspiration criterion* is used only when exploring the neighborhood of a current solution. Its value is set to the cost of the best candidate solution found so far in this particular neighborhood. Its value is updated whenever there is an improvement in the best candidate cost in the same neighborhood. At the start of each iteration, when exploration starts in a new neighborhood, the *local aspiration criterion* gets re-initialized and updated so that each one of the explored neighborhoods has its own local aspiration criterion value. The *local aspiration criterion* is used to bias the local search occurring in a particular neighborhood. A move passes the *local aspiration criterion* test whenever the neighborhood solution cost it creates is better than the best candidate cost found so far within the same neighborhood. The *local aspiration criterion* influences the type and attributes of the move used for the next explored neighborhood solution. For instance if the *local aspiration criterion* test is passed when, through move SW1, 1 unit of machine of type M1 is added to cell C1 at time T1, the next move leading to the next explored neighborhood solution will consist of similar move attributes and the same move operator. In fact the next move will add a higher number of units (2) of machine type M1 to cell C1 at time T1. The number of machines added increases by one unit whenever the same move attributes are chosen for the next explored neighborhood solution when the *local aspiration flag* is active.

Long-term memory structures. These are used by the TS intensification and diversification schemes to strategically guide the search when the short-term memory aspects of TS are no longer improving the solution. The first type of long-term memory used is explicit memory since it records and maintains a list of complete elite solutions. In fact, a list of three best candidate solutions is maintained and updated at each iteration. It is used by the intensification scheme, which selects the best candidate solution with the best (lowest) cost to restart the search. The second type of long-term memory is frequency based. It records the occurrence of solution attributes present in elite solutions. This is a residency measure where the characteristics of best candidate solutions are recorded at each iteration. For example, it counts the number of times that the decision variable $N_{mc}(t)$ takes a value of say, X. In other words, this long-term memory scheme is counting the number of times that an elite solution consists of X machines of type m assigned to cell c at time t. This is used by the diversification scheme to bias the search towards unexplored regions of the solution space.

Intensification. This aspect of long-term memory enables the search to restart from an elite solution when a pre-determined number of consecutive iterations (5) have not brought any improvement. The aim of intensification is to restart using the long-term memory structures and to further explore attractive regions of the search space. From the list of best candidate solutions, it selects the one with the best (lowest) cost and resumes the short-term part of the TS procedure. This chosen elite solution is at the same time erased from the list. The candidate list size (*CLS*) is increased and the tabu list size (*TLS*) for both lists is decreased to allow more solutions to be explored in the attractive regions

Diversification. This gets triggered at a higher number of consecutive non-improving iterations (10) and enables the search to escape local optima. The search resumes in other less explored regions of the search space. It occurs after the intensification scheme to ensure that all visited regions of the search space have been explored adequately. Long-term memory structures are used to keep track of the explored regions of the search space by the use of a frequency residency measure. Using the latter, solution attributes that are the least frequently found in elite solutions are used to build a new diversification-based restarting solution. The latter is used to resume the short-term TS module. At the same time all TS parameters (*CLS, TLS, tabu list flags, global and local aspiration flags*) get re-initialized to their original values (i.e. as in the short-term TS module). Also, both tabu lists are emptied. An example of a frequency residency measure is as follows for machine type M1 assigned to cell C1 at time T1:

N (1,1,1) = 2; frequency count = 5

N (1,1,1) = 4; frequency count = 2

N (1,1,1) = 1; frequency count = 10

This shows that the encountered solutions through the search most rarely yield 4 machines of type M1 assigned to cell C1 at T1, since its frequency count is the lowest at 2. Therefore, this will be used to build part of the restarting solution.

Stopping Criterion. This is used to stop the tabu search procedure and to display the value of the best solution. A possible *stopping criterion* is to set a value for a maximum number of iterations. The disadvantage with this approach is that the search can be stopped too early or too late. If it stops too early, it means that the search has not been allowed to converge sufficiently and this can give poor quality solutions. If it stops too late, it means that the search is continuing even though no better solution can be found. In this case, this leads to high values of required computational times to solve problems. In the light of such considerations, this research defines the *stopping criterion* as being reached when 30 consecutive non-improving iterations are experienced. This allows a good trade-off to be achieved between allowing the search to reach good quality solutions and reasonable computational times.

4.3. Tabu Search Pseudo-code

The following nine steps describe the implementation of tabu search to solve the linear mixed integer programming formulation of the CM model **A**.

- *Step 1.* Obtain an initial feasible *seed solution* **S1**. Feed **S1** into the main problem A to obtain the partially solved problem **A2**. Solve **A2** using CPLEX to obtain an integer feasible seed solution for **A** and its objective cost. Initialize *current solution = seed solution, best solution = seed solution.* Initialize *tabu list size* to 20 and the *candidate list size* to 20; set all *tabu, global* and *local aspiration flags* to inactive. Initialize number of consecutive non-improving iterations (*ni_iter*) to zero. Initialize number of consecutive non-improving iterations (*ni_iter_long_term*), used to trigger intensification and diversification, to zero.

- *Step 2.* Initialize *global aspiration flag* to inactive, *global aspiration criterion = best solution cost.* Initialize number of neighbors explored (*candidate_count = 0*). If iteration count = 1, *current solution = seed solution.* If iteration count >1, *current solution = best candidate solution.*

- *Step 3.* . Initialize *local aspiration flag* to inactive. Apply move operator (SW1 or SW2) to *current solution* to obtain a candidate *neighborhood solution* (**A2**). Record move operator and move attributes used. Solve A2 using CPLEX to obtain neighborhood solution cost.

- *Step 4.* If *candidate_count=1, best candidate = neighborhood solution*; *local aspiration criterion = best candidate cost.* Else, test for the *local aspiration criterion.* If *neighborhood solution cost < local aspiration criterion*, then activate the local aspiration flag and update: *best candidate cost = neighborhood solution cost* and *local aspiration criterion = best candidate cost.* If *local aspiration flag* is active, then keep the same move operator and attributes but increase the number of machines added (SW1) or removed (SW2) for the next explored *neighborhood* *solution*. If *local aspiration flag* is inactive for the next 10 explored candidate *neighborhood solutions*, then keep the same move operator but randomly change the move attributes. If *local aspiration flag* is inactive for the next 20 explored candidate *neighborhood solutions*, then change to the other move operator and select random move attributes to restart the search. If *candidate_count* = *CLS*, then go to step 5. Else repeat steps 3 to 4, and update *candidate_count* = *candidate_count+1*.

- *Step 5*. Update list of elite solutions with the *best candidate solution* from step 4 on a First-In-First-Out basis. Also inspect the complete solution attributes found in this *best candidate solution*, more specifically the number of machines of each type assigned to each cell in each time period. Update the frequency residency count to keep track of how frequently best candidate solutions have X machines of type m assigned to cell c at time t. Record move operator and move attributes used to reach the *best candidate solution*. Check for the *global aspiration criterion*: if *best candidate cost* < *global aspiration criterion*, then the *global aspiration flag* is activated; also, update: *best solution cost* = *best candidate cost* and *global aspiration criterion* = *best candidate cost*. If the *global aspiration flag* is active, then *ni_iter* = *0*, *ni_iter_long_term* = *0*. Else *ni_iter=ni_iter* + *1* and *ni_iter_long_term* = *ni_iter_long_term* + *1*.

- *Step 6*. Check tabu lists for the tabu status of the move that leads to the *best candidate solution*. If move operator SW1 is applied, its move attributes are checked against tabu list *TL2*. If they belong to the list, then the move is classified as tabu active. If move operator SW2 is applied, its move attributes are checked against tabu list *TL1*. If they belong to the list, then the move is classified as tabu active. If the move attributes of the move operator are not tabu active, then the move operator is accepted preceding the update: *current solution = best candidate solution*. If the move is tabu but the *global aspiration flag* is active, then the move operator is accepted preceding the update: *current solution = best candidate solution*. Else the move is rejected.

- *Step 7. Search Intensification Scheme.* If *ni_iter_long_term = 5*, the *Search Intensification Scheme* is activated. Using the list of elite solutions recorded in step 5, select the one with the best (lowest) objective cost and use it as the intensification restarting solution. Erase the selected elite solution from memory. Update: *current solution = intensification restarting solution*. Increase *CLS* to 30 and decrease *TLS* to 10. Go to step 2.

- *Step 8. Search Diversification Scheme.* If *ni_iter_long_term = 10*, the *Search Diversification Scheme* is activated. The frequency residency measure recorded in step 5 for each one of the best candidate solutions found is used to construct the diversification restarting solution. Update: *current solution = diversification restarting solution*. Decrease *CLS* back to 20 and increase *TLS* back to 20. Erase both of the *tabu lists*, clear *all tabu flags, local* and *global aspiration flags*. Reset *ni_iter_long_term = 0*. Go to step 2.

- *Step 9.* Check for the *stopping criterion.* If $ni\_iter = 30$, then the stopping criterion has been reached and search is stopped, and best solution found is displayed. Else if $ni\_iter < 30$, repeat steps 2 to 9.

4.4. Comparison of Exact and Tabu Search Solutions for the Model

The data sets used for all the tests are the same ones used in chapter 3 and will thus serve as a common platform to compare the two solution methodologies. The results of solving CM model **A** with the simplex-based branch-and-cut algorithm of CPLEX (denoted by ES) are also presented. The tabu search procedure is used to solve model **A** for the same problems until the *stopping criterion* is reached. Table 4-1 presents the computational results obtained through the use of the two solution methodologies for all of the problem instances. For each problem instance, table 4-1 reports the objective function values obtained by ES ($OBJA_{ES}$) and TS ($OBJA_{TS}$), in addition to the computational times required. The last column demonstrates the relative deviations of one procedure from the other. More specifically, $OBJA_{ES\text{-}TS}$ measures the improvement made by using the TS over the CPLEX branch and cut procedure (ES). It is calculated as $OBJA_{ES\text{-}TS} = (1 - OBJA_{TS}/OBJA_{ES}) \times 100\%$. Positive values mean that TS performs better, than ES in terms of the solution quality.

Table 4-1 : Computational results and comparisons for the 15 problem instances using ES and TS with model A

Problem Scenario	Simplex-based branch and cut algorithm (ES)		Tabu search (TS)		Comparisons
	$OBJA_{ES}$	Computation time (in seconds)	$OBJA_{TS}$	Computation time (in seconds)	$OBJA_{ES\text{-}TS}$ (%)
1	1,233,860	0.01	1,233,880	29.09	-0.002
2	3,075,620	0.20	3,075,970	234.74	-0.011
3	5,346,645	24.69	5,346,645	163.90	0.000
4	8,018,250	1,314	8,018,760	943	-0.006
5	5,948,922	1,023	5,948,982	866	-0.001
6	15,520,923	2,292	15,521,647	1,232	-0.005
7	8,396,577	1,843	8,398,122	1,452	-0.018
8	5,730,765	1,368	5,730,765	1,054	0.000
9	19,041,767	1,904	19,042,747	1,209	-0.005
10	15,224,592	2,201	15,225,297	1,599	-0.005
11	7,634,805	2,412	7,634,917	1,689	-0.001
12	28,592,575	11,790*	28,592,347	3,898	0.001
13	47,637,300	8,090*	47,636,900	4,232	0.001
14	76,243,715	22,578*	76,242,675	4,102	0.001
15	38,110,450	19,321*	38,109,127	3,900	0.003

* Search was stopped due to memory limitations and the objective function value reports the value of the best feasible integer solution found so far.

4.5. Discussion

In this section, the computational results shown in table 4-1 are discussed in terms of the computational times experienced and of the relative quality of the final solution obtained through the use of ES and TS. ES could only solve problems 1 to 11 to optimality. It could not solve larger sized problems 12 to 15 to optimality but still provides the best feasible solution found after the shown computational times. In fact, ES stopped solving these problems because of insufficient memory. TS was able to solve all of the 15 problems within up to 1.18 hours. The computational times for TS are determined by the time taken for the *stopping criterion* to be reached. For each problem instance, this would occur when the best solution found does not improve for 30 consecutive iterations.

In terms of the computational times shown in table 4-2, ES is seen to be outperformed by tabu search for most of the problems, and especially for large sized ones. The results show that even though ES is faster than TS for problems 1 to 3, TS consistently outperforms ES for problems 4 to 15 in terms of computational times. This result justifies the use of the tabu search procedures for solving medium to large scale problems during CM design. The computational results show that TS generates good quality solutions in reasonable computational times. The final solutions found by TS all deviate by less than 0.1% from those found by ES. In fact, the highest deterioration in solution quality by TS is for problem 7 with -0.018%. For problems 12 to 15, it is found that TS brings a slight improvement with respect to the final solutions found by ES.

4.6. Chapter Summary

Tabu search has been applied to solve the CM model **A** that integrates multi-period production planning, dynamic system reconfiguration, and alternate part process routings. Tabu search was developed to integrate both its short term tabu search components and its long-term memory structures (employing intensification and diversification). For problems of large size, the exact solution (ES) method was observed to have difficulties in solving model **A** to optimality, whereas tabu search proved to be efficient on such problems, providing good quality solutions in better computational times. When the emphasis of the search for near-optimal solutions is on good solution quality and short computational times, TS is a better solution methodology than ES. The next chapter involves enhancing model **A** in various ways to increase the system flexibility during CM design and to cope with disruptions such as machine breakdowns or failure of some part process routings.

Chapter 5

Integrated Model with Routing Flexibility

5.1. Introduction

Chapter 5 presents some important extensions to mathematical model **A**. Mathematical model **B** integrates all of the important manufacturing structural and operational features considered in model **A**, but also considers one important additional manufacturing aspect namely, an emphasis on increasing system flexibility by forming *contingency* routings for all parts along with *main* routings. CM system flexibility is an issue that warrants further research since some disadvantages of CM systems exist, especially with regards to the reduction of system flexibility caused by machine dedication to cells. In addition, a recurrent assumption when performing CM design is that there are no machine break downs and no scheduled machine maintenance. Model **B** addresses such an issue by providing part *contingency* (*backup*) routings that can be used when the part *main* routings are disrupted, for example by machine breakdowns. *Contingency routings* serve as backups so as to effectively address such issues and allow the cellular manufacturing system to operate in a continuous manner even in the event of such disruptions. Model **B** is solved through the small, medium and large problem instances from chapters 3 and 4, using the same exact solution procedure (ES). The computational results for model **B** are then discussed. Model **B** is also compared to model **A** in terms of the increased system flexibility versus the additional investment costs to be incurred.

5.2. Formal Description of the Model

5.2.1. Problem Definition

Model **B** also integrates the consideration of the classical cell formation problem, bridged with the manufacturing problems involved with multi-period production planning and dynamic system reconfiguration. A number of additional structural and operational issues are also incorporated. Most importantly, the proposed model emphasizes the creation of part *contingency* routings to achieve higher overall system flexibility. Such *contingency* routings bring flexibility to the CM system as parts can be re-routed from the unavailable *main* routings to the *back-up* routings to prevent any interruption in production. The absence of *contingency* measures in traditional CM design models creates CM systems that are vulnerable to any *noise* or disruption in the part production routings. An important assumption in previous models is that all machines are failure free, so that there are no break downs. The article by Das *et al.* (2007) is one of the few works which address this issue of machine reliability within CM design, where a model is developed that incorporates machine reliability values and failure rates. Thus, (*main*) process routes with the highest reliability values are chosen when designing the CM, rather than having *contingency* routings for each part type. Model **B** puts emphasis on increasing system flexibility by forming *contingency* routings for all parts along with the *main* routings. Within the model **B**, both types of routings are formed using the alternate part process routing definition reported in Uddin and Shanker (2002) and used in model **A**. At this point, it is important to stress the difference between *main alternate process routings* and *contingency alternate process routings*. First of all, a *process routing* is the set of machines or work centers that a part has to go through in order to undergo each one of its operations; *alternate process routings* occur when there are several machines that can be chosen for a specific part operation. In this research, *main alternate process routings* are the machines that are actively used to perform the operations of each of the part types under production. *Contingency alternate process routings* are the machines that have been simultaneously setup for each one of the parts and that are not activated until there is a disruption in one of the *main alternate process routings* for a part. As such, each part type has both a *main alternate process routing* and a *contingency alternate process routing*. In the event of one machine breaking down during any operation in its *main process routing*, the part can be immediately re-routed to the appropriate machine according to its *contingency process routing* in order to resume the operation that was under way, thus ensuring a smooth production. The overall objective of the model is to minimize the total cost of machine relocation, machine maintenance, machine operation, outsourcing, inventory holding, internal part production, intercellular material and intracellular material handling, and machine procurement. The notations used for the model are presented followed by the objective function, constraints and model properties. The developed CM model **B** can be formulated as a linear mixed integer program by modifying the objective function and certain constraints of model **A**, as well as by adding some new decision variables.

Additional Model Decision Variables

$NM_{mc}(t)$ Number of machines of type m present at cell c at time t and used for *main* routings.

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$NP_{mc}(t)$ Number of machines of type m present at cell c at time t and used for *contingency* routings.

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$XP_{kpmc}(t)$ The quantity of parts of type p that is processed by operation k on machine type m in cell c at time t. This is defined for *contingency* routings.

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

$ZP_{kpmc}(t)$ 1 if operation k for part type p is carried out on machine type m in cell c at time t, and 0 otherwise. This is defined for *contingency* routings.

$\forall k \in \mathcal{K}(p), \ \forall p \in \mathcal{P}, \ \forall m \in Q(k,p), \ \forall c \in \mathcal{C}, \ \forall t \in \mathcal{T}$

5.2.2. Objective Function and Constraints

Model B:

Minimize (1.1), (1.3) – (1.6), (1.9) - (1.11)

$$+ \quad \sum_{t \in \mathcal{T}} \sum_{c \in C} \sum_{m \in \mathcal{M}} \alpha_m \cdot \left(NM_{mc}(t) + NP_{mc}(t) \right) \qquad (1.12)$$

The objective function consists of nine cost components: machine relocation cost (1.1), machine operating cost (1.3), outsourcing cost (1.4), inventory holding cost (1.5), internal part production cost (1.6), machine procurement cost (1.9), intercellular material handling cost (1.10), intracellular material handling cost (1.11), and maintenance and overhead costs for machines involved in *main* and *contingency* routings (1.12).

Subject to:

(2) – (5), (11) – (14), (16) - (26)

The additional constraints of model **B**, along with relevant explanations, are given below.

$$XP_{kpmc}(t) \leq M \cdot ZP_{kpmc}(t) \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in \mathcal{Q}(k,p), \forall c \in \mathcal{C}, \forall t \in \mathcal{T} \qquad (30)$$

Constraint (30) performs the same role as constraint (2) but deals with the decision variables for setting up *contingency* part process routings.

$$\sum_{m \in \mathcal{Q}(k,p)} \sum_{c \in \mathcal{C}} XP_{kpmc}(t) + V_p(t-1) - V_p(t) + O_p(t) = D_p(t) \qquad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall t \in \mathcal{T} \qquad (31)$$

Constraint (31) ensures that part demand can be satisfied through the use of the *contingency* routings if any one of the *main* routings fails and becomes out of service. The demand can again be satisfied in each period *t* through internal production, and/or outsourcing, and/or inventory carried over from the previous period.

$$\sum_{c \in C} \left(NP_{mc}(t) + NM_{mc}(t) \right) \leq A_m^*(t) \qquad \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (32)$$

Constraint (32) relates to the machine availability constraint, where the total number of machines of each type assigned to all cells (for both *main* and *contingency* routings) is less than or equal to the machine availability in the same period (where, through constraints (4) and (5), the machine availability has been updated in each period considering the total number of machines procured).

$$\sum_{m \in \mathcal{M}} \left(NP_{mc}(t) + NM_{mc}(t) \right) \leq B_U \qquad \forall c \in C, \forall t \in \mathcal{T} \qquad (33)$$

$$\sum_{m \in \mathcal{M}} \left(NP_{mc}(t) + NM_{mc}(t) \right) \geq B_L \qquad \forall c \in C, \forall t \in \mathcal{T} \qquad (34)$$

The cell size is user-defined through constraints (33) and (34), where the cell size lies within lower and upper bounds. Machines dedicated to both *main* and *contingency routings* are accounted for in these constraints.

$$NP_{mc}(t) + NM_{mc}(t) = NP_{mc}(t-1) + NM_{mc}(t-1) + N_{mc}^{+}(t) - N_{mc}^{-}(t)$$

$$\forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T} \qquad (35)$$

Constraint (35) is for the machine relocation activity between time periods, where the number of machines of a type assigned to a cell during a period is equal to the number of machines in the previous period, plus the number of machines added and minus the number of machines removed. This ensures that in each period machines can be chosen from the whole amount that is available in the period and used to form either *main* or *contingency* routings. For instance, a machine that was used in a period as part of a *main* routing is available in the next period to be part of either *main* or *contingency* routings. It also ensures that within a period, machines used for *main* routings are not used for *contingency* routings at the same time. Two machines of the same type can be selected, even within the same cell as long as they are two different entities.

$$\sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}(p)} e_{kpm} \cdot X_{kpmc}(t) \leq T_m \cdot NM_{mc}(t) \qquad \forall m \in Q(k,p),\ \forall c \in C,\ \forall t \in \mathcal{T} \qquad (36)$$

Constraint (36) ensures that the required internal production capacities respect the available machine capacities whilst taking into account the quantity of all parts processed and the total number of machines assigned to all cells. This is done for production occurring on machines assigned to the *main* routings only.

$$\sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}(p)} e_{kpm} \cdot XP_{kpmc}(t) \leq T_m \cdot NP_{mc}(t) \qquad \forall m \in Q(k,p),\ \forall c \in C,\ \forall t \in \mathcal{T} \qquad (37)$$

Constraint (37) accounts for the required production and available production capacities for machines assigned to *contingency* routings. It ensures that the required internal production capacities respect the available machine capacities whilst taking into account the quantity of all parts that could require processing were *contingency* routings to be activated, and the total number of machines that have been assigned for such a scenario.

$$\sum_{c \in C} \sum_{m \in Q(k+1,p)} XP_{k+1,pmc}(t) = \sum_{c \in C} \sum_{m \in Q(k,p)} XP_{kpmc}(t) \qquad \forall k \in \mathcal{K}(p) / \{K_p\},\ \forall p \in \mathcal{P},\ \forall t \in \mathcal{T} \quad (38)$$

Constraint (38) also enforces the material flow conservation conditions but does so for the *backup* routings.

$XP_{kpmc}(t) \geq 0$ and integer $\quad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p),$

$$\forall c \in C, \forall t \in \mathcal{T} \tag{39}$$

$$NP_{mc}(t) \geq 0, \quad \text{and integer} \quad \forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T} \tag{40}$$

$$NM_{mc}(t) \geq 0, \quad \text{and integer} \quad \forall m \in \mathcal{M}, \forall c \in C, \forall t \in \mathcal{T} \tag{41}$$

$ZP_{kpmc}(t) \in \{0,1\} \quad \forall k \in \mathcal{K}(p), \forall p \in \mathcal{P}, \forall m \in Q(k,p),$

$$\forall c \in C, \forall t \in \mathcal{T} \tag{42}$$

Constraints (39) to (42) are the additional logical binary and non-negativity integer requirements on the decision variables. In the next section, several properties of model **B** are presented and discussed.

5.3. Properties of the Model

Model **B** takes a holistic approach in CM design in that it integrates several aspects of manufacturing which can be grouped into the key categories of production planning, subcontracting, dynamic system reconfiguration and cell formation. This model brings additional flexibility to CM systems as it encompasses the formation of *contingency* routings. These are further detailed in the rest of this section.

Cell formation with operation sequence and the coexistence of main and contingency routings. Contingency (backup) routings are used to enable parts to be re-routed to another sequence of machines for processing when the *main routing* is disrupted (for instance, by a machine breakdown in the *main routing*). Such *backup routings* are formed by taking into consideration the operations (and sequence) and the machine types (and capacities) required by each part type. Therefore, were a *contingency routing* to be selected instead of the *main routing* for a part type, the system would ensure that the re-routing does not affect the production of other part types. This is achieved by selecting a number of machines and assigning them to cells to form *contingency routings* that can be used to complete the production of any part type. This causes a decrease in the total available machine capacity as machines selected for *contingency routings* cannot be used for *main routings* in the same period, but increases the flexibility of the CM system into meeting the part demands. The inclusion of the machine procurement option enhances the formation of the *contingency routings* at a cost. A machine that has been chosen for a *main routing* in a certain period cannot be used for *contingency routings* in the same period. It can nevertheless be chosen in another period for *contingency routings*, provided it is not being used in *main routings*. However, since multiple copies of the same machine type are present, two machines of the same type can be used (separately) in *main* and *contingency* routings. This is illustrated in figure 5-1, where the *main* and *contingency* routings for a sample part *p* are presented, shown within part of a cellular layout, for the same time period. Part *p* has to undergo three operations to complete its processing, namely operations 1 to 3. In the *main* routing, part *p* undergoes operation 1 on one unit of machine type M1. In the *contingency* routing, operation 1 of part type is also carried out on machine type M1. Although these two machines are of the same type, they are different entities, meaning that there are two units of machine M1 being used. Operation 2 in the *main* routing occurs on machine type M2, whereas in the *contingency* routing machine M3 is used for the same operation type. Operation 3 occurs on another machine of type M3 in the *main* routing, while machine type M4 is used in the *contingency* routing. The presence of multiple copies of the same machine type enables machines of the same type to be selected in both types of routings. Prior to model **B**, no paper was found that discusses the possibilities and advantages arising from the formation of such *part backup* routings

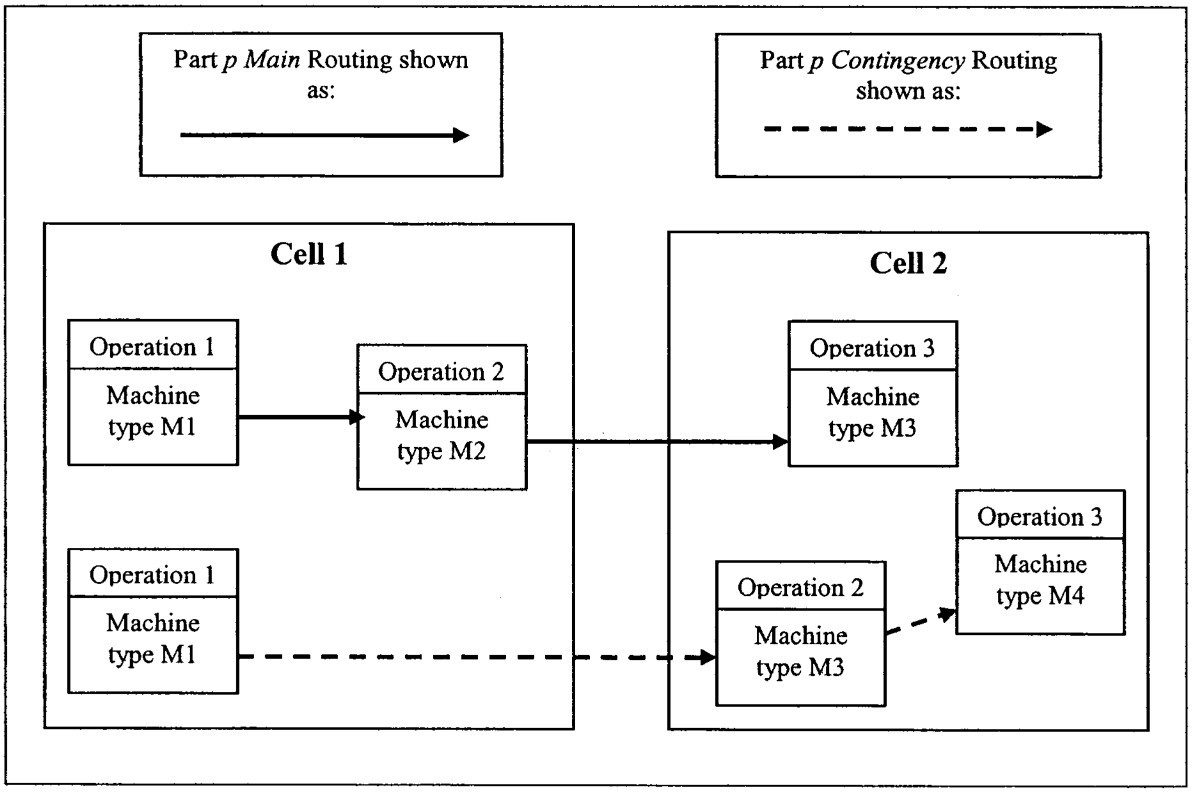

Figure 5-1 : An example of *main* and *contingency* process routings for a sample part in one period

5.4. Exact Solution Procedure and Numerical Examples for the Model

This section presents the computational results obtained by solving model **B** with the exact solution procedure (ES). The numerical examples used are the same as those in chapters 3 and 4. Detailed results for problem 6 are presented for model **B**, which will form the basis for comparing models **A** and **B** later in chapter 5.

5.4.1. Computational Results

Table 5-1 presents the objective cost obtained and the required computational times for solving the 15 problems using ES. It is seen that only 9 out of the 15 problems could be solved to optimality through model **B** using ES. Model **B** could not be solved for some of the medium-scale and all of the large-scale problem instances. This is explained by the fact that model **B** is more difficult to solve as a result of the additional decision variables and constraints that are introduced to have an increased CM system flexibility. In fact, the largest problem solved is problem 9, obtained from Defersha and Chen (2006). Thus, the small-scale problems were solved within 1,509 seconds (25 minutes). The medium to large sized problems (5 to 9) required up to 3,588 seconds of computational time (59.8 minutes). The large-scale problems could not be solved after up to 25,545 seconds (7 hours). However, the best integer solutions obtained for these large-scale problems had optimality gaps of less than 1%.

Table 5-1 : Summary of computational results for model B using the exact solution (ES) procedure

Problem scenario	Source	(p)	(t)	(m)	(K_p)	(c)	Objective Function Value ($OBJA_{ES}$)	Computational Time (seconds)
1	Small-scale	1	2	3	5-9	2	1,234,060	0.20
2	Small-scale	1	5	3	5-9	2	3,075,720	11.82
3	Small-scale	2	5	3	5-9	2	5,347,405	1,509
4	Small-scale	3	5	3	5-9	3	8,020,030	1,214
5	Cao and Chen (2004)	4	3	4	5-9	2	5,951,482	1,122
6	Chen and Cao (2004)	5	6	5	5-9	4	15,528,367	3,399
7	Mungwattanna(2000)	11	2	10	5-9	3	8,399,972	3,122
8	Nsakanda *et al.* (2006)	15	1	15	5-9	3	5,735,145	2,435
9	Defersha and Chen (2006)	25	2	10	5-9	5	19,042,287	3,588
10	Nsakanda *et al.* (2006)	40	1	20	5-9	5	15,227,992	14,323*
11	Nsakanda *et al.* (2006)	20	1	20	5-9	5	7,636,585	12,454*
12	Large-scale	15	5	15	5-9	3	28,596,685	25,553*
13	Large-scale	25	5	10	5-9	5	47,645,910	23,443*
14	Large-scale	40	5	20	5-9	5	76,248,054	25,545*
15	Large-scale	20	5	20	5-9	5	38,118,990	19,332*

*Search was stopped due to memory limitations and the objective function value reports the value of the best feasible integer solution found so far.

Problem 6 is used in the next section to illustrate various features of the solutions and information obtained from model **B**.

5.4.2. Discussion

Problem 6 for model **B** has been solved by ES and the rest of this section presents and discusses the part-machine cell partitioning for time period 1, sample *main* and *process* part routings, and production plans over the planning horizon. Table 5-2 shows all of the *backup* and *main* routings for the parts along with the machine-cell allocation for sample period 1 using model **B**. In the configuration shown, parts are produced in multiple cells and/or different machines, and have both *main* and *contingency* routings set up. Captions **A1** and **A2** refer to the information for *main* and *contingency* routings for all of the parts, along with the total number of machines of each type assigned to each cell during period 1. The total quantity of machines of a certain type is also shown. For instance, in caption **A1**, the machines shown are those that are assigned to *main routings* only. In caption **A2**, only the machines reserved for *contingency routings* are shown. To better illustrate the part-machine partitioning, consider caption **A1** where the operations K1, K2, K3 and K4 of part type P1 are carried out in cell 1 (on machine type M2) and in cell 4 (on machine type M2). In cell 1 there are 3 units of machine type M2 assigned for main *process routings*, whilst there are 2 units of machine type M2 in cell 4 again for main *process routings*. In comparison, in caption **A2**, it can be seen that the *contingency routings* for part P1 is as follows: operation 1 in cell C4 on machine M2, operation 2 in cell C4 on machine M1, operation 3 in cell C3 on machine M1, operation 4 in cell C4 on machine M2, operation 5 in cell C3 machine M3, operation 6 in cell C3 machine M5, operation 7 in cell C3 machine M3 and finally operation 8 is carried out in cell C1 machine M4 and in cell C4 machine M3.

Table 5-2 : Comparison of back-up routings and main routings for time period 1 using model B

Caption A1: Main routings with corresponding part-machine cell allocation for period $t = 1$

Cell	Machine/Part	P1	P2	P3	P4	P5
C1	M2=3	K1 K2 K3 K4	K1 K2			K1 K2 K3 K4
	M3=1			K4 K5 K6 K7		
	M4=1				K2 K3 K4 K5	
	M5=1			K4 K5 K6 K7		
C2	M2=3			K1 K2 K3		K1 K2 K3 K4
	M3=4	K5 K6 K7 K8			K2 K3 K4 K5	
C3	M1=1				K1	K1 K2 K3 K4
	M3=1		K3 K4 K5 K6			
	M5=1			K4 K5 K6 K7		K5 K6 K7 K8
C4	M2=2	K1 K2 K3 K4				K1 K2 K3 K4
	M4=4		K3 K4 K5 K6	K4 K5 K6 K7		K5 K6 K7 K8

Caption A2: Back-up routings with corresponding part-machine cell allocation for period $t = 1$

Cell	Machine/Part	P1	P2	P3	P4	P5
C1	M1=1		K1			K1 K2
	M4=2	K8	K6	K6		K7
	M5=1				K2	
C2	M1=2		K1	K1 K2		K3
	M3=1	K8		K4	K5	
	M5=1		K4		K5	
C3	M1=2	K3	K1 K2			K4
	M2=1				K1	
	M3=2	K5 K7	K5			K8
	M4=1					K5
	M5=1	K6			K4	
C4	M1=1	K2				K1
	M2=2	K1 K4		K3		K1
	M3=2	K8	K3	K7		K6
	M5=2			K5	K3 K4 K5	K8

Contingency routings serve as backups so as to effectively address the reality of part process routing disruptions owing to machine breakdowns and allow the cellular manufacturing system to operate in a continuous manner even in the event of such breakdowns. Table 5-3 illustrates detailed aspects of the chosen routings for part P3, pointing to the quantity of parts processed by each operation in each process routing as well as to the machines and cells in which the operation occurs. Caption **B1** shows the quantity of parts P3 involved in *main routings* and *backup routings* respectively, obtained through model **B**. Caption **B2** shows the quantity of parts P3 involved in *main routings* only (no *backup routings* are formed), obtained through model **A** for the same problem. A detailed explanation of the *main* routing for part P3 is given (refer to table 5-3). For operations 1 to 4, 30426 parts are processed on machine type M2 in cell 2. Operations 4 to 7 are performed as follows: 12499 on M3 in cell 1, 12500 on M5 in cell 1, 4374 on M4 in cell 4, and 1053 on M5 in cell 3. Owing to the model constraints and presence of multiple copies of each machine type, each one of the machines used in the *contingency* routings is not used anywhere in the *main* routings, and vice versa. Even if similar machine types are used for the same operation, these machines are different entities. This offers the flexibility and safety obtained through the formation of *contingency* routings, for instance, failure of a *main* routing (due to machine breakdown) can be addressed by re-routing the parts to the contingency routing, where they are allocated to the appropriate machines and cells. From table 5-3, it can be seen that the *contingency routing* for operation 1 of part P3 are as follows: 30426 parts can be processed on machine M1 cell C2. Therefore, if the machines M2 in cell C2 (involved in *main routings*) were to fail, up to 30426 units of part P3 can be re-routed to machines M1 in cell 2 to resume operation 1 and maintain a smooth production. The routings shown by caption **B2** demonstrate that any machine break down (or production disruption) cannot be addressed effectively since there are no *contingency routings* formed.

Table 5-3 : Main and backup process routings for part type P3 in period 1 of problem 6

Caption B1: Operations and routings obtained with CM model **B**							
Main routings with model B							
Operation	1	2	3	4	5	6	7
	30426	30426	30426	12499	12499	12499	12499
	M2/C2	M2/C2	M2/C2	M3/C1	M3/C1	M3/C1	M3/C1
				12500	12500	12500	12500
				M5/C1	M5/C1	M5/C1	M5/C1
				4374	4374	4374	4374
				M4/C4	M4/C4	M4/C4	M4/C4
				1053	1053	1053	1053
				M5/C3	M5/C3	M5/C3	M5/C3
Contingency routings with model B							
Operation	1	2	3	4	5	6	7
	30426	30426	30426	30426	30426	30426	30426
	M1/C2	/M1/C2	M2/C4	M3/C2	M5/C4	M4/C1	M3/C4

Caption B2: Operations and routings obtained with CM model **A**							
Main routings with model A							
Operation	1	2	3	4	5	6	7
	30426	30426	30426	19533	19533	19533	19533
	M1/C1	M1/C1	M1/C1	M5/C2	M5/C2	M5/C2	M5/C2
				10893	10893	10893	10893
				M5/C4	M5/C4	M5/C4	M5/C4

Table 5-4 shows how the part demands are met for sample part types P3 and P5 through internal production, inventory holding and outsourcing for the whole planning horizon. Within model **B**, since the option of keeping inventory is considered, the system leverages the excess capacity of capable machines to produce surplus quantities of part types P3 and P5 in certain periods to meet demands in future periods. Although the outsourcing option is viable, the solution demonstrates that parts should not be sub-contracted in any of the periods considered. To illustrate the information presented in table 5-4, consider the production plan for part types P3 and P5 for all 6 time periods. Although the part demand is 30,619 in period 3 for part P3, there is an internal production of 40,000 units of P3. The surplus 9,381 parts are kept as inventory and carried over to period 4. In period 4, the demand for P3 is 30,439 and the internal production has to only yield 21,058 parts, with the 9,381 inventory parts contributing to satisfy the demand in that period. For part P5, inventory is only kept in period 4. The surplus 102 parts produced in period 4 are carried over to period 5, where they are added to the 30,252 internally produced parts to satisfy the total demand of 30,354 parts of P5.

Table 5-4 : Comparison of production plans for parts P3 and P5 over the whole planning horizon for problem 6

Time period/ Parts		P3	P5
T1	Internal production	30,426	30,927
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,426	30,927
T2	Internal production	30,444	30,567
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,444	30,567
T3	Internal production	40,000	30,856
	Outsourced	0	0
	Inventory held	9,381	0
	Demand	30,619	30,856
T4	Internal production	21,058	30,598
	Outsourced	0	0
	Inventory held	0	102
	Demand	30,439	30,496
T5	Internal production	30,729	30,252
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,729	30,354
T6	Internal production	30,116	30,585
	Outsourced	0	0
	Inventory held	0	0
	Demand	30,116	30,585

5.5. Implications regarding the formation of contingency routings

To assess the effects of considering *contingency* routings in CM design to increase system flexibility, the objective costs obtained using both models **A** and **B** are compared, with emphasis on the additional investment costs that are incurred. The optimal values obtained for problems 1 to 9 are first used to compare models **A** and **B**. $OBJ_{ES(A\text{-}B)}$, gives the additional percentage cost of implementing *contingency routings*, using the results from ES. It can be calculated as $OBJ_{ES(A-B)} = \left(OBJB_{ES} / OBJA_{ES} - 1\right) \times 100\%$. Positive values indicate an increase in system costs and represent the additional percentage investment costs.

A comparison of the optimal costs from models **A** and **B** shows that the cost impact of implementing *contingency routings* is very low for all of the problems solved, with a percentage additional cost of less than 0.1%. In fact the highest additional percentage cost is 0.076% for problem 8, followed by 0.048% for problem 6, and the lowest additional percentage cost is 0.003 for problems 2 and 9. Therefore, this comparison of solutions obtained from the proposed models **A** and **B** shows that implementing *contingency routings* requires insignificant additional costs. An important observation is that this additional cost is offset by the increased routing and system flexibility brought about by *contingency* routings.

Table 5-5 : Comparison of costs from models A and B

Problem scenario	Model **A**	Model **B**	Cost Comparison
	Objective Function Value ($OBJA_{ES}$)	Objective Function Value ($OBJB_{ES}$)	$OBJ_{ES(A-B)}$ (%)
1	1,233,860	1,234,060	0.016
2	3,075,620	3,075,720	0.003
3	5,346,645	5,347,405	0.014
4	8,018,250	8,020,030	0.022
5	5,948,922	5,951,482	0.043
6	15,520,923	15,528,367	0.048
7	8,396,577	8,399,972	0.040
8	5,730,765	5,735,145	0.076
9	19,041,767	19,042,287	0.003
10	15,224,592	15,227,992*	0.022
11	7,634,805	7,636,585*	0.023
12	28,592,575*	28,596,685*	0.014
13	47,637,300*	47,645,910*	0.018
14	76,243,715*	76,248,054*	0.006
15	38,110,450*	38,118,990*	0.022

*Search was stopped due to memory limitations and the objective function value reports the value of the best feasible integer solution found so far.

A similar cost comparison is performed for problems 10 to 15 since the final best integer solutions found from both of the models (and corresponding problem instances) have negligible optimality gaps (less than 1%). From table 5-5, it can be seen that the highest additional percentage cost encountered is 0.023%, further justifying the fact that the benefits of implementing routing flexibility offset the required investment costs, especially for the medium to large-scale problems which are likely to occur in real-life scenarios.

5.6. Chapter Summary

In chapter 5, some extensions are brought to model **A** with a view to increasing the system flexibility of the proposed CM model. Model **B** is thus formulated to integrate *contingency routings* for all parts in addition to the *main part routings*. Several small to large sized problems have been solved through the mixed integer linear program **B** using the simplex-based branch-and-cut algorithm (ES). The solutions obtained demonstrate that *main routings* can be formed simultaneously with *contingency routings* for all parts in the designed CM system. One of the benefits of this approach to CM design is that the failure of a *main routing* can be effectively addressed by re-routing the parts being processed to the machine type and cell that is specified in the part *contingency routing* card. This is achieved through the proposed model since machines used in *main routings* are not used in *contingency routings* (and vice versa) within the same time period but can be of a similar type to a machine used in *contingency routings* (provided that it is a different entity). Therefore, model **B** allows CM designs to have flexible routes (*main* and *contingency* routings) that can address routing disruptions such as machine breakdowns and workload imbalance. Through model **B**, the CM system is also designed to address even the worst case scenario, whereby all *main process routings* for all of the parts have been disrupted, since all of the parts can be re-routed to their corresponding *backup routings*. The solutions obtained from model **B** show that implementing *contingency routings* can be done with insignificant additional costs. The additional cost to be incurred is offset by the increased routing and system flexibility brought about by *contingency* routings.

Chapter 6

Summary, Conclusions and Future Research

6.1. Summary and Conclusions

A lot of research has been conducted towards the design of cellular manufacturing systems in the last three decades. This type of manufacturing system has been found to bring several benefits as it is a promising strategy that can cope with mid-volume and mid-variety product demands. As such, the contributions of this research are in two main areas: firstly, the development of comprehensive mathematical models that integrate several important manufacturing aspects and, secondly, the development and implementation of efficient solution procedures to solve the proposed models, especially for real-life sized problems (represented by the upper end of the medium-scale problems and all of the large-scale problems solved in this research).

6.1.1. Contributions to CM System Design and Modeling

An effective design of CM systems has to take into account a number of structural and operational issues. In addition to the cell formation problem (assignment of machines and parts to cells), a number of key manufacturing attributes have to be integrated within CM design so that the benefits of CM systems can be achieved. In chapter 2, the review of proposed models and taxonomies enables several important manufacturing attributes to be identified:

1. Material handling
   a. Intercellular material handling costs
   b. Intracellular material handling cost
2. Part internal production cost
3. Subcontracting cost
4. Inventory holding in production planning
5. Part demand requirements
   a. Stochastic demand requirements
   b. Deterministic demand requirements
6. Multi-period planning
7. Cellular Manufacturing configuration
   a. Robust cell configuration
   b. Agile cell configuration
8. Machine characteristics
   a. Machines with multiple copies
   b. Machine with limited capacities
   c. Machine operating cost
   d. Machine maintenance and overhead cost
   e. Machine relocation and cost
   f. Machine procurement
9. *Main* part routings
   a. Alternate routings chosen from user-specified routings (limited subset)
   b. Alternate routings chosen from all possible options based on operation and machine type
10. Formation of part *contingency* routings

11. Part operation sequence and processing times.

12. Cell size limits – upper and lower bound

From the literature review, it is found that recent research on CM modeling tends to include only some of these manufacturing attributes. Indeed, a recurring theme is a piecemeal approach when formulating CMS models, as many of the existing models include only a limited subset of these manufacturing attributes. As such, this present research takes a holistic approach in CM design by integrating the identified important manufacturing aspects within two proposed CM models, **A** and **B**, thus bringing several important extensions to previous models.

In chapter 3, a comprehensive mathematical model (**A**) is presented and solved using an exact solution procedure (simplex-based branch and cut algorithm, referred to as ES). Numerical examples consisting of small to medium sized problems are solved, showing that model **A** allows better decision-making in CM design, especially in terms of multi-period planning, production planning and dynamic system reconfiguration. The CMS solutions generated through model **A** allow the cell designer to determine for a multi-period horizon:

- The alternate process routings for all of the part types.
- The machine to cell composition in each time period and machine relocation.
- Optimal values for internal part production, inventory holding and outsourcing.
- The number of machines to be procured.

In chapter 5, model **B** is developed with a view to incorporating system flexibility when designing CM systems. This is done to address the manufacturing reality pertaining to machine breakdowns, machines undergoing scheduled (or even unscheduled) preventive maintenance and workload imbalance. As such, model **A** is extended to include the formation of part *contingency* routings. These part *contingency* routings coexist with the *main* part routings and are used whenever disruptions occur within the latter. Therefore, failure of a *main* process routing can be addressed by re-routing the related parts to their planned part *contingency* routings, thus enabling production to occur in a continuous manner. Model **B** is also solved using ES, showing that it assists the system designer when designing flexible CM systems that can deal with disruptions in *main part process routings*. This research also demonstrates that the increased flexibility of model **B** offsets the additional associated investment costs.

6.1.2. Contributions to CM System Solution Procedures

In addition to developing comprehensive integrated models for Cellular Manufacturing systems, this research also contributes in developing efficient CM solution approaches. In chapter 3, it is found that model **A** can only be solved for small to medium sized problems when using an exact solution procedure, namely, the simplex-based branch and cut algorithm of CPLEX (ES). Larger problems cannot be solved even after more than 6 hours of computation.

Subsequently, in chapter 4, an efficient tabu search meta-heuristic based on tabu search is proposed and implemented to solve model **A**. The developed tabu search meta-heuristic uses short-term memory as well as long-term memory through the use of intensification and diversification schemes. TS outperforms ES in terms of computational time, especially for the larger sized problems. Also, TS is found to give good quality solutions that are very close to the final solutions found by ES.

6.2. Future Research

6.2.1. Multiple Criteria Decision Making for CMS Design

The objectives sought in model **A** and **B** are cost-oriented, with a number of different costs being considered. Performance-oriented objectives in CM design form part of future research work that are beyond the scope of this thesis, given the limited time frame. As such, these possible CM system criteria and objectives include performance measures that cannot be expressed in monetary terms and are as follows:

- Maintaining acceptable within-cell machine utilization levels
- Minimizing setup times
- Minimizing work-in-progress
- Maximizing the CM system reliability

CM models that address such multiple design criteria can be formulated as a multi-objective decision making (MODM) problem, which can be solved either via a constraint or weighted method. Goal programming is another approach that can be used for solving cellular manufacturing MODM problems.

6.2.2. Further CM Design Phases

This research deals only with the first phase of CM design, namely the cell formation problem. Additional research can be conducted for the two next phases in CM design. Further CM design work can center on phase 2, which deals with the layout of the cells and equipment within the manufacturing facility (facility layout problem), and phase 3, dealing with the layout of machines within the cells.

6.2.3. Further efficient solution procedures

One of the recent key trends is the use of parallelization techniques to solve various manufacturing design problems, including CM design. Parallelization techniques have been coupled with the use of genetic algorithms and simulated annealing to solve the CM design problem with encouraging results. A highly appealing avenue of research would be to apply these parallelization techniques to the use of tabu search when solving the CM problem. This could lead to greater benefits in terms of computational time and solution quality. For instance, an adaptation of parallelization within TS could be the simultaneous exploration of different solution neighborhoods.

6.2.4. Socio-technical Issues of CM Design

The organizational behavior aspects of implementing CM systems have not been sufficiently researched although there have been many reports extolling the worker-related benefits of implementing CM systems. CM models can also be developed that can assist in assigning (or re-assigning) operators to CM teams and in allocating (or re-allocating) teams to cells, based on the sets of competencies that each operator possesses and on the skills needed in the formed cells. Such models can help in considering the Human Resource Management aspects of CM design in addition to the structural and operational issues addressed in this research.

Bibliography

[1] Adil, G.K., Rajamani, D., and Strong, D., 1993. A mathematical model for cell formation considering investment and operational costs. *European Journal of Operational Research*, 69, 330-341.

[2] Askin, R. G. and Chiu, K. S., 1990. A graph partitioning procedure for machine assignment and cell formation in group technology. *International Journal of Production Economics*, 28, 1555–1572.

[3] Askin, R. G. and Vakharia, A. J., 1991. Group Technology - Cell Formation and Operation. In Automated Factory Handbook: Technology and Management. *D.I. Cleland and B. Bidanda, University of Pittsburgh (Eds). TAB Books, Inc.*, New York, 317–366.

[4] Askin, R.G., Creswell, J.B., Goldberg, J.B. and Vakharia, A.J., 1991. A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research*, 29, 1081 – 1100.

[5] 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, 281-309.

[6] Ballakur, A., 1985. An Investigation of Part Family/Machine Group Formation in Designing a Cellular Manufacturing System. *PhD thesis*, University of Wisconsin, Madison, WI.

[7] Ballakur, A., and Steudel, H.J., 1987. A within-cell utilization based heuristic for designing cellular manufacturing systems. *International Journal of Production Research*, 25, 639-665.

[8] Bertsekas, D., and Tseng, P., 1988. Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research*, 36, 93-114.

[9] Boctor, F., 1991. A linear formulation of the machine-part cell formation problem. International Journal of Production Research, 29, 343–356.

[10] Boctor, F.F., 1996. The minimum-cost, machine-part cell formation problem. *International Journal of Production Research*, 34, 1045-1063.

[11] Brandon, J.A., 1996. Cellular Manufacturing: Integrating Technology and Management. *Research Studies Press Ltd.*, 61-79.

[12] Burbidge, J.L., 1963. Production Flow Analysis. *The Production Engineer*, 42, 742.

[13] Burbidge, J.L., 1975. *The Introduction of Group Technology* (New York: Wiley).

[14] Burke, L.I., and Kamal, S., 1992. Fuzzy art and cellular manufacturing. In *Artificial Neural Networks in Engineering*, 779 – 784.

[15] Cao, D., and Chen, M., 2004. Using penalty function and Tabu search to solve cell formation problems with fixed cell cost, *Computers and Operations Research*, 31, 21-37.

[16] Cao, D., and Chen, M., 2005. A robust cell formation approach for varying product demands. *International Journal of Production Research*, 49, 1587–1605

[17] Carpenter, G. A. and Grossberg, S., 1987. A massive parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, 37, 54–115.

[18] Caux, C., Bruniaux, R., and Pierreval, H., 2000. Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics*, 64, 279-284.

[19] Chan, H. and Milner, D., 1982. Direct clustering algorithm for group formation in cellular manufacturing. *Journal of Manufacturing Systems*, 1, 65–74.

[20] Chandrasekharan, M. P. and Rajagopalan, R., 1986a. MODROC: an extension of rank order clustering for Group Technology. *International Journal of Production Research*, 24, 1221-1233.

[21] Chandrasekharan, M. P. and Rajagopalan, R., 1986b. An Ideal Seed Nonhierarchical Clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24, 451–464.

[22] Chandrasekharan, M. P. and Rajagopalan, R., 1987. ZODIAC-an algorithm for concurrent formation of part families and machine cells. *International Journal of Production Research*, 25, 835–850.

[23] Chen, M., 1998. A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. *Annals of Operations Research* 74, 109 – 128

[24] Chen, M., 2001. A model for integrated production planning in cellular manufacturing systems. *Integrated Manufacturing Systems*, 12, 275-284.

[25] Chen, M. and Cao, D., 2004. Coordinating production planning in cellular manufacturing environment using Tabu search. *Computers and Industrial Engineering*, 46, 571–588.

[26] Chen, S. J. and Cheng, C. S., 1995. A neural network-based cell formation algorithm in cellular manufacturing. *International Journal of Production Research*, 32, 293–318.

[27] Chen, C.L., Cotruvo, N., and Baek, W., 1995. A simulated annealing solution to the cell formation problem. *International Journal of Production Research*, 33, 2601–2614.

[28] Choobineh, F., 1988 A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, 26, 1161–1172.

[29] Chu, C.H., 1993. Manufacturing cell formation by competitive learning. *International Journal of Production Research*, 31, 829-843.

[30] Chu, C.H. and Hayya, J.C., 1991. A fuzzy clustering approach to manufacturing cell formation. *International Journal of Production Research*, 29, 1475 – 1487.

[31] Dagli, C. H. and Sen, C. F., 1992. Art1 neural network approach to large scale group technology problem. ASME Press, New York, 4, 787–792.

[32] Dake, S., Lin, L. and Batta, R., 1995. Cell formation using tabu search. Computers and Industrial Engineering, 28(3), 485 – 494.

[33] Das, K., Lashkaria, R.S. and Sengupta, S., 2007. Reliability consideration in the design and analysis of cellular manufacturing systems. *International Journal of Production Economics,* 105, 243-262.

[34] De Witte, J., 1980. The use of similarity coefficients in production flow analysis. *International Journal of Production Research*, 18, 503 – 514.

[35] Defersha, F., 2006. An Integrated Approach to the Design of Cellular Manufacturing Systems for Dynamic Production Requirements. . *PhD thesis*, University of Concordia, Montreal, Quebec.

[36] Defersha, F. and Chen, M., 2006. A comprehensive mathematical model for the design of cellular manufacturing systems. *International Journal of Production Economics*, 103, 767 – 783.

[37] Deutsch, S. J., Freeman, S. F., and Helander, M., 1998. Manufacturing cell formation using an improved p-median model. *Computers and Industrial Engineering*, 34, 135–146.

[38] Dimpoulos, C. and Zalzala, A.M.S., 1998. Evolutionary programming for designing manufacturing cells. Proceedings from the 1998 *IEEE International Conference on Evolutionary Computing, Piscataway NJ IEEE*, pages 148- 153.

[39] El-Essawy, I.F.K. and Torrance, J., 1972. Component flow analysis – an effective approach to production systems' design. *Production Engineer*, 51, 165.

[40] Elmaghraby, H.A, and Gu, P., 1989. Feature based expert parts assignment in cellular manufacturing. *Journal of Manufacturing Systems*, 8, 139-152.

[41] Faber, Z. and Carter, M. W., 1986. A new graph theory approach for forming machine cells in cellular production systems. *In Flexible Manufacturing Systems: Methods and Studies*, ed. A. Kusiak. North-Holland, New York, 301–318.

[42] Garey, M. and Johnson, D., 1979. *Computers and Intractability: A Guide to Theory of NP-completeness*. Freeman, San Francisco, CA.

[43] Glover, F., 1990. Tabu Search. A tutorial. *Interfaces*, 20, 74-94.

[44] Glover, F. and Laguna, M., 1997. *Tabu Search*. Boston: Kluwer.

[45] Greene, T. and Sadowski, R., 1984. A review of cellular manufacturing assumptions, advantages and design techniques. *Journal of Operations Management*, 4, 85 – 97.

[46] Gungor, Z. and Arikan, F., 2000. Application of fuzzy decision making in part-machine grouping. *International Journal of Production Economics*, 63, 181–193.

[47] Gupta, Y., Gupta, M., Kumar, A. and Sundram, C., 1995. Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach. *International Journal of Computer Integrated Manufacturing*, 8, 92 - 101.

[48] Gupta, Y., Gupta, M., Kumar, A. and Sundaram, C., 1996. A genetic algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research*, 34, 447–482.

[49] Heragu, S.S. and Chen J.S., 1998. Optimal solution of cellular manufacturing system design: Bender's decomposition approach. *European Journal of Operational Research*, 107, 175-192.

[50] ILOG Inc., 2006. ILOG Cplex 9.0.1 User's Manual. ILOG Inc., Mountain View, CA.

[51] Jayaswal, S. and Adil, G.K., 2004. Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings. *International Journal of Production Research*, 42, 2419 - 2433.

[52] Joines, J.A, Culbreth, C.T. and King, R.E., 1996. Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transactions*, 28, 69 – 85.

[53] Jones, D.F., Mirrazavi, S.K., and Tamiz, M., 2002. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137, 1-9.

[54] Josien, K. and Liao, T. W. R., 2002. Simultaneous grouping of parts and machines with an integrated fuzzy clustering method. *Fuzzy Sets and Systems*, 126, 1–21.

[55] Kaparthi, S. and Suresh, N., 1992. Machine-component cell formation in group technology: a neural network approach. *International Journal of Production Research*, 30, 1353–1367.

[56] King, J.R., 1980a. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, 18, 213–232.

[57] King, J.R., 1980b. Machine-component group formation in Group Technology. *OMEGA: The International Journal of Management Science*, 8, 193-199.

[58] King, J. and Nakornchai, V., 1982. Machine-Component Group Formation in Group Technology: Review and Extension. *International Journal of Production Research*, 20, 117 – 133.

[59] Kirkpatrick, S., Gelatt, C., and Vecchi, M., 1983. Optimization by simulated annealing. *Science*, 220, 671–680.

[60] Kusiak, A., 1987. The generalized group technology concept. *International Journal of Production Research*, 25, 561 – 569.

[61] Kusiak, A., 1988. Exgt-s: A knowledge based system for group technology. *International Journal of Production Research*, 26, 887–904.

[62] Kusiak, A. and Chow, W., 1987. Efficient solving of the group technology. Problem. *Journal of Manufacturing Systems*, 6, 117-124.

[63] Kusiak, A. and Chow, W., 1988. Decomposition of manufacturing systems. *Robotics and Automation*, 4, 457-471.

[64] Kusiak, A. and Chung, Y., 1991. Using neural networks to form machine cells. *Manufacturing Review*, 4, 293–301.

[65] Lee, H. and Garcia-Diaz, A., 1996. Network flow procedures for the analysis of cellular manufacturing systems. *IIE Transactions*, 28, 333–345.

[66] Lemoine, Y. and Mutel, B., 1983. Automatic recognition of production cells and part families. In *Advances in CAD/ CAM*, Ellis, T. M. R., and Semenkov, O. I., (Eds). North-Holland, Amsterdam, pp. 239–248.

[67] Liao, T.W., and Chen, L.J., 1993. An evaluation of ART1 neural models for group technology part family machine cell forming. *Journal of Manufacturing Systems*, 12, 282-290.

[68] Liao, T. W. and Lee, K. S., 1994. Integration of a feature-based cad system and an ART1 neural model for GT coding and part family forming. *Computers and Industrial Engineering*, 26, 93–104.

[69] LINDO Systems Inc., 2005. *Extended Lingo: Trial version + Users guide.* LINDO Systems Inc., Chicago.

[70] Logendran, R., Ramakrishna, P. and Srikandarajah, C., 1994. Tabu Search-Based Heuristics for Cellular Manufacturing Systems in the Presence of Alternative Process Plans". *International Journal of Production Research*, 32, 273 - 297.

[71] Lozano S., Adenso-Diaz, B. and Onieva, L., 1999. A one-step Tabu search algorithm for manufacturing cell design. *OR: the Journal of the Operational Research Society*, 50, 509 – 516.

[72] Lozano S., Dobado, D., Larraneta, J., and Onieva, L., 2002. Modified fuzzy c-means algorithm for cellular manufacturing. *Fuzzy Sets and Systems*, 126, 23-32.

[73] Luong, L., He, J., Abhary, K., and Qiu, L., 2002. A decision support system for cellular manufacturing system design. *Computers and Industrial Engineering*, 42, 457–470.

[74] Luong, L.H.S., 1993. A cellular similarity coefficient algorithm for the design of manufacturing cells. *International Journal of Production Research*, 31, 1757-1766.

[75] Malave, C.O., and Ramachandran, S., 1991. A neural network-based design of cellular manufacturing systems. *Journal of Intelligent Manufacturing*, 2, 305–314.

[76] Mansouri, S.A., Moattar-Hussein, S.M. and Newman, S.T., 2000. A review of the modern approaches to multi-criteria cell design. *International Journal of Production Research*, 38, 1201–1218.

[77] McAuley, J., 1972. Machine grouping for efficient production. *Production Engineering*, 51, 53–57.

[78] McCromick, W., Schweitzer, P., and White, T., 1972. Problem decomposition and data reorganization by a cluster technique. *Operations Research*, 20, 993–1009.

[79] Moon, C., and Kim, J., 1999. Genetic algorithm for maximizing the parts flow within cells in manufacturing cell design. *Computers and Industrial Engineering*, 36, 379-389.

[80] Mungwattana, A., 2000. Design of Cellular Manufacturing Systems for dynamic and uncertain production requirements with presence of routing flexibility. PhD thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University, Blackburg, VA.

[81] Nair, G. J. and Narendran, T. T., 1998. Case: A clustering algorithm for cell formation with sequence data. *International Journal of Production Research*, 36, 157–179.

[82] Nsakanda, A.L., Diaby, M. and Price, W.L., 2006. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research*, 171, 1051 – 1070.

[83] Ohta, H. and Nakamura, M., 2002. Cell formation with reduction in setup times. *Computers & Industrial Engineering*, 42, 317–327.

[84] Olorunniwo, F.O., 1997. A framework for measuring success of cellular manufacturing implementation. *International Journal of Production Research*, 35, 3043 – 3061.

[85] Pirlot, M., 1996. General local search methods. *European Journal of Operational Research*, 92, 493 – 511.

[86] Purcheck, G., 1974. A mathematical classification as a basis for the design of group technology production cells. *Production Engineer*, 54, 35–48.

[87] Purcheck, G., 1975. A linear programming method for the combinatorial grouping of an incomplete power set. *Journal of Cybernetics*, 5, 51–76.

[88] Rajagopalan, R. and Batra, J., 1975. Design of cellular production systems–a graph theoretic approach. *International Journal of Production Research*, 13, 56–68.

[89] Rajamani, D., Singh, N. and Aneja,Y. P., 1996. Design of cellular manufacturing systems. *International Journal of Production Research*, 34, 1917 - 1928.

[90] Rao, H.A, and Gu, P., 1992. Design of a cellular manufacturing system: a neural network approach. *International Journal of Systems Automation: Research and Applications,* 2, 407-424.

[91] Rao, H.A, and Gu, P., 1994. Expert self-organizing neural network for the design of cellular manufacturing systems. *Journal of Manufacturing Systems*, 13, 346-358.

[92] Raza, S.A., Akgunduz, A., and Chen, M.Y., 2006. A tabu search algorithm for solving economic lot scheduling problem. *Journal of Heuristics*, 12, 413 – 426.

[93] Ribeiro, J. F. F. and Pradin, B., 1993. A methodology for cellular manufacturing design. *International Journal of Production Research*, 31, 235–250.

[94] Ross, G.T., and Soland, R.M., 1977.A branch and bound algorithm for the generalized assignment problem. *Computers and Operations Research,* 8, 91-103.

[95] Safaei, N., Saidi-Mehrabad, M., Jabal-Ameli, M.S., 2007. A Hybrid Simulated Annealing for Solving an Extended Model of Dynamic Cellular Manufacturing System, *European Journal of Operational Research*, (In press).

[96] Salhi, S., 2002. Defining tabu list size and aspiration criterion with tabu search methods. *Computers and Operations Research*, 29, 67 - 89.

[97] Sankaran, S., 1990. Multiple objective decision making approach to cell formation: a goal programming model. *Mathematical Computer Modeling*, 13, 71-82.

[98] Selim, H.M., Askin, R.G. and Vakharia, A.J., 1998. Cell Formation in group technology: review, evaluation and direction for future research. *Computers and Industrial Engineering*, 34, 3-20.

[99] Shafer, S.M., and Rogers, D.F., 1991. A goal programming approach to cell formation problems. *Journal of Operations Management*, 10, 28-43.

[100] Shtub, A., 1989. Modeling group technology cell formation as a generalized assignment problem. *International Journal of Production Research*, 27, 775– 782.

[101] Singh, N. and Rajamani D., 1996. Cellular Manufacturing Systems: design, planning and control. *Chapman and Hall*,

[102] Singh, N., 1993. Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, 69, 281- 511.

[103] Solimanpur, M., Vrat, P. and Shankar, R., 2004. A multi-objective genetic algorithm approach to the design of cellular manufacturing systems. *International Journal of Production Research*, 42, 1419 – 1441.

[104] Song, S. and Hitomi, K., 1996. Integrating the production planning and cellular layout for flexible cellular manufacturing. *Production Planning and Control*, 7, 585-593.

[105] Spiliopoulos, K. and Sofianopoulou, S., 2003. Designing manufacturing cells: a staged approach and a tabu search heuristic. *International Journal of Production Research*, 41, 2531 – 2546.

[106] Srinivasan, G., Narendran, T., and Mahadevan, B., 1990. An assignment model for the part family problem in group technology. *International Journal of Production Research*, 28, 145–152.

[107] Srinivasan, G., and Narendran, T., 1991. GRAFICS – a nonhierarchical clustering algorithm for group technology. *International Journal of Production Research*, 29, 463-478.

[108] Steudel, H., and Ballakur, A., 1987. A dynamic programming based heuristic for machine grouping in manufacturing cell formation. *Computers and Industrial Engineering*, 12, 215-222.

[109] Su, C.T. and Hsu, C.M., 1998. Multi-objective machine-part cell formation through parallel simulated annealing. *International Journal of Production Research*, 36, 2185–2207.

[110] Suresh, N. C. and Kaparthi, S., 1994. Performance of fuzzy art neural network for group technology cell formation. *International Journal of Production Research*, 32, 1693–1713.

[111] Uddin, M.K. and Shanker, K., 2002. Grouping of parts and machines in presence of alternative process routes by genetic algorithm. *International Journal of Production Economics*, 76, 219–228.

[112] Vakharia, A.J., and Chang, Y., 1997. Cell formation in group technology: A combinatorial search approach. *International Journal of Production Research*, 35, 2025 – 2043.

[113] Vakharia, A. J. and Slim, H., 1994. Group Technology. In R.C. Dorf and A. Kusiak (Eds). *The Handbook of Design, Manufacturing, and Automation*. Wiley, New York, pp. 435–460.

[114] Venugopal, A., and Narendran, T. T., 1992a. A neural network approach for designing cellular manufacturing systems. *Advances in Modeling and Analysis*, 32, 13–26.

[115] Venugopal, V. and Narendran, T., 1992b. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers and Industrial Engineering*, 22, 469-480.

[116] Venugopal, V. and Narendran, T., 1992c. Cell formation in manufacturing Systems through simulated annealing: An experimental evaluation. *European Journal of Operational Research*, 63, 409 – 422.

[117] Viswanathan, S., 1996. A new approach for solving the p-median problem in group technology. *International Journal of Production Research*, 34, 2691–700.

[118] Vohra, T., Chen, D., Chang, J., and Chen, H., 1990. A network approach to cell formation in cellular manufacturing. *International Journal of Production Research*, 28, 2075–2084.

[119] Wang, J. and Roze, C., 1997. Formation of machine cells and part families in cellular manufacturing: an experimental study. *Journal of Operations Management*, 35, 1259–1286.

[120] Wemmerlov, U. and Hyer, N., 1986. Procedures for part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management,* 6, 125-145.

[121] Wemmerlov, U. and Hyer, N., 1987. Research issues in cellular manufacturing. *International Journal of Production Research*, 25, 413 – 431.

[122] Wemmerlov, U. and Hyer, N., 1989. Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research*, 27, 1511 – 1530.

[123] Wemmerlov, U. and Johnson, D.J., 1997. Cellular manufacturing at 46 user plants: implementation experiences and performance improvements *International Journal of Production Research*, 35, 29-49.

[124] Wu, N. and Salvendy, G., 1993. A modified network approach for the design of cellular manufacturing systems. *International Journal of Production Research*, 35(1), 29-49.

[125] Wu, H., Venugopal, R., and Barash, M., 1986. Design of a cellular manufacturing system: A syntactic pattern recognition approach. *Journal of Manufacturing Systems*, 5, 81–88.

[126] Xu, H., and Wang, H.P., 1989. Part family formation for GT applications based on fuzzy mathematics. *International Journal of Production Research*, 27, 2053-2062.

[127] Yin Y. and Yasuda K., 2002. Manufacturing cells design in consideration of various production factors. *International Journal of Production Research*, 40, 885-906.

[128] Zhao, C. and Wu, Z., 2000. A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, 38, 385–395.