The Development of Methods for the Design and Evolution of Reconfigurable Cellular Manufacturing Systems

Submitted By:

Mr. Jared Padayachee (BScEng, MScEng UKZN) – 203505399

Supervisor:

Prof. Glen Bright

January 2016

Submitted in fulfilment of the degree of Doctor of Philosophy in Mechanical Engineering.
College of Agriculture, Engineering and Science, University of KwaZulu-Natal.

DECLARATION - SUBMISSION

Supervisor

As the Candidate's Supervisor, I agree to the submission of this thesis.

Signed: …………………………… Date:…………………………

Prof. Glen Bright

Student

I declare this thesis has not previously been submitted for any degree in this university, or any other university, and it is my (Jared Padayachee) original work.

Signed: …………………………… Date: …………………………

Mr. Jared Padayachee

DECLARATION - PLAGIARISM

I, Jared Padayachee (203505399), declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
    a. Their words have been re-written but the general information attributed to them has been referenced
    b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed: …………………………… Date: …………………………

Mr. Jared Padayachee

DECLARATION - PUBLICATIONS

Details of contribution to publications that form part and/or include research presented in this thesis (include publications in preparation, submitted, *in press* and published and give details of the contributions of each author to the experimental work and writing of each publication)

Published

In the publications listed, Mr. J. Padayachee was the developer of the work and lead author:

J. Padayachee, G. Bright*;* A Theoretical Framework for a Manufacturing Planning System for RMSs; *RobMech 2011: 4th Mechatronics and Robotics Symposium*, 23 - 25 November 2011, Pretoria, South Africa

J. Padayachee, G. Bright; Automated Design of Reconfigurable Manufacturing Systems: A Framework based on a Genetic Algorithm and Discrete System Simulation; *ICINCO 2012: International Conference on Informatics in Control, Automation and Robotics*; 28 – 31 July 2012; Rome; Italy

J. Padayachee, G. Bright; The Design of Reconfigurable Manufacturing Systems for Product Mass Customisation; *International Conference on Competitive Manufacturing (COMA '13)*; 30 January – 01 February 2013; Stellenbosch; South Africa

J. Padayachee, G. Bright; Synthesis of Evolving Cells for Reconfigurable Manufacturing Systems; *27th International Conference on CADCAM, Robotics and the Factories of the Future*; 22-24 July 2014; London England

Journal Articles in Preparation

J. Padayachee, G. Bright; A Simultaneous Fuzzy Clustering Heuristic for Cell and Part Family Formation in Cellular Systems.

J. Padayachee, G. Bright; A Cost Minimization Reconfiguration Planning Model for CMS Based on a Factory Configuration Template Concept.

Signed: ……………………………   Date: …………………………

Mr. Jared Padayachee

Acknowledgements

I would like to thank the Lord Jesus Christ for the ability he has given me to complete this research.

I would like to thank my supervisor, Prof. Glen Bright for the opportunity to have completed this research under his supervision.

I would like to acknowledge the National Research Foundation (NRF) for the award of a sabbatical grant for lecturing relief to complete this research.

I would like to acknowledge the Technology Innovation Agency (TIA) for financing this research project under the Reconfigurable Manufacturing Systems grant.

I would like to thank my colleague Dr Anthony Walker for the many conversations that have helped refine the ideas and concepts that were developed in this research.

I would like to thank my wife Anneline and my parents for their continuous support and prayer during the completion of this thesis.

Abstract

The concept of reconfigurable manufacturing is presently being researched due to the need for production systems that are able to economically respond to changes in markets and the rapid introduction of new products. Cellular Manufacturing Systems (CMS) are a central concept in just-in-time and lean manufacturing. Although CMS are able to provide a strategic operating advantage, machine cell clusters do not remain optimal over an extended period of time.

The concept of a Dynamic CMS (DCMS) has received attention in recent years; a DCMS is a system where the layout of machines change in order to improve the responsiveness of CMS to changing production requirements. A deficiency in existing DCMS methods is that reconfiguration plans are generated without the consideration of an initial design of the factory floor space for future change. This research distinguishes Reconfigurable CMS (RCMS) from DCMS, as a system that is designed at the outset for changes to system layout and cell configurations. The concept of a Factory Configuration Template (FCT) is proposed in this research; the FCT is a design of the factory floor space to ensure the feasible implementation of reconfiguration plans generated by mathematical models. A nine step method for FCT design is presented that uses a Simultaneous Fuzzy Clustering Heuristic to develop manufacturing cells and part families. A Tabu Search algorithm was develop to generate the optimal arrangement of machine sites in cells.

Three multi-period machine assignment models were developed that determine reconfiguration plans based on changing product demand and the introduction of new products. The models that were developed included two integer linear programs that determine the distribution of machine resources among cells over multiple periods. A quadratic zero-one programming model was developed that distributes machines among available sites in cells to promote unidirectional part flow. The results show that RCMS is able to provide a more economical solution than traditional CMS with the added advantage of improved part flow in the system.

Keywords: Dynamic Cellular Manufacturing, Reconfigurable Manufacturing, Group Technology, Reconfiguration Planning

This page was intentionally left blank.

Table of Contents

List of Acronyms and Abbreviations

ALC	Average Linkage Clustering
ART	Adaptive Resonance Theory
BFO	Bacteria Foraging Optimization
CAD	Computer Aided Design
CIM	Computer Integrated Manufacturing
CLC	Complete Linkage Clustering
CMS	Cellular manufacturing System
CS	Cluster of Sites
DFS	Depth First Search
DMS	Dedicated Manufacturing System
EE	Exceptional Element
FCMdd	Fuzzy C-Medoids
FCT	Factory Configuration Template
FFA	Factory Flow Analysis
FMS	Flexible Manufacturing System
GA	Genetic Algorithm; Group Analysis
GT	Group Technology
ILP	Integer Linear Program
LA	Line Analysis
LB	Lower Bound
LCCA	Linear Cell Clustering Algorithm
LP	Linear Program
MC	Machine Cell
MIP	Mixed Integer Program
MODROC	Modified Rank Order Clustering
MU	Machine Unit
PF	Part Family
PFA	Production Flow Analysis
RCMS	Reconfigurable Cellular Manufacturing System
RMS	Reconfigurable Manufacturing System
ROC	Rank Order Clustering
SA	Simulated Annealing
SFC	Simultaneous Fuzzy Clustering

SLC	Single Linkage Clustering
TA	Tooling Analysis
TS	Tabu Search
TSP	Travelling Salesman Problem
UB	Upper Bound
WIP	Work-In-Process

Nomenclature

Latin alphabet

a	Number of positive matches (1's) when comparing two binary vectors.
a_k	A zero-one variable indicating that the machine in the first site has been gained by reconfiguration in period *k*.
a_{ij}	Element of a zero-one machine component incidence matrix.
a_{ik}	The number of parts in cluster *i* that require processing on machine *k*.
$a(k)$	The average dissimilarity of member *k* with respect to other members of the same cluster.
A	A norm-inducing matrix; a $C \times N$ affinity matrix; a coefficient matrix in a system of linear inequality constraints.
Aeq	A coefficient matrix in a system of linear equality constraints.
b	Instances of 1's in the first binary vector that are not present in a second vector; vector of right hand side values in a system of linear inequality constraints.
beq	Vector of right hand side values in a system of linear equality constraints.
$b(k)$	The lowest dissimilarity between the element *k* and another cluster.
b_k	A zero-one variable indicating that the machine in the first site has been removed by reconfiguration in period *k*.
c	Instances of 1's in the second binary vector that are not present in the first vector.
c_{ij}	Flow volume between facility *i* and facility *j*.
c_k	The cost per part per intra-cell backtracking movement in period *k*.
C	Number of clusters; a vector of objective function coefficients corresponding to linear terms.
C_{ijk}	Cost of an hour of deficit capacity in cell j due to an inadequate number of machine type *i* in period *k*.
d	Number of negative matches (0's) when comparing to binary vectors.
d_{ij}	Distance between the centre of facility *i* and facility *j*.
D	A dissimilarity symmetric matrix.
D_{ik}	Distance or dissimilarity between element *i* and cluster (or cluster prototype) *k*.
e	Total number of 1's in a machine-component incidence matrix.
e_1	Number of 1's in a block diagonalised matrix.
e_v	Number of voids in a block diagonalised matrix.
e_o	Number of exceptional elements in off-diagonal blocks.

E_{ik}	Hours of excess capacity of machine type i available in period k.
f	A vector of objective function coefficients.
F	Total hours of surplus and deficit capacity in cells for all periods of interest.
F_k	Total hours of surplus and deficit capacity in cells in period k.
g_k	Number of clusters among which the maximum membership of element k is shared.
H_{ik}	Hours of processing time required on machine i in period k.
H_{ijk}	Hours of processing time required on machine i in cell j in period k.
I	Set of machines.
J	Set of Cells.
K	Set of periods.
L	Relative part load; set of machine locations.
m	Fuzziness parameter ($m>1$).
m_i	Market potential of product generation i
M	Number of crisp clusters that may be derived from a fuzzy partitioning matrix; a constant in linear conditional constraint; number of machines.
n	Number of facilities.
N	Total parts; number of data elements.
N_{ik}	Machines of type i are that are required/available in period k.
N_{ijk}	Machines of type i are that are required/available to cell j in period k.
N_k	The maximum number of machines of a specific type that have been made available to a cell in period k.
o	Total number of 0's in matrix.
p	Innovation coefficient.
p_{ijk}	The number of machines of type i added to cell j in period k.
p_k	A zero-one variable indicating that the machine in the second site has been gained by reconfiguration in period k.
P	Set of parts.
q	Weighting in the range [0, 1]; a medoid index; imitation coefficient.
q_{ijk}	Number of machines of type i removed from cell j in period k.
Q	Q vector of objective function coefficients corresponding to quadratic terms.
Q_p	Ordered set of matched processes and machine locations.
r_k	Cost of a cell gaining/losing a machine in period k; cost of an intra-cell machine relocation in period k.
R_k	Budget available for reconfiguration (machine re-layout) in period k.
$s(k)$	Silhouette of a data element k $(-1 \le s(k) \le 1)$.
S_{ij}	Similarity between machine/part i and cell/family j.

$S_i(t)$	Total sales pf product generation i at time t
T	Total number of periods
u_{ij}	Instances of part flow between machine i and j.
u_{ijk}	Hours of excess capacity with respect to machine type i, in cell j in period k.
U	$C \times N$ membership matrix.
$U_{i\alpha j\beta}$	Instances of flow from machines type i in location α to machine type j in location β.
U_k	Total excess capacity in cells in period k.
v_i	Prototype of cluster i.
v_{ijk}	Hours of deficit capacity with respect to machine type i, in cell j in period k.
V	A vector of cluster prototypes.
V_k	Total hours of deficit capacity in cells in period k.
W_k	Total system operating hours in period k.
$x_{i\alpha}$	A zero-one variable indicating the assignment of machine i to location α.
$x_{i\beta}$	A zero-one variable indicating the assignment of machine j to location β.
x_{ijk}	Number of machines of type i allocated to cell j in time period k.
x_k	A zero-one variable indicating the assignment of the machine to the first site.
X	Number of additional machines/sites allowed in series.
X_{ij}	Number of sites available for machine i in cell j.
X_{ijk}	Zero-one variable indicating that operations on part k are performed on machine i and j.
y_{ijk}	A zero-one variable that is used to enforce conditional constraints in mathematical programs.
y_k	A zero-one variable indicating the assignment of the machine in the second site.
Y_{ik}	Zero-one variable indicating that an operation on part k takes place on machine i.
Z_{jk}	Zero-one variable indicating that an operation on part k takes place on machine j.

Greek alphabet

α_k	The volume of parts that will perform an intra-cell backtracking movement if a machine is not available in the first site in period k.
β	Fuzzy membership model constant.
β_k	The volume of parts that will perform an intra-cell backtracking movement if a machine is not available in the second site in period k.
γ	Weighted average efficacy.
δ	Instances of part movements from machine i to j.
ζ	Maximum copies of each machine type that may be added to the system.
η	Group Efficiency.

μ_{ik}	Membership value of element *i* with respect to cluster *k*.
$\tilde{\mu}_{ik}$	Crisp membership value of element *i* with respect to cluster *k*.
ξ_g	Grouping Measure.
ξ_u	Machine utilization rate.
ξ_m	Inter-cell part movement rate.
τ	Group Efficacy.
τ_i	Time to market of i^{th} product generation

List of Figures

List of Tables

This page was intentionally left blank.

1 Introduction

1.1 Manufacturing Challenges

The motivation for this research originates in the need to address manufacturing challenges that have emerged as a result of global economic competition in recent years. Researchers such as Koren et al [1] have identified the following competitive strategies in modern manufacturing to achieve the objectives of profitability and increased market share:

(i) the frequent introduction of new products with improved technology;
(ii) the offering of increased product variety to the consumer;
(iii) the production of high quality goods at competitive prices;
(iv) a rapid response to market trends, demand and consumer needs.

The manufacturing of products, while employing these competitive strategies presents a challenge to production planners and engineers. High product variety, technology innovation and unstable demand cause manufacturing systems to operate under rapidly changing conditions [2]. Challenges that arise from the implementation of competitive strategies include:

(i) frequent adaptation of manufacturing processes to new products;
(ii) large fluctuations in product demand and mix;
(iii) frequent changes in process technology;
(iv) the manufacturing of multiple product types on the same equipment.

The cost effective initialization of manufacturing systems that facilitate the competitive strategies, has initiated the revision of manufacturing paradigms; this has been the inspiration for much research in recent years. This study addresses the Cellular Manufacturing Systems (CMS) paradigm in the context of coping with the introduction of new products, changes to part demand and mix, the phasing out of older product models and the practicality of addressing these challenges via system reconfiguration.

1.2 Introduction to Cellular Manufacturing Systems

CMS are a manufacturing system design paradigm that emerged in the 1970's and gained popularity in the 1980's [3]. Cellular manufacturing is a central concept in achieving lean manufacturing and just-in-time manufacturing [4]. Lean manufacturing aims to minimize waste and underutilization of manufacturing resources and just-in-time manufacturing is an element within the context of lean manufacturing that deals with minimising Work-In-Process (WIP) and inventory levels. Both lean and just-in-time manufacturing find their roots in the development of the Toyota Production System [5]. The CMS is therefore an important and established manufacturing paradigm.

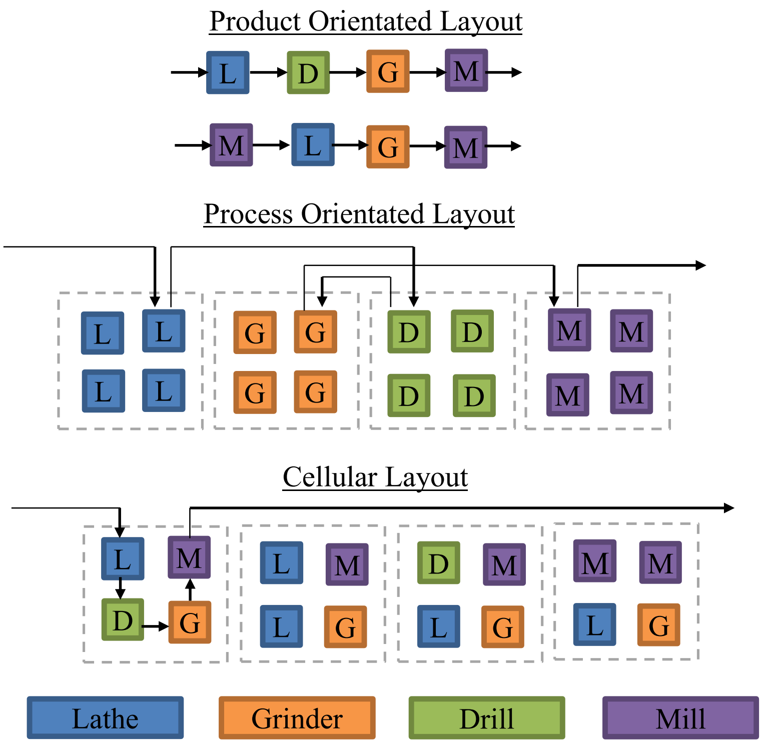

Figure 1.1: Types of manufacturing facility layouts. (adapted from [6])

Cellular Manufacturing Systems are higher variety, medium volume production systems, that are capable of greater product variety than Dedicated Manufacturing Systems (DMS) and higher production rates than job shops [7]. Figure 1.1 illustrates three common types of manufacturing system layouts. The product orientated layout is best suited to the manufacturing of a dedicated product type and is the least flexible in accommodating changes to products due to rigid flow paths. The process orientated layout, which is also known as a functional layout, is a typical job-shop layout. It is capable of high part routing flexibility, however, this flexibility is accompanied by complex part flow.

CMS exploit the concept of Group Technology (GT) to cluster parts into families and machines into cells. Figure 1.1 illustrates the concept of clustering machines into manufacturing cells and usually each cell is responsible for the production of a single part family. The objective of creating part families and machine cells is to exploit similarities in the manufacturing requirements of components and products. The benefits of CMS include reduced part set-up times on machines, better organization of tooling, lower work-in-process levels, reduced expenditure on material handling, less chaotic work flow, improved product quality, lower complexity scheduling and hassle free system management [3, 8]. Literature on CMS also contains reports of improved worker morale and productivity after the transformation of a system to a cellular type layout [6].

The disadvantages of CMS include the lack of flexibility to accommodate changes to products, changes to the mix of parts being manufactured, the underutilization of machinery and the cost incurred by having to re-layout cells [6]. Other disadvantages, from a theoretical perspective, include the lack of a single best method for cell formation and the planning of the re-layout of cells when cluster configurations cease to be optimal. Note that re-layout refers to the addition/removal and swapping of machines between cells to create new machine clusters. The terms re-layout and reconfiguration are used synonymously in this research. Re-layout results in modified cells with the appropriate manufacturing functions and production capacity to cope with change.

The re-layout of a CMS is key to exploiting the benefits of a cellular layout and maintaining profitability and competitiveness over an extended period of time. Practical cases of re-layout in literature were found to be conducted in a reactive manner; after symptoms of poor system performance were prevalent. The re-layout of a manufacturing system is an endeavour that bears significant financial risk. If the system is not designed at the outset, with the vision of adapting cellular configurations in the future, the cost of reconfiguration may be exorbitant. Other types of risk that will manifest due to the lack of an initial reconfigurable design include the risk of an extended system downtime and slow system ramp-up time; both of which equate to a loss of productivity and revenue. In addition to the various types of risk, a system that is not designed for re-layout may exhibit irreversibility; that is the installation of structures in the system that physically prohibit the movement of some machines and other types of equipment to different cells. In these instances it may be impossible to implement the most optimal solution that may be proposed by a theoretical analysis.

1.3 Necessity for the Study

The Reconfigurable Manufacturing Systems (RMS) paradigm was proposed by Koren et al [9, 10], with machine and system level configurability as the core mechanism to meet the demands of modern manufacturing. RMS's were conceptualized through an international collaboration between researchers at universities that included the University of Stuttgart, Politechnico di Milano, Katholieke Universiteit Leuven and the University of Michigan. The definition of a RMS is as follows [1, 9]:

"A Reconfigurable Manufacturing System (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or regulatory requirements."

The concept of reconfigurability, as defined here, is applicable to the design of Cellular Manufacturing Systems. The key characteristic in distinguishing a system as RMS is the initial design of the system for change, which includes changes in the facility layout. Other important characteristics include the initial design of the system to be convertible and customisable [11]. Convertibility implies a system design for a rapid conversion of manufacturing functions and production capacity in response to changes within a product/part family. Customization implies system design and subsequent reconfiguration for exact functionality and capacity in response to changes in part demand and mix [1, 12].

The re-layout or reconfiguration of a CMS does not imply that the system is a type of RMS. The re-layout of CMS has been studied under the topic of Dynamic Cellular Manufacturing Systems (DCMS) since the 1990's [13]. DCMS exploits the reorganization of machinery to alter cell configurations, however, the literature review has shown that little consideration is given to the initial design of the system for reconfigurability and the practicality of implementing layout changes as proposed by optimization models.

In the literature that has been found, some research has been conducted on the topic of Reconfigurable Cellular Manufacturing Systems (RCMS), however, researchers have tended to use the terms RCMS and DCMS interchangeably, with no significant distinction between the concepts. Likewise, the works that have been published on the topic of RCMS have granted little or no consideration to the initial system design for future re-layout.

This research involves the development of design methods for RCMS. Based on the inconsistent use of the terminology in the literature, the following definition is meant to bring a consistent understanding to the use of the term RCMS as it appears in this study:

RCMS: is a system that is designed at the outset to be reconfigurable and reconfigurability is defined in this context to refer to the extent to which the system has been designed to accommodate future layout changes.

The methods developed in this research adhered to the fundamental principle that for a CMS to be considered reconfigurable, the re-layout planning method must make provision for future re-layout as an integral part of the initial design. An appropriately designed RCMS should impart the flexibility to a system to add, remove or redistribute manufacturing equipment among cells without incurring inhibiting expenses and averting irreversibility in areas of the system structure. Reconfigurable CMSs, based on an initial design for re-layout, are envisioned to combine the benefits of traditional CMS with economical adjustments of production capability to provide a suitable solution to the challenges and demands of modern manufacturing.

1.4 Aim and Objectives

The aim of this research was to review existing CMS and DCMS design and reconfiguration planning techniques and formulate a methodology for the design and subsequent evolution of the layout of a Reconfigurable Cellular Manufacturing System (RCMS). This was to establish if an appropriately designed RCMS can respond to the introduction of new products and changing product demand in a manner that is economical and advantageous from a production management perspective.

The objectives of this research were to:

(i) Research and establish the deficiencies in existing CMS and DCMS design and reconfiguration planning methods.

(ii) Research and develop an original method of imparting reconfigurability to a CMS at the design stage.

(iii) Research and develop an original cell formation method that supports the initial design of the system for reconfigurability.

(iv) Research and develop an original line analysis method that supports the initial design of a RCMS for reconfigurability.

(v) Research and develop an original multi-period reconfiguration planning method for the re-layout of a RCMS within the provisions for reconfiguration created in the system at the design stage.

(vi) Investigate the advantages of RCMS over CMS with fixed layouts through the application of the design and reconfiguration planning methods that have been developed.

1.5 Contributions of this Study

This research provides an original design method and reconfiguration planning methods for RCMS. The design method imparts reconfigurability to the manufacturing facility at the outset. The methods that have been developed include:

(i) the development of the concept of a Factory Configuration Template (FCT) that is to impart reconfigurability to RCMS at the design stage;

(ii) the development of an original GT method, based on the concept of fuzzy clustering for cell formation, that supports future re-layout;

(iii) the development of an original line analysis method that pre-empts possible intra-cell rearrangements of machines to support unidirectional part flow.;

(iv) the development of original models for multi-period planning of reconfigurations, based on integer programming methods, that propose both inter-cell and intra-cell re-layout plans for machines within the provisions for reconfiguration made on the FCT.

The individual techniques are consolidated into a design and reconfiguration planning procedure that is demonstrated by application to the design of a RCMS. The RCMS responds to changing product demand, the introduction of new products and the termination of old products in a manner that is economical and reduces the complexity of part flow in the system.

1.6 Outline of Thesis

Chapter 1 introduced the reader to the concept of a CMS, the motivation for the research and the resulting scientific contributions. The objectives of the research were also presented. Chapter 2 reviews and analyses the inadequacies of classical and recent design methods for CMS, DCMS and RCMS. The methods reviewed include group analysis methods and line analysis methods. Chapter 3 develops the concept of a Factory Configuration Template. The re-layout of CMSs within the flexibility provided by the template is a distinguishing characteristic between this research and other methods found in the literature. Chapter 4 presents the formation of manufacturing cells and families by the process of fuzzy clustering. The Fuzzy C-Medoids model, together a Simultaneous Fuzzy Clustering (SFC) heuristic, are used to identify the flexibility cells should possess in terms of their ability to host machines of different types over the planning horizon. This information is used in the creation of a suitable FCT. Chapter 5 presents a line analysis technique that is independent of the volume of specific product demand. The technique is based on a Tabu Search algorithm with the ability to propose duplicate sites for machines in cells. This information is also essential for the creation of a suitable FCT. Chapter 6 presents three models for the multi-period re-layout of a CMS. The models are based on integer programming techniques. Chapter 7 demonstrates the design of a RCMS through methods developed in this research. The study demonstrates that an economic and production management advantage can be gained be re-layout of the manufacturing system. Chapter 8 discusses the performance of the system design and reconfiguration planning methods and areas for improvement and future research are identified. The implications of a FCT for RCMS are also discussed. Chapter 9 concludes this thesis with a discussion of the highlights of this research and a summary of the problems that will require future research.

1.7 Chapter Summary

This chapter introduced the reader to the challenges in modern manufacturing that have provided the motivation for this research. The scientific contributions of the research toward the design and planning of reconfigurations in cellular systems we discussed. The research objectives were presented, that contribute to the aim of formulating a methodology for the design and subsequent evolution of a RCMS.

2 A Review of Classical and Recent Design Methods for CMS

2.1 Chapter Introduction

A literature review on the design methods for Cellular Manufacturing System is presented in this chapter. Two problems that are of particular importance in the design of CMS are the Group Analysis and Line Analysis problems. Methods for Group Analysis and Line Analysis were researched for both static and dynamic manufacturing requirements. The variety of approaches that have been developed to solve these problems has been documented here. The deficiencies in existing methods for the design of a viable reconfigurable CMS are discussed at the end of the chapter.

2.2 Elements of CMS Design

The early works on the design of CMS were conducted by Burbidge [14-16], who defined four stages for the design of cellular systems:

1. Factory Flow Analysis (FFA): examines flows between shops and buildings. This eliminates unnecessary interdepartmental routes [14]. Backtracking is reduced by the redeployment of machinery.
2. Group Analysis (GA): analyses process sequences of parts being produced in a particular shop/building in order to generate manufacturing cells. Each cell will have a complete set of machinery for manufacturing associated parts, inter-cellular material movements are allowed to/from vendors.
3. Line Analysis (LA): analyses intra-cellular material routings in order to generate a layout that maximises unidirectional flow.
4. Tooling Analysis (TA): Material size, shape, tooling and fixturing for parts are analysed. TA is necessary for the development of schedules such that there is maximum synergy between the operation sequence, tools and set-ups used by successive product batches. The objective is to reduce changeover delays and increase machine availability to alleviate high WIP and bottlenecks.

The combination of the stages of analysis was collectively termed Production Flow Analysis (PFA). The GA and LA stages are concerned with the layout of manufacturing cells within a single building and are topics that are relevant to the development of a RCMS.

2.3 Fundamentals of Group Analysis

Group Analysis is concerned with the generation of machine cells and part families. The science and practice of grouping machines and parts is commonly known as Group Technology (GT). Since the early works by Burbidge [14-16] a variety of methods have been proposed for the development of machine cells and part families. Within the variety of methods that are available, there are three elementary strategies [17]:

1. Simultaneous machine cell and part family formation.
2. Machine cells are formed first and parts are grouped into families according to the cell that is able to provide common manufacturing services.
3. Part families are formed first and machines are grouped into cells according to the family that has greatest need of those processes.

Further distinction may be found between the various methods based on the criteria they consider such as part demeaned, balancing workloads between cells, tooling requirements and cell size. The majority of methods available begin with the analysis of a machine-component incidence matrix. Consider the example from King [18], shown in Table 2.1. Here row indices 1-5 signify machine types and column indices 1-6 signify part/component types. The table captures the incidence data of parts on machines and the table is meant to be treated as a matrix with *i* and *j* as the respective row and column indices. Here $a_{ij} = 1$ indicates that the part corresponding to column index *j* requires the manufacturing services of the machine corresponding to index *i*; $a_{ij} = 0$ indicates no coupling between the corresponding part and machine.

Table 2.1: Initial Machine - Component Incidence Matrix

		PARTS					
		1	**2**	**3**	**4**	**5**	**6**
MACHINES	**1**	0	0	1	0	1	0
	2	0	1	1	0	0	0
	3	1	0	0	1	1	0
	4	0	1	1	0	1	0
	5	1	1	0	1	0	1

At the time of formulating the initial machine-component incidence matrix, process plans for parts have already been created. This enables the machines that are able to provide these processes to be identified. The strategic issue of machine flexibility must also be addressed prior to formulating the incidence matrix. Wahab et al [19] emphasise the importance of machine flexibility in maintaining the future competitiveness of a manufacturing system. They also present different measures of machine flexibility that may assist decision in making. Machines that are available on the market must be evaluated and selected based on their ability to provide the processing operation of parts that are to be manufactured both in the present and the future. Table 2.2 is a rearranged version of Table 2.1; it shows two block structures along the diagonal of the matrix. These block structures identify individual machine cells and part families. Machine Cell (*MC1*) contains machines 3 and 5 and Part Family (*PF1*) contains parts 1, 4 and 6; i.e.

MC1 = {3, 5} and *PF1 = {1, 4, 6}*. Likewise, *MC2 = {1, 2, 4}* while the corresponding part family is *PF2 = {2, 3, 5}*.

Table 2.2: Modified Machine - Component Incidence Matrix with Block Structures

		PARTS					
		1	**2**	**3**	**4**	**5**	**6**
MACHINES	**1**	1	1	0	0	0	1
	2	1	1	1	0	1	0
	3	0	0	0	1	0	1
	4	0	0	0	1	1	0
	5	0	0	0	1	1	1

An examination of the modified table reveals that parts 5 and 6 are dependent on machines not within their cell; these parts are referred to as *exceptional parts*. The corresponding off-diagonal elements a_{16} and a_{25} are referred to as *exceptional elements*. Under the existing machine cell and part family configurations, machines 1 and 2 will process parts that are not within the corresponding family. These machines are referred to as *bottleneck machines*. In order for part 5 and 6 to access machines 1 and 2, these parts will have to move between the first cell and the second cell; this type of material flow is referred to as *inter-cell* part flow.

Table 2.2 shows that there are elements $a_{ij} = 0$ within the block diagonal structures, for example $a_{13} = 0$. Elements $a_{ij} =0$ within the block diagonal structures are referred to as *voids*. Voids are an indicator of machine underutilization by the corresponding part family. The number of voids and exceptional elements in a solution are widely used in metrics for determining the performance of a cell formation solution [20]. The minimization of voids and exceptional elements have also been used as objectives in optimization algorithms such as the multi-objective genetic algorithm developed by Arkat et al [21].

The objective of the cell and part family formation process is to create maximum independence among cells by eliminating coupling due to parts requiring services from multiple cells. When the members of a part family require the services of multiple cells, this creates inter-cell material flow and corresponding inter-cell material handling costs. Other complications that arise from inter-cell part flow include more a complex scheduling task, the requirement for additional buffering, increased work-in-process levels and increased difficulty in quality control. When each cell functions with autonomy by exclusively producing it's corresponding part family, the result is hassle free system management and control [22]. In order to improve the autonomy of cells, duplicate machines may be purchased in order to eliminate exceptional elements. Another objective of the Group Analysis process is to determine the cell and part family configurations that will require the least number of duplicate machines cells [23].

There are numerous factors that are taken into consideration when designing machine cells. The term *cell size* is often used to refer to the number of machines in a cellular cluster. Cell size and the number of cells generally share inversely proportional relation. An increase in the number of cells is usually accompanied by a decrease in cell size. Cell size and the number of cells may be dictated by parts that are common to a set of machines, however, this is not without consideration of available floor space and operator requirements. A large number of small cells may require a larger work force. A smaller number of larger cells may benefit from fewer operators tending to multiple machines; however the control of a large cell may be challenging.

2.4 The Group Analysis Problem and Solution Approaches

The classical group analysis problem involves the formation of machine cells and part families over a single period planning horizon based on machine-component incidence data such as that presented in Table 2.1. The problem of cell and part family formation from a zero-one incidence matrix is a NP-hard problem [24]. Another important class of GT problems are those that consider alternative process plans (alternative process routes for parts). These problems are referred to as generalized GT problems [25].

Burbidge [14-16] proposed a heuristic for group analysis that evaluates the usage frequency of machines by examining the machine-component incidence matrix. The heuristic then forms modular clusters around significant machines. Modules are split in cases where there are distinct sub-clusters. Modules that share a large number of common machines may be merged until the desired number of cells is obtained. Bottleneck machines may be redistributed among the modules at this point to eliminate exceptional parts. The final set of machine cells are verified by performing a load check to determine if the cells are able to meet the part demand. The reader is referred to [16] for a complete description of the PFA methodology and the charts and heuristics that are used to complete the process. Hameri [26] reported that although the heuristic proposed by Burbidge is a well-documented and proven technique for simplifying complex material flow networks, the technique is not widely used.

A variety of GT methods have been developed to produce improved results or consider additional factors with respect to the group analysis problem. Methods available include similarity coefficient generation and linkage analysis, part coding analysis such as the Opitz part classification system [27], array manipulation techniques, clique partitioning approaches, mathematical programming approaches, heuristic approaches such as genetic algorithms and simulated annealing, and artificial intelligence approaches such as expert systems and neural networks. A selection of literature based on these approaches will be discussed briefly.

2.4.1 Similarity Coefficient Approaches

McAuley [28] was the first researcher to propose a cell formation method based on similarity coefficients, particularly the Jaccard Similarity coefficient. The similarity S_{ij}, between two machines was based on the number of common parts that visited each machine and is given by:

$$S_{ij} = \frac{\sum_{k=1}^{N} X_{ijk}}{\sum_{k=1}^{N}\left(Y_{ik} + Z_{jk} + X_{ijk}\right)} \tag{1}$$

Where: $X_{ijk} = 1$ if an operation on part k is performed on both machines i and j.

$X_{ijk} = 0$ otherwise.

$Y_{ik} = 1$ if an operation on part k is performed on machine i. $Y_{ik} = 0$ otherwise.

$Z_{jk} = 1$ if an operation on part k is performed on machine j. $Z_{ik} = 0$ otherwise

Jaccard Similarity was computed with the incidence data obtained from the machine-component incidence matrix. The method then proceeded with Single Linkage Cluster (SLC) analysis, i.e. a linkage via a sufficiently high similarity with any member of an existing cluster warrants the admission of the additional machine into that cluster. The method first clusters machines with the highest calculated similarity and successively lowers the level of admission into a cell. The number of individual clusters decreases to produce less cells as the similarity criteria for clustering is reduced. Parts are then assigned to machine cells based on the cell that is able to provide most of the manufacturing operations. A weakness of the method is the use of SLC, where the machine being introduced into the cluster may have a strong similarity to one member of the cluster yet may still have a relatively low similarity to other members. Seifoddini [45] applied Average Linkage Cluster (ALC) analysis to the cell formation problem and found it to overcome the deficiencies found in SLC. The Jaccard similarity co-efficient, first used by McAuley [28] has been found to have the following limitations [29]:

(i) there is no precise identification of the number of inter-cell movements made by parts;
(ii) the similarity is only between a pair of machines, it does not determine the similarity between a machine and a cluster;
(iii) it does not give significance to a part that is not processed by machine pairs in a cluster.

Researchers have therefore endeavoured to propose alternative similarity coefficients and dissimilarity coefficients. Ghosh and Dan [30] presented a hybrid clustering method that exploited Pearson's sample correlation coefficient as a measure of similarity between two parts. Seifoddini and Djassemi [31] developed a production data similarity co-efficient that considered part production volumes, combined with single linkage clustering. In comparison to the Jaccard coefficient, it produced lower inter-cell and intra-cell material handling costs decreased in six out of ten cases. Gupta and Seifoddini [32] proposed an alternative similarity coefficient combined with Complete Linkage Clustering (CLC) Analysis for the purpose of cell formation. The coefficient takes into consideration part routings, production volumes and unit operation time for each operation performed on a part.

Wei and Kern [33] proposed the commonality score as an alternative to the Jaccard similarity coefficient. The authors proposed an algorithm called the Linear Cell Clustering Algorithm (LCCA) for cell formation using the commonality score. Chow and Hawaleshka [34] proposed an improved algorithm based on the concept of a Machine Unit (MU). The results were compared to an application of the methods proposed by Wei and Kern [33] and Sefoddini [35]. In both instances the algorithm proposed by Chow and Hawaleshka produced machine clusters with lower inter-cellular movement of parts.

2.4.2 Array Manipulation Algorithms

Array based cell formation algorithms rearrange the rows and columns of a machine-component matrix to obtain block diagonal structures; these algorithms simultaneously identify machine cells and part families. The Rank Order Clustering (ROC) algorithm was the first such algorithm to be applied to the problem; originally proposed by King [18, 36]. King and Nakornchai [37] later developed an extended version of the algorithm called ROC2, which includes bottleneck relaxation and exploits the properties of sparse matrices to address the issues of storage and matrix manipulation in large problems. ROC has been observed to be a fast converging algorithm, however Chandrasekharan and Rajagopalan [22] proved that the ROC algorithm does not always identify the block diagonal structures in a matrix. Chandrasekharan and Rajagopalan [22] developed the MODified Rank Order Clustering (MODROC) algorithm to overcome these deficiencies. The authors demonstrated the application of the MODROC algorithm on an 8 machine 20 part matrix that was otherwise not capable of being block diagonalised in a suitable manner by an application of the ROC. Other matrix based clustering algorithms are the Bond Energy Algorithm presented by McCormick et al [38] and the Matching Algorithm presented by Bhat and Haupt [39]. The work presented by these authors was not with specific intent on cell formation but rather as general matrix based clustering techniques.

2.4.3 Non-Hierarchical Clustering

Chandrasekharan and Rajagopalan [40] proposed a non-hierarchical clustering method for cell and part family formation. The authors applied their method to the data set from King and Nakornchai [37]. The authors showed improved grouping by their method when compared to an extension of the Rank Order Cluster Algorithm. Chandrasekharan and Rajagopalan [41] presented an improved and extended version of the method called ZODIAC.

Srinivasan and Narendran [42] proposed the use of a non-hierarchical clustering algorithm called GRAFICS for determining machine cells and part families. The authors compared GRAFICS and ZODIAC on 38 test problems. GRAFICS showed higher group efficiency for 14 problems and higher group efficacy for 16 problems. ZODIAC produced higher group efficiency for 6 problems and higher group efficacy for 4 problems. In other cases both methods produced identical solutions.

2.4.4 Clique Partitioning for Cell Formation

Rajagopalan and Batra [43] proposed a graph-theoretic approach to the cell formation problem. They proposed that the machines be modelled as vertices. The relationship between two machines was modelled as edges. This relationship is created by the parts that visit a pair of machines and was formally modelled by the Jaccard similarity coefficient. The cell formation process begins by identifying cliques in the graph. In graph theory a clique is a sub-graph, identified as a subset of vertices with maximal connectivity between them. The Bron-Kerbosch algorithm was used for clique identification. Cliques are then merged and the graph is partitioned so as to minimize inter-cell movement of parts. The Kernighan-Lin approach was used to partition the graph with limitation on cell size (i.e. number of vertices in a partition). No limit on the number of cells was imposed. Parts are allocated to formed cells such that inter-cell movement is minimized. The authors demonstrated the usefulness and feasibility of this approach although no comparative study was conducted against other methods.

Wang et al [44] identified a computation problem in the work presented by Rajagopalan and Batra [43], in that for moderate to large size graphs the clique partitioning optimization model explodes in size, making it difficult to solve. The authors presented an alternative model for clique partitioning based on a 0-1 quadratic assignment formulation, solved by a Tabu Search heuristic. The authors applied the new clique partitioning approach to 36 test problems and demonstrated improvements in computational time, Grouping Measure and Grouping Efficiency (see section 2.5).

2.4.5 Methods based on the Travelling Salesman Problem

Srinivasan et al [45] used a similarity coefficient matrix to solve the part-families assignment problem. A Travelling Salesman Problem (TSP) analogy was used to form part families from the coefficient matrix. The concept is to identify closed-loops or sub-tours, as they would appear in the solution to the TSP. These closed loops are the basis for grouping parts into families. The researchers proposed a nine step algorithm for solving the part families based on the TSP sub-tour concept. The method and results were compared to the p-median assignment method (discussed in section 2.4.6) and the results were shown to be superior in quality and computational time.

2.4.6 Mathematical Programming Models

Kusiak [46] proposed an integer formulation to the GT problem; this model is known as the p-median model. The model requires the prior calculation of a part-part similarity index s_{ij}. Kusiak proposed an alternative similarity measure for parts based purely on the total number of common processes (machines). In the p-median model, a zero-one decision variable x_{ij} was used to allocate part i to part family j, i.e. $x_{ij} = 1$ if part i belongs to family j, $x_{ij} = 0$ otherwise. The objective of the formulation is to maximize the similarity of parts within a family for all families. Constraints ensures that each part is only assigned to one family and that exactly p families are formed. Machine cells were then created based on the needs of individual part families.

Boctor [23] developed a zero-one model for the GT problem that explicitly considers restrictions on cell size and the objective of reducing the number of exceptional elements, in a solution. The model simultaneously groups machines into cells and parts into families; unlike the p-median model which only forms part families. The model was solved using Simulated Annealing (SA) with 90 experiments applied on a set of ten test problems. In 64 % of the problems SA provided the optimal solution.

Boctor [47] considered the effect of machine and part assignment costs in the GT problem through a linear Mixed Integer Program (MIP) formulation. The model simultaneously assigns machines and parts to cells. For a given set of machines and parts with acquisition and running costs, the problem was to minimize inter-cell material handling costs and machine acquisition and operating cost. An important point to note about this formulation is that it considers pre-existing machines as well as the possibility of purchasing new machines. Simulated annealing was used to solve the model and the optimal solution was found in 69 of 75 test problems.

Saidi-Mehrabad and Ghezavati [48] proposed a stochastic model for cell formation as a linear MIP. The model takes into consideration that processing times for parts and part arrival times are stochastic; these times were modelled as exponential distributions. The model was tested by formulating random problems and solving them by the Branch and Bound algorithm. The authors presented eight test problems, each with 40 parts and 25 machines to be grouped into 6 cells. The results demonstrated a trend where machine utilization increased as inter-cell part movement increased. The economic objective was to determine the most profitable trade-off between these factors.

2.4.7 Heuristic and Artificial Intelligence Approaches

Egilmez et al [49] proposed the use of a Genetic Algorithm (GA) for CMS design with capacity consideration. The problem formulation assumed a deterministic, unchanging demand and capacity requirements for 10 parts. The genetic algorithm used an integer string representation and the objective function was formulated to maximize the similarity between parts grouped into a cell. The results were compared to the capacitated p-median model and were shown to be within 2.44 % of the global optimum.

Gomez et al [50] approached the classical group analysis problem with a Tabu Search (TS) algorithm. The objective was to cluster parts and machines with an emphasis on shared tooling problem and minimization of machine set-up times. The results were compared to other Tabu Search approaches in literature and was demonstrated to produce improved production efficiency.

Nouri and Hong [51] developed a Bacteria Foraging Optimization (BFO) algorithm for the creation of machine cells and part families. The algorithm was applied to twenty five different group analysis problems. The performance of the BFO algorithm was compared to algorithms such as the K-Means and C-link clustering algorithms and a genetic algorithm and was shown to outperform all of these on the given selection of problems.

Won and Currie [52] developed a Fuzzy Adaptive Resonance Theory (ART) neural network for machine cell and part family formation in CMS. The method also considered operation sequences, multiple part visits to the same machine and production volumes as part of the cell formation problem. The strength of the algorithm and method proposed was the ability to be applied to large problems.

Ghosh et al [53] presented a comprehensive literature review of meta-heuristics applied to the group analysis problem. The review includes the application of Simulated Annealing, Evolutionary Algorithms, Ant Colony Optimization and other novel approaches. Chattopadhyay et al [54] presented a comprehensive literature review of artificial intelligence approaches that included various types of neural network approaches to the machine cell and part family formation problem.

2.5 Measures of Performance in Group Analysis

Researchers have aimed to improve the results of cell formation by exploring alternative heuristics and algorithms. Improvements in cell formation are measured against indicators such as exceptional elements, machine utilization, group efficiency and efficacy [20]. A selection of popular performance metrics are discussed here.

Group Efficiency η, is a measure of performance in machine cell grouping that takes into consideration cell work load and the number of inter-cell movements made by parts. Chandrasekharan and Rajagopalan [22, 40] defined this performance measure as a function with weighting q:

$$\eta = q\eta_1 + (1-q)\eta_2 \tag{2}$$

The authors defined the individual efficiencies as follows:

$$\eta_1 = \frac{e_1}{e_1 + e_v} \tag{3}$$

$$\eta_2 = \frac{o - e_v}{(o - e_v) + (e - e_1)} \tag{4}$$

Where e_1 is the number of 1's in the block diagonals, e_v is the number of voids in the block diagonal, e is the total number of 1's and o is the total number of 0's in the entire machine-component incidence matrix. A problem with Group Efficiency is that it is difficult to choose a weighting factor q because the formula yields a high value of η_2; including problems with a large number of exceptional elements. Sarker [20] has also noted that the η_2 has a diminishing influence on efficiency as problems get larger.

The performance measure called Group Efficacy τ was introduced by Kumar and Chandrasekharan [55] to overcome the inadequacy of the Group Efficiency indicator for large machine-component matrices. Group Efficacy is defined as follows:

$$\tau = \frac{e - e_o}{e - e_v} \tag{5}$$

Where e_o is the number of exceptional elements in off-diagonal blocks. This performance measure has been known to be more sensitive to voids in a block diagonal as compared to the number of exceptional elements. Ng [56] proposed a weighted average efficacy γ with weighting q to correct this bias:

$$\gamma = \frac{q(e - e_o)}{q(e + e_v - e_o) + (1 - q)e_o} \tag{6}$$

Miltenburg and Zhang [57] proposed the concept of Grouping Measure ξ_g for a machine-component group. Grouping Measure is the difference between the machine utilization rate ξ_u and the inter-cell part movement rate ξ_m; it is defined as follows:

$$\xi_g = \xi_u - \xi_m = \frac{e_1}{e_1 + e_v} - \frac{e_o}{e} \tag{7}$$

Measures of performance for machine cell and part family formation that consider only the information contained in a machine-component incidence matrix are naive. The number of exceptional elements in off-diagonal blocks are only a crude indicator of the volume of inter-cell part flow. For instance, two exceptional elements corresponding to parts that are in relatively low demand may be more favourable to a manufacturer than one exceptional element for a part that is in great demand. Similarly, voids in the block diagonals of a machine-component incidence matrix are only a crude indicator of the underutilization of machines in a cell. The real extent of the underutilization of a machine is only known when the demands of different parts are taken under consideration. The measures of performance are therefore to be accompanied by other types of post-clustering analysis such as the cost of a solution, the actual inter-cell part volumes, etc.

2.6 Dynamic and Reconfigurable Cellular Manufacturing Systems

The classical Group Analysis problem and solution techniques consider a single-period planning horizon with a static part mix and fixed demand [58]. These methods are not suitable for dynamic manufacturing demands as the cells developed in one period may not be optimal in the next period. The concept of Dynamic Cellular Manufacturing Systems (DCMS) was proposed by Rehault et al [13]. Note that in the literature, DCMS and Reconfigurable CMS (RCMS) are terms that are used interchangeably to refer to a system with cell configurations that changes over time.

The concept of DCMS is illustrated in Figure 2.1, where machines are added, removed or relocated between cells as the manufacturing system progresses from one operating period to another. The reconfiguration/re-layout is initiated by a change in part types and market demand. In the example of Figure 2.1, in Period Two: machine 3 is relocated from Cell 1 to Cell 2, machine 7 is relocated from Cell 2 to Cell 1 and machine 1 is removed from Cell 1 and machine 8 is added in its place. The advantages of the re-layout include simplified scheduling, higher machine utilization, lower work in process and lower inter-cell part flow volumes [59]. If cell configurations remain fixed, as in a traditional CMS, an increasing inter-cell part flow, reduced machine utilization and an increasing tardiness of parts will be observed when part types and demand change. These factors ultimately imply a reduction in the profitability of the system. It is therefore feasible to invest a portion of present profits toward the periodic re-layout of the system to ensure that profitability is maintained in the long-term. The term fixed CMS will henceforth be used to refer to systems where cell configurations do not change.

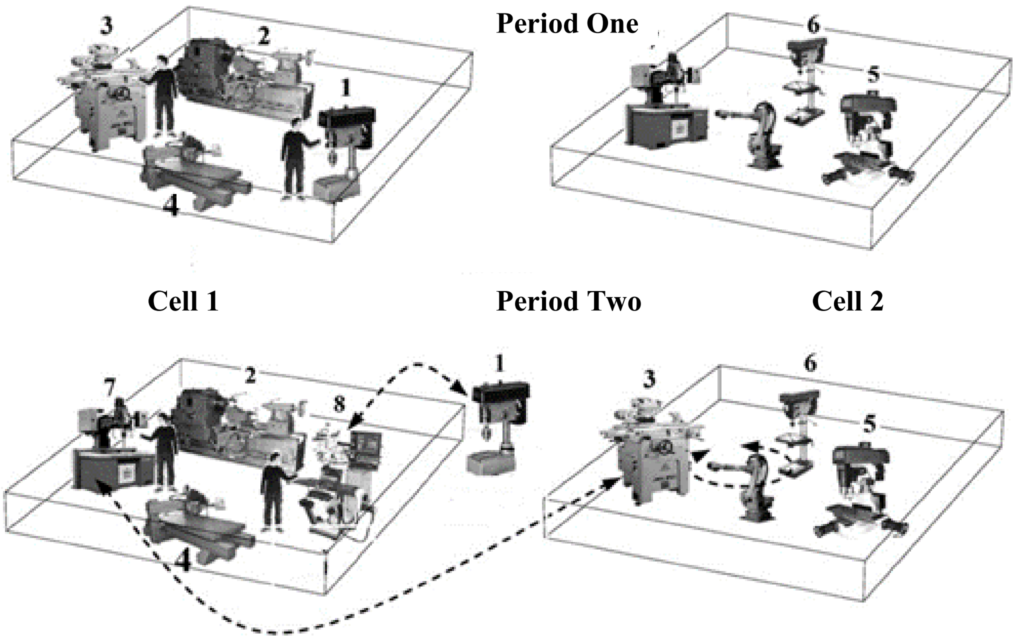

Figure 2.1: Re-Layout of Cells in a DCMS (adapted from [59])

Rehault et al [13] presented a cost minimization quadratic assignment model to aid with the decisions of relocating workstations and machinery between cells. The method proposed did not include any long term planning for re-layout but rather serves as a tool for determining the next configuration of the DCMS when the present configuration is no longer suitable.

Safaei et al [58] developed a mixed integer program for a DCMS. The objective function minimises machine relocation, material handling and reconfiguration costs. The formulation considers inter- and intra-cell material handling, alternative process plans, machine duplication and re-layout via reconfiguration. The model assumes that all machines are flexible, allowing operations to be performed on a variety of machines. The model also takes into consideration the sequence of operations in order to minimize intra-cell part movement. A fuzzy programming-based approach was adopted to solve the model for a two period, two cell, eight part, six machine problem. The method was able to successfully generate new cell configurations in the second period with a demonstration of a cost trade-off between the re-layout of the system and other cost bearing factors such as inter-cell part movement. In a similar work Safaei et al [60] applied a hybrid Simulated Annealing algorithm to solve a larger three period, two cell, eight part, six machine problem. Safaei and Tavakkoli-Moghaddam [59] developed another similar MIP for a DCMS that considers the additional factors of subcontracting and inventory carrying costs.

Mahdavi et al [61] proposed an integer programming model that considered worker assignment. The objective function included penalties for hiring and retrenchment of workers and salary costs when the manufacturing system is reconfigured. The model was able to determine optimal cell configurations over a three period planning horizon with two cells comprising of six machines and a total of six parts.

Bayram and Sahin [62] developed a non-linear mixed integer programming model that considered both inter and intra-cell material handling costs, costs for installing/uninstalling machines as well as the possibility of purchasing new machines in determining new system configurations. The model was solved by a combination of Simulated Annealing and a Linear Program (LP). Due to the complexity of the model it was also solved by a combination of a Genetic Algorithm and a Linear Program. Both SA-LP and GA-LP approaches were shown to obtain near optimal solutions.

Sharifi et al [63] used a dynamic programming approach, combined with a Genetic Algorithm to solve the multi-period cell formation problem. The objective of this study was to use the DCMS design problem as a basis for evaluating the proposed algorithm against other optimization approaches such as LINGO (a modelling approach for solving linear and nonlinear optimization problems) and a Genetic Algorithm.

2.7 Intra-Cell Layout and Line Analysis for CMS

2.7.1 Common Cellular Layouts

The layout of machines within a CMS may follow one of the common patterns as shown in Figure 2.2. The circular layout is typical of a robot orientated cell where the robot is located at the centre of the loop. The U-Shaped layout is often referred to as a one piece flow layout or just-in-time layout [64].

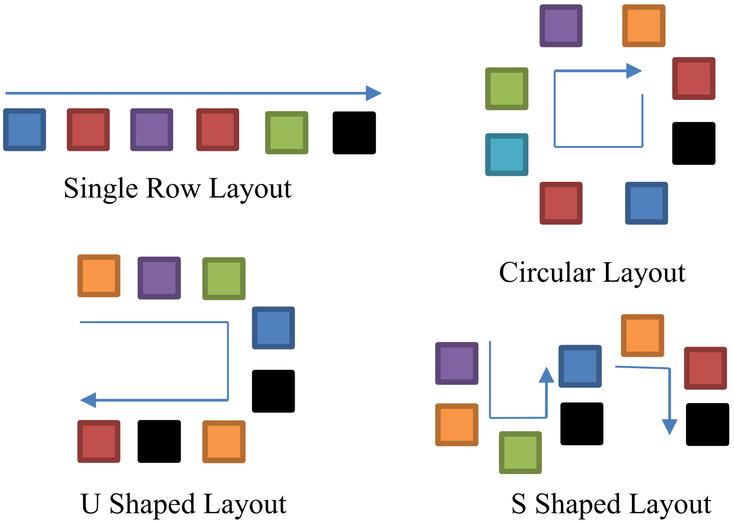

Figure 2.2: Common Cellular Layouts

Irrespective of the type of layout selected for implementation, the ideal scenario is that all parts should flow in a single direction. Flow in a single direction is referred to as unidirectional flow. When the sequence of operations on a part does not follow the sequence in which machines are laid out, the part will bypass some machines. In order to follow the prescribed sequence of operations, the part may backtrack to machines that were previously bypassed. Part backtracking or the backward flow of parts against the planned flow direction increases the complexity of managing the cell and may incur costs due to the additionally material handling. The reconfigurability of DCMS/RCMS should be exploited, where possible, to eliminate part backtracking movements.

2.7.2 Unidirectional flow by Part Family Design

Unidirectional flow within cells may be promoted by appropriate part family formation. Goyal and Jain [65] presented a part family formation that promotes unidirectional flow within cells by an analysis of the similarities in operation sequences between parts. They proposed the BMIM (Bypassing Moves and Idle Machines) similarity coefficient and used the average linkage clustering method to generate part families.

2.7.3 Unidirectional Flow by Machine Sequence

The primary method of promoting unidirectional flow is the appropriate design of the cell layout. The problem of cell design for unidirectional flow is referred to as Line Analysis. Burbidge [15] presented a heuristic for Line Analysis that uses the sequence in which machines are visited by parts to iteratively simplify a network flow diagram that represents flow within a cell.

An alternative method for obtaining unidirectional flow involves the use of a From-To matrix [66]. The development of a From-To matrix requires the sequence of operations for each part in a family to be taken under consideration. Table 2.3 presents an example part family consisting of parts 1-8 with the sequence of machines on which each manufacturing process will take place. For example, the first process on part 1 is performed on machine 2, the second process is performed on machine 5 etc.

Table 2.3: Machine Visitation Sequence for Example Part Family

Part	Sequence of Machine Visitation
1	2-1-5-6-3-4-7-8-9-7-5
2	1-2-3-4-6-5-7-10-8
3	2-3-1-4-5-6-8-9-10
4	2-3-1-4-5-7-6-8-9-10-9
5	1-2-3-6-4-7-9-8-10
6	2-3-1-5-7-10-9
7	3-1-4-6-10-9
8	2-5-6-9-7-8-10-9

By examining Table 2.3 it is clear that each part in the family visits machines for processing in different sequences. Promoting unidirectional flow (minimizing backtracking/back flow) is a challenge that is addressed by arranging machines in an optimized sequence.

Table 2.4: From-To Frequency Matrix (M = Machine)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0	2	0	3	2	0	0	0	0	0
M2	1	0	5	0	1	0	0	0	0	0
M3	4	0	0	2	0	1	0	0	0	0
M4	0	0	0	0	2	2	2	0	0	0
M5	0	0	0	0	0	3	3	0	0	0
M6	0	0	1	1	1	0	0	2	1	1
M7	0	0	0	0	1	1	0	2	1	2
M8	0	0	0	0	0	0	0	0	3	2
M9	0	0	0	0	0	0	2	1	0	2
M10	0	0	0	0	0	0	0	1	4	0
Sum	5	2	6	6	7	7	7	6	9	7

Table 2.4 represents the ‘From-To’ frequency in which parts move from one machine to another. Here the prefix ‘M’ is added to machine numbers for additional clarity. Table 2.4 is populated from Table 2.3; as an example consider parts 1, 3 and 8 which move from machine 5 to machine 6. The From-To frequency between machine 5 and machine 6 is 3, highlighted in green in Table 2.3. The From-To matrix is used in practice to identify backtracking movements of parts; all numbers listed below the diagonal of the matrix indicate instances of parts backtracking in order to visit a machine that was previously bypassed. If the machines are arranged in the sequence 1,2,3,4,5,6,7,8,9,10 there will be 18 instances of backtracking in the flow patterns of parts inside the cell.

Table 2.5: Rearranged From-To Matrix (M=Machine)

	M2	**M3**	**M1**	**M4**	**M5**	**M7**	**M6**	**M8**	**M10**	**M9**
M2	0	5	1	0	1	0	0	0	0	0
M3	0	0	4	2	0	0	1	0	0	0
M1	2	0	0	3	2	0	0	0	0	2
M4	0	0	0	0	2	2	2	0	0	0
M5	0	0	0	0	0	3	3	0	0	0
M7	0	0	0	0	1	0	1	2	2	0
M6	0	1	0	1	1	0	0	2	1	0
M8	0	0	0	0	0	0	0	0	2	0
M10	0	0	0	0	0	0	0	1	0	0
M9	0	0	0	0	0	2	0	1	2	0

In order to reduce the number of backtracking movements, the sequence in which machines are arranged can be reordered to produce a smaller sum of numbers below the diagonal of the matrix. Table 2.5 shows a rearranged From-To matrix, where the number of instances of backtracking movements were reduced from 18 down to 12 by reordering the sequence in which machines would be placed on the factory floor.

Hollier [67] proposed four heuristic methods that use the information in a From-To matrix to obtain acceptable machine layouts with minimum backtracking movements. The heuristics are based on the fundamental principle that machines with the fewest ‘to’ instances should be placed at the beginning of the sequence. By examining the column sum of Table 2.4 it is found that machine 2 has the lowest ‘to’ count and machine 9 has the highest ‘to’ count. In the rearranged matrix machine 2 is the first in the sequence and machine 9 is the last in the sequence.

2.7.4 Single Row Layout Problem

A problem related to the Line Analysis problem is the Single Row Layout Problem (SRLP) [68]. The SRLP aims to minimise the product of flow frequency and distance travelled; the objective is described by the following equation [69]:

$$\min \sum_{i=1}^{n-1} \sum_{j=1+1}^{n} c_{ij} d_{ij} \qquad (8)$$

Where n is the number of facilities to be arranged in a single row, c_{ij} is the amount of flow between facility i and facility j and d_{ij} is the distance between the centres of the facilities. Ho and Moodie [70] explored the combination of Hollier's methods with a Simulated Annealing algorithm applied to the SRLP to generate cell layouts that minimise total flow distance and the maximise the number of in-sequence part movements. Other works include the research by Solimanpur et al [71], who presented an alternative nonlinear zero-one programming model for the SRLP that minimizes the backtracking distances travelled by material handling platforms.

In the literature that has been reviewed, the problem of creating unidirectional part flow and the SRLP for DCMS/RCMS has received insufficient attention. The cost of intra-cell part flows has been considered by authors such as Safaei et al [60] and Tavakkoli- Moghaddam [2] in the problem of cell creation. The intra-cell re-layout of machines was not fully exploited to improve the internal flow characteristics of cells.

2.8 Assessment of Existing DCMS and RCMS Methods

The literature that has been reviewed reveals that researchers use the terms DCMS and RCMS interchangeably to refer to cellular systems where machine relocation is permitted. In all the cases reviewed the problem was first set-up as an integer or mixed integer optimization model before being processed and solved by the various methods available. The researchers did not integrate or modify any of the established classical solution techniques in order to adapt them to the multi-period problem. Common optimization objectives included minimizing inter-cell material handling, intra-cell material handling and the cost of reconfiguration. Other objectives included minimizing labour costs and the costs of hiring and firing workers due to new factory configurations.

In the literature reviewed in section 2.6, the problems were small, consisting of no more than ten parts and the planning horizon was no more than three periods. The size of the problems that could be solved was usually restricted by the complexity of the models. In many instances, part family combinations were also reconfigurable, where parts did not belong a definite family.

In the cases reviewed, there was no initial design of the factory floor space to support reconfigurability. All machines were allowed to move to any cell without restriction, irrespective of physical constraints such as the need for sufficient floor space and supply connections such as compressed air, water and single-phase or three-phase electricity. The relative positioning of other machines when accepting new machines into a cell was also not considered. Although intra-cell material handling costs were taken under consideration, insufficient attention was given to the internal design of cells to support unidirectional part flow.

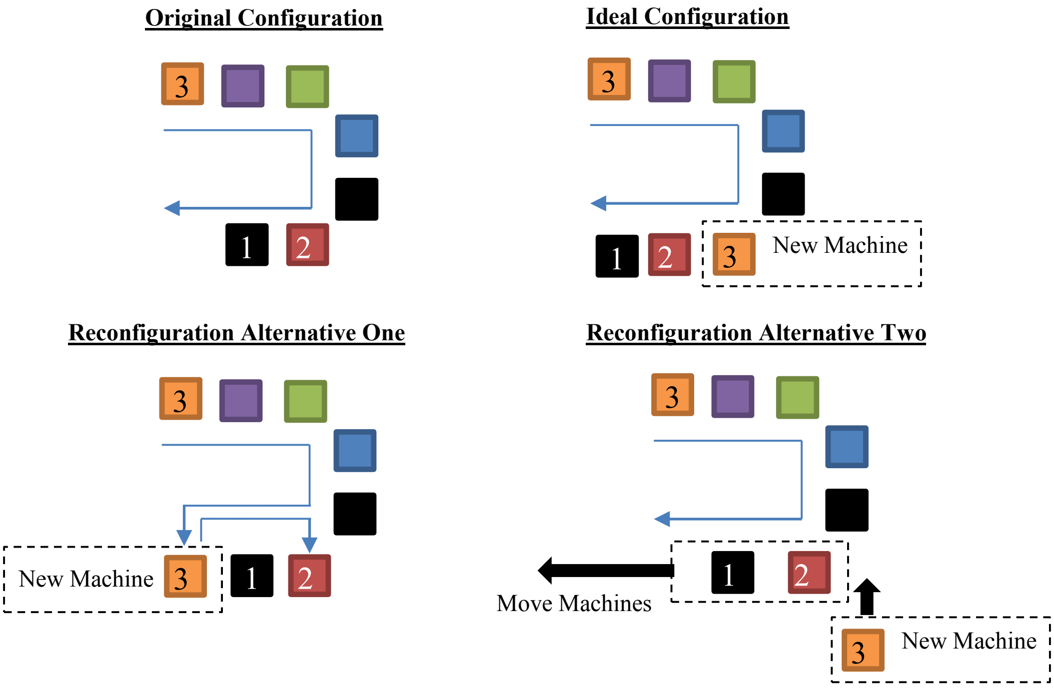

Figure 2.3: Impact of Relative Machine Position on Cell Reconfiguration

Consider the scenario in Figure 2.3 where an additional copy of machine 3 has been assigned to a cell by a DCMS/RCMS multi-period group analysis model. In order to support unidirectional part flow, the machine must be placed before machine 1 and 2 according to the flow sequence as shown in the ideal configuration. In this example no provision has been made for the cell to accept machine 3. In order to accept the additional machine two choices exist; in the first reconfiguration alternative, the machine is added to the end of the sequence. This does not support unidirectional flow and part backtracking moves are introduced. In the second reconfiguration alternative, machines 1 and 2 are relocated in order to create a space for the new machine. If this choice is adopted three machines machine movements are involved and the actual cost of reconfiguration is more than what may be reflected in the DCMS/RCMS optimization model. A closer investigation may reveal that the reconfiguration cost does not outweigh the benefit of the cell accepting the new machine.

The situation may exist where reconfiguration as shown in the second alternative is infeasible. If machine 1 or machine 2 are large and immovable machines, this creates an irreversibility. The ideal configuration for unidirectional part flow may therefore be impossible and the cell will be forced to operate under a sub-optimal configuration.

2.9 Direction for this Research

This research will address the design of Reconfigurable Cellular Manufacturing Systems. Koren et al [1, 9] define a reconfigurable manufacturing system as a system that is designed at the outset for changes in its structure (see section 1.3) . Based on this definition, a RCMS is distinguished from a DCMS through its initial design for configuration changes. A significant portion of this research will be dedicated to the initial design of the factory floor space in order to support feasible and cost effective reconfigurations. Attention will be given to both group analysis and line analysis to ensure that the process of reconfiguration enhances the functions of a manufacturing cell and that these manufacturing functions are arranged to support unidirectional flow.

In this research the reconfiguration of manufacturing cells will occur within the context of part families; this is a distinguishing factor between this research and other DCMS literature. The decision to perform reconfiguration within the context of a part family is also consistent with the definition of a reconfigurable system offered by Koren et al [1, 9]. Parts within a family may be active or inactive during the various periods under review, however, members of a part family will not leave the cluster to become members of another part family. Each cell will be responsible for the manufacturing of a single part family, therefore the manufacturing of a part will never become the responsibility of another cell under future system configurations. Practically this can reduce the amount of operator retraining required after reconfigurations and will reduce the need to rotate operators between cells.

2.10 Chapter Summary

This chapter presented a review of methods for the design of cellular manufacturing systems. Methods for the design of fixed CMS were presented to highlight the number of ways in which the problem of machine cell and part family formation may be approached. The design of a CMS with configuration changes to cells is a difficult problem that is subject to numerous cost bearing factors and physical limitations. A review of recent literature in the field of DCMS and RCMS design was reviewed. The terms were found to be used interchangeably by authors referring to the problem of multi-period machine cell formation. By examining the literature it was found that the available methods do not consider the initial set-up of the factory floor to support future configuration changes.

It was also found that insufficient attention was given to the intra-cell layout design to support unidirectional part flow when machines are added to cells in future periods. This research will focus on the initial design of the manufacturing system floor space to support configuration changes. Thereafter, the matter of evolving cell configurations will be addressed within the context of the physical allowance that has been made for the re-layout of machines.

3 Designing RCMS for Reconfiguration

3.1 Chapter Introduction

The design of a viable Reconfigurable Cellular Manufacturing System requires provision for reconfigurability to be included in the initial system installation. This chapter introduces the concept of a Factory Configuration Template (FCT). The FCT concept has been developed in order to create a floor layout design for a RCMS that supports future reconfigurations. If the FCT is implemented in the factory at the point of installing a CMS, the system will be reconfigurable from the outset. The features of the FCT are presented and matters pertaining to the creation of the FCT are discussed.

3.2 Re-layout of Cellular Manufacturing Systems

Cellular Manufacturing Systems exploit similarities in the manufacturing requirements of parts and products. The benefits of CMS have been reported to include reduced part set-up times on machines, better organization of tooling, lower work-in-process levels, reduced expenditure on material handling, less chaotic work flow, improved product quality, lower complexity scheduling and hassle free system management [3, 8]. A CMS will remain optimal over a planning horizon in which part mix and demand do not change [7], however, the advantage of grouping a set of machines in one period may prove to be a disadvantage in the next period. Over a period of time, the following will occur:

- increased inter-cell material handling movements;
- increased work-in-process;
- longer manufacturing lead times;
- lower machine utilization.

These symptoms increase the difficulty of managing the system and reduce profitability. These symptoms indicate that the cell groupings and layout of the manufacturing system is no longer best suited to the production requirements of current parts and products. These symptoms occur due to the introduction of new parts, alterations to process plans and changes to process routings. These symptoms also occur when the relative demand characteristic of individual parts change as compared to what the system was initially designed to facilitate. Case studies have been documented where the reformulation of manufacturing cells and the re-layout of cells and machinery was necessary and beneficial to the long term productivity of the system. Allam and Irani [72] present a case study where cells were redesigned and equipment was rearranged in an unnamed small wheel manufacturing CMS. The result was improved material flow characteristics and a 46 % reduction in the material handling travel distance for parts visiting a mill-turn machining centre.

In cases where CMS were redesigned, the process of rearranging the layout of machines and cells was conducted after symptoms of inefficiency were observed. This research presents a methodology of pre-emptive planning for the reconfiguration of manufacturing cells, through the re-layout of machines, across multiple time periods in order to avoid the manifestation of the symptoms of inefficient machine clusters.

3.3 Reconfiguration Planning Strategy

There are two approaches that may be adopted with respect to formulating manufacturing cells and planning subsequent reconfigurations. The first approach is to plan cellular reconfigurations based on forecast changes to the master production schedule; this is a short term approach. The second approach is to plan cellular reconfigurations based on the long term product release and market expansion strategy of the enterprise.

Researches such as Safaei, Tavakkoli-Moghaddam and Sassani [59, 73] have adopted the first approach, presenting methods for cell formation and reconfiguration that are tightly coupled to Production Planning (PP). Note that in this context, production planning refers to planning the management of production resources on the scale of weeks. This approach assumes that reconfiguration occurs frequently and is directly driven by anticipated changes in the master production schedule. The cell formation and reconfiguration methods developed by these authors consequently include inventory carrying costs, etc.

The approach to cell formation and reconfiguration planning adopted in this work views reconfiguration as a long term strategy to maintaining the profitability of a CMS. Reconfiguration is therefore not performed in a manner that is reactive to the production schedule but rather pro-active with respect to the long term product release and marketing expansion strategies of the business. Reconfiguration is an activity that is disruptive to the productivity of a manufacturing system, therefore frequent reconfiguration on the scale of weeks is practically undesirable. Manufacturing facilities have planned “shutdown and maintenance” periods that are determined by the seasonal behaviour of markets and national holidays. These periods are usually scheduled annually or biannually. The feasibility of performing reconfiguration during these planned shutdown and maintenance periods is higher and methods for planning the reconfiguration of manufacturing cells should be based on an appropriate scale of time (i.e. biannually and annually). The adoption of an appropriate scale of time ultimately determines the factors that are considered in the methods of designing Reconfigurable CMS.

3.4 Provision for Reconfigurable Machine Cells

A RMS is a system that is designed for a rapid change in structure [1, 9]. A system cannot change its structure rapidly and economically if the current manufacturing system design does not facilitate a changeover to a new configuration. A RCMS can therefore be defined from a design perspective as a system whose current manufacturing cells support a quick, seamless evolution to a new cellular configuration. Machines can only be relocated within a system if:

- a budget has been made available for re-layout of cells;
- a space is available for machines in a different cell;
- the cell is able to provide the electricity, water, pressurized air or any other supply that the machine may need for its operation;
- the machine is not confined by surrounding machines or structures that would prevent it from being moved.

Machines that are likely to assume different locations across multiple time periods must be identified as early as the cell formation phase. If these machines are identified early, provision can be made during the commissioning of the system to ensure that cells are practically reconfigurable.

3.5 Factory Configuration Template

This research introduces the concept of a Factory Configuration Template (FCT) for RCMS. The FCT is a plan of the factory floor that has been designed to support reconfiguration. The plan is also a template because it defines all possible cell configurations that the RCMS will be allowed to assume over the planning horizon for which the FCT has been designed. The template defines possible reconfigurations that may be implemented to reduce material handling costs and improve the utilization of machinery while considering the long term profitability and production objectives of the business. At an elementary level, the template defines the machines that each cell is able to accept and also identify sites for fixed and mobile machines. Consider the following definitions of static and mobile machines:

Fixed Machine: a machine that resides permanently in one cell, this may be due to its physical characteristics such as weight. A machine may also be classified as a fixed machine due to no advantage being found in the possibility of relocating it to other cells.

Mobile Machine: a machine that may be located in multiple cells or multiple sites within a single cell to enable improved resource utilization and cost saving. The relocation of such a machine must be physically possible.

Figure 3.1 illustrates the concept of the FCT for reconfigurable CMS. The template identifies sites for fixed and mobile machines within cells. Fixed machines may be automatically assigned to sites identified for them; for example machines 3 and 7 have one each and may be placed in the designated sites without further analysis. Sites that are occupied are shaded in red and sites that are available are outlined in green. Mobile machines 1, 2, 4, 5, 6, 8, 9 and 10 have multiple sites that they may occupy as per the template, however, each site will remain empty until a need or an advantage has been identified in placing the machine at the site.

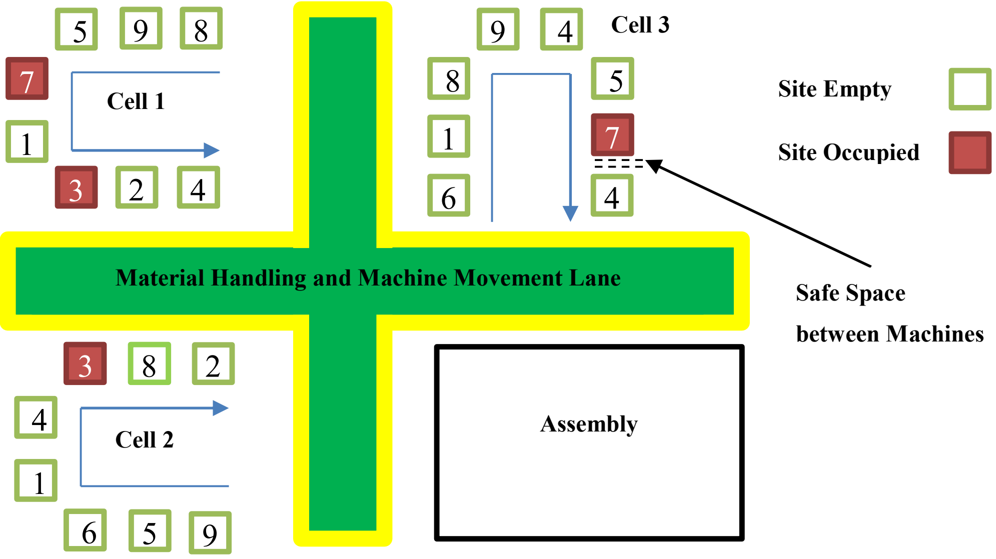

Figure 3.1: Graphical Illustration of Factory Configuration Template

As an example, consider machine 5 which has multiple possible locations. In the first year one copy of machine 5 is available and is required by Cell 3. In year two, an additional copy of machine 5 is purchased and the part family corresponding to Cell 1 has the greatest need for the new machine. In year three, the part family corresponding to Cell 2 may have a greater need of machine 5 and another unit of machine 5 is unavailable to the system. The machine may be relocated to this cell from either Cell 1 or Cell 3. Because the factory floor has been designed according to a plan that has made provision for reconfiguration, machine 5 may be placed in Cell 3 without affecting the unidirectional flow of parts or requiring machine 9 to be moved.

3.6 Features of the Reconfiguration Template

The FCT is effectively a plan of sites to be created on the factory floor and within cells to support reconfiguration. In addition to creating sites for the future re-layout of the system, the FCT must include provision for lanes in which machines can be moved. Material handling lanes may be created to be wide enough to allow machines to move. The FCT contains dedicated sites for each machine type, this is to avoid the need to move additional machines and to avoid irreversibility's in the cell configuration. The example of Figure 2.3 is relevant. In addition to these factors, dedicated sites are also created on the factory floor since machines have individual floor space requirements. A larger machine cannot be placed in a site that was previously created for a machine with a smaller footprint. Many machines require a safe distance to be maintained between other machines. This is to avoid interference between machines and to create a safe working environment. The FCT must include provision for distances between machines and a safe operating space for workers.

Additional information that may be added to the FCT include the provision of supplies of water, electricity, compressed air and other types of connections that machines may need. These connections are often built into the physical structure of a building and the installation of these connections may be costly and inconvenient if installed or altered as an afterthought.

From a theoretical point of view, the FCT will dictate the formulation of multi-period cell formation models in the following ways:

- it defines a fixed number of cells for the planning horizon;
- it identifies specific sites for fixed and mobile machines,;
- a machine cannot be assigned to a cell if a site is not available for it according to the FCT;
- it determines the number of machines of a single type that the cell may host; for example in Figure 3.1, a maximum of two machines of type 4 are allowed in Cell 3;
- it imposes an upper bound on cell size (number of machines in a cell) by the number of sites available;
- it defines the shape and orientation of the cells that are to be implemented;
- it defines the sequence in which machines may be placed within cells;
- it defines the default direction of material/part flow in cells.

3.7 The RCMS Life Cycle under a Factory Configuration Template

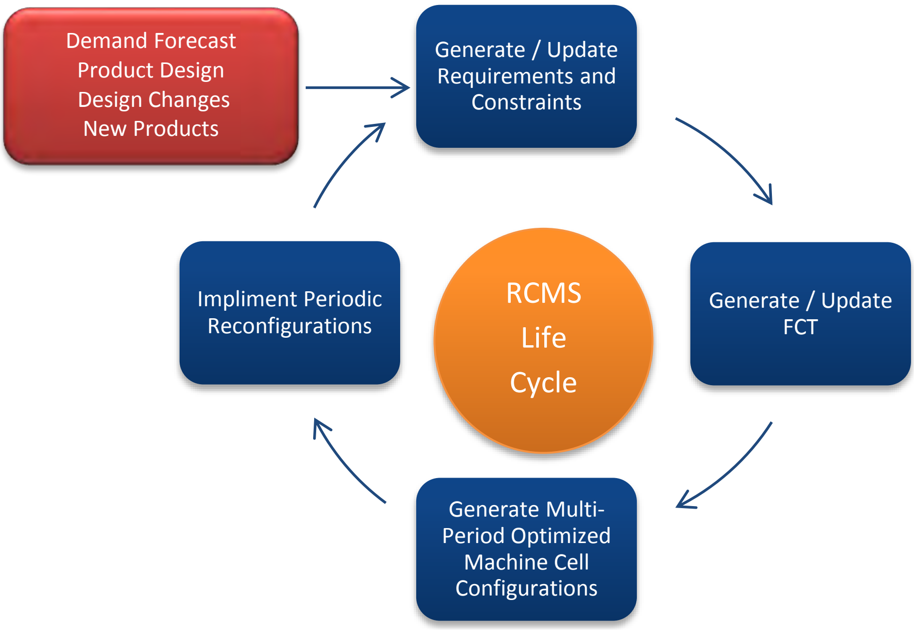

Figure 3.2: Reconfigurable Manufacturing Systems Life Cycle

The primary advantage of operating a RCMS within the constraints imposed by a Factory Configuration Template is that potential machine movements are known and this is factored into layout and space planning of the system. This ensures that configurations proposed by a theoretical model are feasible to implement. The disadvantage of operating a RCMS within the constraints imposed by a FCT is that the template is restrictive and will eventually cease to allow optimal configurations to be implemented.

Figure 3.2 illustrates the lifecycle of RCMS; the design process begins by first obtaining information pertaining to demand forecasts, product design information in relation to manufacturing requirements and information pertaining to the planned release of new products. This information is used to generate requirements and constraints that are then used to generate a FCT that is relevant for many years.

Once a viable FCT has been established the system may be reconfigured within bounds. Advantageous configurations are obtained by the application of multi-period reconfiguration planning models that propose adjustments to cell functionality and capacity as needed. These models must also use reconfiguration to optimise the inter-cell and intra-cell flow of components.

Over time, new information will become available on emerging market trends and the release of new products. Sub-contractor type manufacturing facilities will also obtain new contracts involving different parts with new demand requirements. When new information emerges this requires that the FCT be updated, however, this must be done in a manner that does not incur prohibitively cost intensive changes to the infrastructure within the production facility.

3.8 Prerequisite Information and Assumptions

3.8.1 Part Demand and Mix

A significant amount of prerequisite information is required prior to the implementation of a RCMS. Information pertaining to immediate and future products and their manufacturing requirements is required. Forecasting the demand of current and future products is essential to determining the types of parts and the volumes in which they must be produced. In subcontractor type manufacturing systems, operating on the make-to-order strategy, this information might be available due to contracts signed with other manufacturers or chain stores. These contracts may specify the types of products and the volumes in which they must be produced based a multi-year agreement.

In other types of businesses, operating on the make-to-stock strategy; information on future part demand and mix may be determined through liaison with the ‘product development department’ and the ‘market research department’. Product developers can provide information on the parts to be used in future products and the corresponding manufacturing requirements. Market researchers can provide forecast information on the performance of future products in the market based on the performance of current products or the performance of competing products. Forecasting future part demand and mix is not an objective of this research. This research provides methods for RCMS design once such information is already available.

3.8.2 Machine Flexibility and Machine Portfolios

The formation of the FCT requires that the machines to be used by the system be determined beforehand. The flexibility of machines is an important aspect of ensuring the ability of a CMS to cope with changes to parts and product designs. Researchers such as Taha and Rostam [74] have addressed the matter of appropriate machine selection to impart necessary flexibility to manufacturing cells. In this research, the process of developing the FCT begins with the same information as CMS and DCMS design problems; that is the assumption that relevant machines have been identified and relationships between parts and machines have been established. These relationships include processing times and sequences of operations on machines.

3.8.3 Machine Sizes and Floor Space Requirements

The contributions of this research are the methods that may be used to identify sites on the FCT. The identification of sites on the factory floor is the component of FCT design that requires the development of new analytical tools. The relation of machine sizes to the floor area of sites is a rudimentary procedure and will not be given significant attention in this research.

3.8.4 Design of Assembly Cells

This research does not deal specifically with assembly cells. Gyulaia et al [75] stated that manual assembly of products is prevelant in systems where there is high part variety due to the rigidity of automation in accomodating part variations. Manual assembly processes are intrinsically reconfigurable due to human ability. Assembly cells may not require machines to be relocated but may require the re-layout of manual workstations. The methods developed in this research may be applied to the re-layout of assembly stations, however, no example is presented. This research focues on the re-layout of component manufacturing cells over multiple periods.

3.9 Chapter Summary

This chapter presented the concept of a Factory Configuration Template as a means for making appropriate provision for future configuration changes in a reconfigurable CMS. The FCT identifies future potential sites that machines may occupy, this allows a manufacturer to make sufficient provision for floor space and other resources that machines may need. The FCT requires a significant amount of pre-requisite information to be designed. Assumptions with respect pre-requisite information and other simplifying assumptions that are applicable to this research were addressed. The chapters that follow will proceed with methods that have been developed for FCT design and the models that are used to plan reconfigurations within the bounds of the FCT.

This page was intentionally left blank.

4 Site Identification for Factory Configuration Templates

4.1 Chapter Introduction

The creation of a Factory Configuration Template requires that potential sites for machines in be identified in multiple cells. The machines may occupy these sites at the initial system installation or in future system configurations where the re-layout of machines has been conducted. This chapter presents the development of a fuzzy clustering method for site identification in the FCT. The suitable identification of sites is essential to ensuring that the FCT imparts the appropriate level of reconfigurability to cells in order to meet future manufacturing demands.

4.2 Significance of Fuzzy Clustering in FCT Formation

The creation of the FCT requires the identification of sites for machines in cells. Hard partitioning, which is a typical characteristic of well-known group analysis algorithms, restricts the membership μ_{ik} of machine/part k in cluster i to hold a value in the set $\{0, 1\}$, ($1 \leq i \leq C$ and $1 \leq k \leq N$). Hard partitioning therefore implies the definite placement of a machine in a cell and is not suitable for FCT creation. The membership of N data points with respect to C clusters usually represented in a $C \times N$ matrix:

$$U = [\mu_{ik}]_{c \times N} \tag{9}$$

Fuzzy partitioning, which is a characteristic of fuzzy clustering methods, allows the membership μ_{ik} of machine/part k in cluster i to hold real values in the range $[0, 1]$. A machine is not restricted to be an exclusive member of a cell as in conventional group analysis techniques. If the membership value of a machine in a cell is significant then it is justified that a site be created for the machine in that cell. The assignment of sites to machines in cells according to membership values is the approach that is adopted in this research. The method of obtaining membership values is determined by an appropriate dissimilarity coefficient, a membership model and the application of a fuzzy clustering model; these are presented in the sections that follow.

4.3 The Fuzzy C-Means and Fuzzy C-Medoids Models

4.3.1 The Fuzzy C-Means Model

The model selected for further investigation was the Fuzzy C-Means Clustering (FCM) model, first formulated by [76] and improved by Bezdek [77]. The model is a modification of the K-Means (hard C-Means) algorithm originally developed by Macqueen [78]. The model is based on the concept of clustering around prototypes. Prototypes may be real elements from the data set or artificial elements that act as the nucleus for cluster formation.

The Fuzzy C-Means Clustering Model is as follows:

$$min \sum_{i=1}^{C} \sum_{k=1}^{N} (\mu_{ik})^m \|z_k - v_i\|_A^2 \tag{10}$$

Subject to:

$$\sum_{i=1}^{C} \mu_{ik} = 1\,, \qquad 1 \le k \le N \tag{11}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \qquad 1 \le i \le C \tag{12}$$

$$\mu_{ik} \in [0,1], \qquad 1 \le i \le C, \qquad 1 \le k \le N \tag{13}$$

The objective function of the model minimises the product of membership values and the squared inner-product distance norm between a data point and the cluster prototype. The value m determines the fuzziness of clusters for $m>1$; in this research $m=2$ will be used throughout. The first constraint of the model ensures that the sum of memberships for each data element across all clusters equals one. The membership value is therefore comparable to a percentage. The second constraint ensures that all memberships are not allocated to a single cluster.

In this model z_k is a data point existing in n-dimensional space:

$$z_k = [z_{1k}, \dots, z_{nk}]^T, \qquad z_k \in R^n \tag{14}$$

and the set of N elements to be clustered is:

$$Z = \{z_k | k = 1, \dots, N\} \tag{15}$$

The vectors z_k are usually represented by an $n\ X\ N$ matrix, where the rows are the data elements and the columns are measured dimensions:

$$Z = \begin{bmatrix} z_{11} & \cdots & z_{1N} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nN} \end{bmatrix} \tag{16}$$

The objective of optimization problem represented in equation 10 requires the selection of cluster prototypes and the allocation of membership values to data elements, so as to minimize the “in cluster distance” or dissimilarity of clustered elements. The vector of cluster prototypes is:

$$V = [v_1, \dots, v_c], \qquad v_i \in R^n \tag{17}$$

For data in n-dimensional Euclidean space, the squared inner-product distance norm between a data point and the cluster prototype is given by equation 18, were A is the norm-inducing matrix.

$$D_{ikA}^2 = \|z_k - v_i\|_A^2 = (z_k - v_i)^T A (z_k - v_i) \tag{18}$$

4.3.2 Membership Models

There is a variety of models that may be used to relate the membership of an element to a cluster; two examples provided by Krishnapuram et al [79] are:

$$\mu_{ik} = \frac{\left(\frac{1}{D_{ik}}\right)^{1/(m-1)}}{\sum_{j=1}^{C}\left(\frac{1}{D_{jk}}\right)^{1/(m-1)}} \tag{19}$$

$$\mu_{ik}^{m} = \frac{exp\{-\beta D_{ik}\}}{\sum_{j=1}^{C} exp\{-\beta D_{jk}\}} \tag{20}$$

where β is a constant and D_{ik} is the dissimilarity between object k and cluster i. Usually D_{ik} is the dissimilarity between object k and a representative object of cluster i (i.e. the cluster prototype). Equation 19 is the original Fuzzy C-Means membership model and will be adopted in this research. Other membership models, not listed here are documented in in Krishnapuram et al [79].

4.3.3 Fuzzy C-Medoids Model

The Fuzzy C-Medoids model is a special case of the Fuzzy C-Means model with the restriction that cluster prototypes are data elements from the set Z. The term medoid is used to refer to a cluster prototype from the set Z. This is useful for group analysis type problems where data in the machine-part incidence matrix are not elements of $\mathbb{R}$ but elements of $\mathbb{Z}_2$. Most authors do not choose to distinguish this case as a new model from the standard Fuzzy C-Means model, however, for clarity, the special case is presented here:

$$min \sum_{i \in V} \sum_{k \in Z} (\mu_{ik})^m D_{ik} \tag{21}$$

Subject to:

$$\sum_{i \epsilon V}^{C} \mu_{ik} = 1, \qquad k \in Z \tag{22}$$

$$0 < \sum_{k \epsilon Z}^{N} \mu_{ik} < N, \qquad i \in V \tag{23}$$

$$\mu_{ik} \in [0,1], \qquad i \in V, \qquad k \in Z, \qquad V \subsetneq Z \tag{24}$$

In addition to requiring that cluster prototypes being members of the data set Z, the model also allows an arbitrary distance or dissimilarity measure D_{ik} to be selected by the user. An investigation of an appropriate dissimilarity measure will be conducted in section 4.4.

4.3.4 The Fuzzy C-Medoids Algorithm

Krishnapuram et al [79] presented the Fuzzy C-Medoids (FCMdd) algorithm for fuzzy clustering based on partitioning around medoids. The algorithm is classified as an Alternating Cluster Estimation heuristic and is not guaranteed to always produce the best solution. The algorithm has an alternating iteration architecture where elements from the data set are alternated as medoids in order to minimize the objective function. The solution to which the algorithm converges can change according to the initial selection of medoid elements from the data set; it is therefore advised that multiple experiments with varying initial medoids be conducted. The algorithm as presented in Krishnapuram et al [80] is as follows:

FCMdd Algorithm

```
//Pre-processing: Compute a square dissimilarity matrix D for all data elements within the set Z

C=Clusters; //Fix number of clusters
it_count = 0; //Set iteration counter to zero
ITMAX= MaxIterations; //Define maximum number of iterations
$V=\{v_1, \ldots v_c\}$, $v_i \in Z$; //Randomly select an initial set of medoids from Z

Repeat
//Compute membership of all elements with respect to current medoids
    for i = 1 to c
        for k = 1 to N
            compute $\mu_{ik}$ ; // Compute using equation 19
        end
    end
    $V_{old} = V$ ;//Store current medoids
    for i = 1 to C
        $q = \arg\min_{1 \le k \le N} \sum_{j=1}^{N} (\mu_{ik})^m D_{jk}$ ; //Determine the index of the new medoid
        $v_i = z_q$; // Add the new medoid to the set of medoids
    end

    it_count= it_count +1;
Until ($V_{old}$ = V or it_count=ITMAX)
```

Prior to executing the algorithm a square dissimilarity matrix *D* must be calculated for all data elements in the set *Z*; if the cardinality of *Z* is *n*, then the dimensions of *D* is $n \times n$. The dissimilarity between each element in *Z* may be computed using any appropriate dissimilarity measure.

The algorithm begins by fixing the number desired clusters *C*. The Fuzzy C-Means and Fuzzy C-Medoids models do not have any means of identifying the natural number of clusters that exist in a data set. Cluster silhouettes, developed by Rousseeuw [80], are a graphical aid in assessing the appropriate number of clusters in a dataset. The method will be presented in section 4.3.6.

Once an appropriate number of clusters has been determined, the iteration counter is set to zero and a random set of prototype's (medoids) *V* is selected from the set *Z* ($V \subsetneq Z$). These elements serve as the initial set of cluster prototypes. The algorithm then enters a repeating main loop and the first step within the loop is to compute the membership values μ_{ik}. The membership is computed using equation 19 in the structure of a nested *for* loop. The current set of medoids is then stored and the algorithm proceeds to test new elements from the set *Z* as potential medoids for the next iteration of the main loop. The medoid that replaces the medoid v_i is the one that minimizes the sum of the product of membership μ_{ik} and distance D_{jk} for all *k* and *j*.

The main repeating loop of the algorithm terminates when there is no change to the set *V* or if the maximum number of iterations of the loop has been reached. The outputs of the algorithm are a final set of cluster prototypes $V=\{v_1,\ldots v_c\}$ and the final membership matrix $U=[\mu_{ik}]$. The membership matrix is of particular interest as it can provide an indication of sites that should be created for machines in cells on a Factory Configuration Template.

4.3.5 FCMdd Algorithm and Constraint Satisfaction

The first constraint of the Fuzzy K-Medoids model requires that the sum of membership values for a single data element *k* across all *C* clusters must equal one. The FCMdd algorithm described here satisfies the first constraint (equation 22) of the model through the use if an appropriate membership model for μ_{ik}. In particular, the membership model of equation 19 satisfies this constraint.

The second constraint (equation 23) prevents all elements in the set *Z* from being allocated to one cluster. The algorithm does not include any mechanism for satisfying the second constraint of the model and it is possible that the membership of all elements in *Z* may be allocated to a single cluster. The algorithm may converge with a single element *k'* being selected more than once, up to a total of *C* times, to serve as the medoid v_i of a cluster. If the same element is selected more than once to serve as the medoid of different clusters it signifies an effective merging of those clusters. It is uncommon for the algorithm to select the same element *k'* more than once to serve as a medoid unless the data has a distribution that would lead to a number of natural clusters that is less than the user prescribed number of clusters *C*. The algorithm may also exhibit this behaviour if the user has selected a dissimilarity measure with a poor discriminating ability. In the event that the algorithm violates constrain equation 23, the user should adjust the number of clusters *C* or choose a more appropriate dissimilarity measure.

4.3.6 Silhouette Analysis

The Fuzzy C-Medoids model does not have any inbuilt mechanism for identifying the natural number of clusters that exist in a data set. The model depends on the user of the algorithm to specify the number of clusters *C* that are to be created. The quality of a solution based on the parameter *C* may be verified by a post cluster analysis using cluster silhouettes. Cluster silhouettes are a graphical aid in assessing the appropriate number of clusters in a dataset. The method was developed by Rousseeuw [80] and is documented here briefly due to its future use in determining an appropriate number of machine cells to be created in a Factory Configuration Template design.

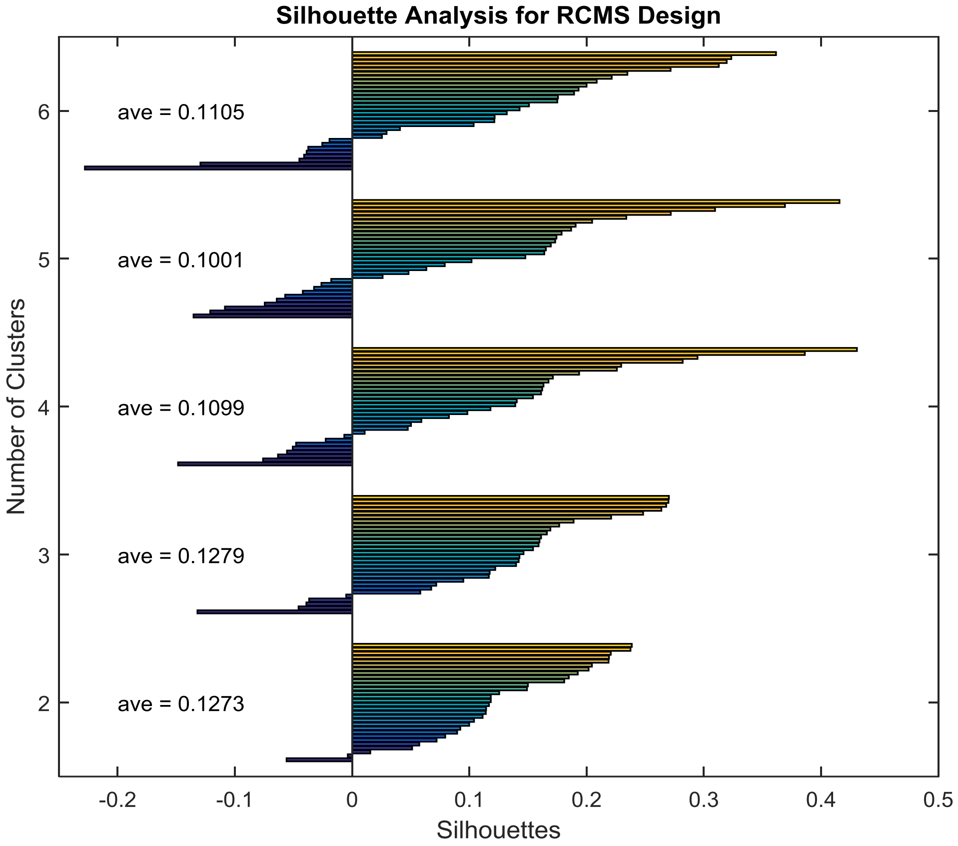

Figure 4.1: Sample Silhouette Analysis for a Part Family

Consider a data element *k*; the silhouette of *k* in the cluster it has been placed is calculated as follows:

$$s(k) = 1 - {a(k)}/{b(k)}, \qquad if\ a(k) < b(k) \tag{25}$$

$$s(k) = 0, \qquad if\ a(k) = b(k) \tag{26}$$

$$s(k) = {b(k)}/{a(k)} - 1, \qquad if a(k) > b(k) \tag{27}$$

From the above definition the silhouette values are restricted as follows: $-1 \leq s(k) \leq 1$. In these equations $a(k)$ is the average dissimilarity of member k with respect to other members of the same cluster and $b(k)$ is the lowest dissimilarity between the element k and another cluster.

The creation of a silhouette as defined by Rousseeuw [80] requires crisp clusters. In order to obtain crisp clusters from fuzzy clusters, elements may be assigned to the cluster in which it holds maximum membership. That is done by applying the following rule to the membership matrix: $\tilde{\mu}_{ik} = 1$ if $i = \arg\max_{1 \leq i \leq C} \{\mu_{ik}\}$ and $\tilde{\mu}_{ik} = 0$ otherwise.

The appropriate selection of the number of clusters *C* may then be verified by creating a silhouette plot. The features of a silhouette plot that are used to determine if a suitable number of clusters has been specified is the width of the silhouette, the average silhouette value and the presence of negative silhouettes. A sample silhouette plot is provided in Figure 4.1. The silhouettes were obtained for the clustering of parts into families. The part data that was used in the example may be found in Table A - 1, in Appendix A.1. The dissimilarity measure used in the clustering was Jaccard Dissimilarity (presented in section 4.4.1) and the algorithm was the Fuzzy-C Medoids algorithm.

When the similarity between data elements decreases the horizontal width of the silhouette becomes narrower. A poor match between a data element and its cluster is identified by a negative *s(k)*. Figure 4.1 shows values of the average *s(k)*, or average silhouette width for a given number of clusters. The average silhouette is calculated by the following equation:

$$ave = \frac{1}{N}\sum_{k=1}^{N} s(k) \tag{28}$$

If three part families are used (i.e. $C = 3$) the average silhouette width is 0.1279. Although this is the highest average, this set of clusters has a higher number of poorly clustered elements as indicated by the negative silhouettes when compared to the case where two families are used (i.e. $C = 2$). A manufacturer may opt for three part families where many parts are well clustered with some outlying cases. The manufacturer may opt for two part families where the similarities between parts are less strong yet with less significant outlying cases.

The choice of *C* may also depend on other factors such as the feasibility of creating two or three cells and part families; for example two cells and two part families may be more cost effective from the point of view of labour requirements. The manufacturer may opt for three cells and three part families as smaller cells and families may enable easier control of the system. The silhouette plot is therefore an additional tool that may be used in determining the number of cells that should be created on a FCT.

4.4 Selection of a Dissimilarity Measure

4.4.1 Available Binary Dissimilarity Measures

The data used for Group Analysis for CMS is conventionally represented in a machine-component incidence matrix. These matrices will also be used for FCT design in this research. The values of the matrix are binary (or zero-one values); where the value ‘1’ denotes that part *j* requires the services of machine *i* and ‘0’ otherwise. The Fuzzy C-Medoids model presented in section 4.3.3 must be used with a suitable binary distance or dissimilarity measure for *n*-dimensional binary data.

Equations 29- 35 list some of the binary distance/dissimilarity measures documented in a survey of binary similarity and distance measures presented by Choi et al [81]. Jaccard Dissimilarity was not included in the survey by Choi et al, however the Jaccard Similarity, presented in equation 1, has been known for its usefulness in grouping parts and machines. The Jaccard Similarity is rewritten here, in equation 35, to represent dissimilarity.

$$D_{MANHATTAN/HAMMING} = b + c \tag{29}$$

$$D_{MEAN\_MANHATTAN} = \frac{b + c}{a + b + c + d} \tag{30}$$

$$D_{VARI} = \frac{b + c}{4(a + b + c + d)} \tag{31}$$

$$D_{SIZE\_DIFFERENCE} = \frac{(b + c)^2}{(a + b + c + d)^2} \tag{32}$$

$$D_{SHAPE\_DIFFERENCE} = \frac{n(b + c) - (b - c)^2}{(a + b + c + d)^2} \tag{33}$$

$$D_{LANCE\&WILLIAMS} = \frac{b + c}{(2a + b + c)} \tag{34}$$

$$D_{JACCARD} = 1 - \frac{a}{a + b + c} \tag{35}$$

In these equations a is the number positive matches or instances '*1*' is present at the same index in both vector z_i and z_j ,b is the number of instances '*1*' is present in vector z_i and '*0*' is present in vector z_j at the same index, c is the number of instances '*0*' is present in vector z_i and '*1*' is present in vector z_j at the same index, d is the number of negative matches or instances '*0*' is present at the same index in both vector z_i and z_j and n is defined by equation 36:

$$n = a + b + c + d \tag{36}$$

The measures presented here are those identified to have the greatest potential to correlate the objectives of cell and part family formation with the general objective of clustering which is to minimize in-cluster dissimilarity.

Table 4.1: Machine-Part Incidence Matrix [23]

		PARTS										
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
MACHINES	**1**	1	0	1	0	0	0	1	0	0	0	1
	2	1	1	0	0	0	1	0	0	0	0	0
	3	0	1	0	0	0	1	0	0	1	0	0
	4	0	0	0	1	1	0	0	0	0	1	0
	5	0	0	1	0	0	0	1	0	0	0	0
	6	0	0	1	1	0	0	0	0	0	0	1
	7	0	0	0	0	1	0	0	1	0	1	0

As an example, consider the machine-component incidence matrix presented in Table 4.1, which was originally presented by Boctor [23]. The FCMdd algorithm requires a dissimilarity measure in order in order to perform fuzzy clustering. The algorithm can either generate fuzzy partitions for part families or machine cells; it cannot generate both simultaneously. If machine cells are to be identified, a machine-machine dissimilarity matrix must be created. Rows 1…7 are treated as the data elements and the information stored in the columns are treated as the “coordinates” of the element (data point) in each dimension. If part families are to be identified, a part-part dissimilarity matrix must be created. Columns 1…11 are treated as the data elements and the information stored in the rows columns are treated as the “coordinates” of the element (data point) in each dimension.

The part-part dissimilarity matrix shown in Table 4.2 was generated by applying the Lance & Williams dissimilarity measure. A sample calculation is presented in Appendix B.1. According to this measure, the dissimilarity between part 1 and part 2 is 0.5, etc.

Table 4.2 Part-Part Dissimilarity Matrix Based on the Lance & Williams Coefficient (P=Part)

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**
P1	0.0	0.5	0.6	1.0	1.0	0.5	0.5	1.0	1.0	1.0	0.5
P2	0.5	0.0	1.0	1.0	1.0	0.0	1.0	1.0	0.3	1.0	1.0
P3	0.6	1.0	0.0	0.6	1.0	1.0	0.2	1.0	1.0	1.0	0.2
P4	1.0	1.0	0.6	0.0	0.5	1.0	1.0	1.0	1.0	0.5	0.5
P5	1.0	1.0	1.0	0.5	0.0	1.0	1.0	0.3	1.0	0.0	1.0
P6	0.5	0.0	1.0	1.0	1.0	0.0	1.0	1.0	0.3	1.0	1.0
P7	0.5	1.0	0.2	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.5
P8	1.0	1.0	1.0	1.0	0.3	1.0	1.0	0.0	1.0	0.3	1.0
P9	1.0	0.3	1.0	1.0	1.0	0.3	1.0	1.0	0.0	1.0	1.0
P10	1.0	1.0	1.0	0.5	0.0	1.0	1.0	0.3	1.0	0.0	1.0
P11	0.5	1.0	0.2	0.5	1.0	1.0	0.5	1.0	1.0	1.0	0.0

4.4.2 Testing of Dissimilarity Measures with Fuzzy C-Medoids

The use of an appropriate dissimilarity measure is essential to align the objectives machine cell and part family identification, with that of fuzzy clustering. A measure with a poor discriminating ability can be identified by a large number of ‘ties’ in the membership matrix. Let g_k be the number of clusters for which the maximum membership value for element k is shared. The number of crisp clusters M that may be derived from the fuzzy partitioned membership matrix, by allocating an element to a cluster of maximum membership, is given by equation 37:

$$M = \prod_{k=1}^{N} g_k \tag{37}$$

The binary dissimilarity measures presented in the previous section were tested for their discriminating ability when used with the FCMdd algorithm. The measures where applied to a list of well-known Group Analysis problems; these are listed in Table 4.3. The corresponding machine-component matrices are presented in Appendix A.2.

Table 4.3: List of Problems for Testing of Dissimilarity Measures

Problem	Source	Machines	Parts	Cluster
1	McAuley [28]	4	6	Machines
2	McAuley [28]	4	6	Parts
3	Chandrasekharan &Rajagopalan [22]	8	20	Machines
4	Chandrasekharan &Rajagopalan [22]	8	20	Parts
5	Boctor [23]	16	30	Machines
6	Boctor [23]	16	30	Parts
7	King[18]	16	43	Machines
8	King[18]	16	43	Parts
9	King & Nakornchai [37]	16	43	Machines
10	King & Nakornchai [37]	16	43	Parts

The problems were tested by clustering of parts and clustering of machines in order to highlight any difference in the number of possible crisp partitions *M*. The results are shown in Table 4.4.

Table 4.4: Binary Dissimilarity Measures – Growth in M

Problem	1	2	3	4	5	6	7	8	9	10
Manhattan	1	1	1	16	2	1	32	8	48	2.99×10^{6}
Mean Manhattan	1	1	1	16	2	1	32	2.99×10^{6}	48	2.99×10^{6}
Vari	1	1	1	16	2	1	32	8	48	2.99×10^{6}
Size Difference	1	1	1	16	2	1	16	3.11×10^{4}	32	3.11×10^{4}
Shape Difference	1	1	1	16	2	4.19×10^{6}	4	6561	4	6561
Lance & Williams	1	1	1	1	1	1	1	16	1	16
Jaccard	1	1	1	16	1	2	1	16	1	16

The results indicate that all binary dissimilarity measures display adequate discriminating ability for relatively small problems (problems 1 – 4). The Manhattan/Hamming, Mean Manhattan, Vari, Shape Difference and Size Difference measures do not perform well on larger problems, i.e. problems with more than 8 machines and 20 parts. This is indicated by the extreme increase in the number of crisp clusters that may be derived from the resulting fuzzy partitioning matrix. All dissimilarity measures display a larger value for *M* when the FCMdd algorithm is applied to the fuzzy clustering of parts when compared to the fuzzy clustering of machines. This is an expected result as there are usually more part types than machine types in a manufacturing system. The Lance & Williams and Jaccard dissimilarity measures displayed consistently good discriminating ability, with no drastic increases in the value of *M* with respect to increasing problem size. The Jaccard and Lance & Williams measures were used for further development.

4.5 A Simultaneous Fuzzy Clustering Heuristic

The outcome of the fuzzy clustering process or in this case the FCMdd algorithm is the $C \times N$ membership matrix and a set of medoids. These outputs may be obtained for either machine cells or part families and not both at the same time. In this section a heuristic is developed that simultaneously identifies multiple sites for machines in cells and creates part families.

4.5.1 Architecture of a Simultaneous Clustering Heuristic

The coupling between parts and machines via a machine-component incidence matrix creates the opportunity for clusters of machine sites to be created from a set of part families. It is therefore possible that secondary clusters (machine sites) can be derived from a primary clusters (part families) when a relational mapping is available. The following definitions are relevant:

Primary Cluster: the clusters for which medoids and a membership matrix were originally generated by the FCMdd algorithm.
Secondary Clusters: the clusters generated from a relational mapping to the primary cluster.

In order to improve the match between primary clusters and secondary clusters it is possible to set-up an iterative procedure by using the secondary clusters to reformulate the primary clusters until no changes occur. In fuzzy clustering, the calculation of a membership matrix depends on cluster prototypes. When cluster prototypes change, the membership matrix changes and the crisp cluster derived from the membership matrix changes. The following definition is relevant:

Primary Prototypes: the prototypes used to compute the membership matrix of primary clusters.

A method has been identified for updating primary prototypes from secondary clusters; this is basis for the Simultaneous Fuzzy Clustering (SFC) heuristic proposed here. The architecture of the algorithm is displayed in Figure 4.2.

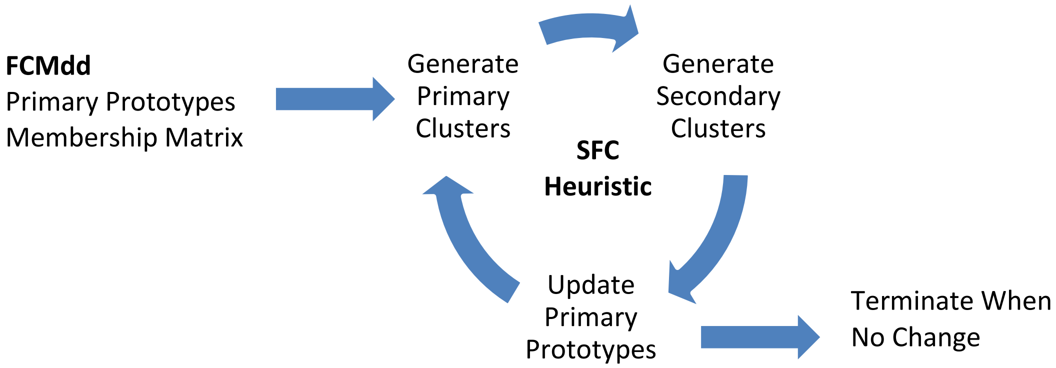

Figure 4.2 Architecture of the SFC heuristic

The primary prototypes (medoids) are initially generated by the FCMdd algorithm. The primary prototypes and membership matrix are used to create primary clusters. The SFC heuristic forms secondary clusters from primary clusters; secondary clusters are then used to create updated primary prototypes. Updated primary prototypes allows the reformulation of primary clusters. The SFC heuristic terminates when there is no change to the updated primary prototypes.

Figure 4.2 shows that the FCMdd algorithm serves the single purpose of providing an initial seed of primary cluster prototypes and a membership matrix as a seed to the SFC heuristic. The seed is determined by using part families as the primary cluster.

4.5.2 The SFC Heuristic

The SFC heuristic is presented in this section. It identifies sites for machines in cells on a FCT and determines part families simultaneously.

Simultaneous Fuzzy Clustering Heuristic

```
 [Medoids_FCMdd, U_FCMdd] = FCMdd(); // Generate seed using FCMdd algorithm
T = Matrix; // Machine-component incidence matrix
L = MinumumMembership; // Minumum membership value for a site to be created

it_count = 0; //Initialize iteration counter
ITMAX= MaxIterations; //Define maximum number of iterations

Repeat
          it_count=it_count+1; //Update iteration counter

          if it_count=1 //For the first iteration
                    // Set medoids generated by FCMdd as the primary prototypes
                    CurrentPrototypes = Medoids_FCMdd;
                    //Set U generated by FCMdd clustering as initial parts membership matrix
                    U_Parts = U_FCMdd;
          else //For every iteration after the first iteration
                    //Set prototypes generated in previous iteration as current prototypes
                    CurrentPrototypes = NewPrototypes;
                    //Calculate an updated membership matrix for parts using cluster prototypes
                    U_Parts = Generate_U_Parts (CurrentPrototypes);
          end

          //Generate part families (primary clusters) from membership matrix
          PartFamilies = ClusterParts (U_Parts);

          //Generate a membership matrix for machines using part family information
           U_Machines = Generate_U_Machines (PartFamilies, T);

          //Generate machine sites (secondary clusters) from membership matrix
           MachineSites = ClusterSites (U_Machines,L);

          //Generate updated primary medoids from secondary cluster information
           NewPrototypes = PrototypeGeneration (MachineSites);
```

```
Until (CurrentPrototypes = NewPrototypes or it_count=ITMAX )
[PartFamilies] = PostProcessing( PartFamilies)
```

The SFC heuristic requires a membership matrix and medoids as a seed; this seed is generated by the FCMdd algorithm. Without a good initial seed the SFC heuristic may fail to find the best solution with high repeatability, particularly for larger problems. The primary cluster in the SFC heuristic are part families and the secondary clusters are machine sites. An additional input to the algorithm is the machine-component incidence matrix. This matrix is required to calculate a membership matrix for the secondary cluster; in this instance machine cells. The parameter *L* must also be defined by the user, this parameter is discussed in section 4.5.2.3.

The algorithm requires the definition of an iteration counter and a maximum iteration count during initialization. This is defined because the algorithm has been tested and found to cycle indefinitely when a dissimilarity measure has a poor discriminating ability.

The first iteration of the repeating loop takes the medoids and membership matrix generated by the FCMdd algorithm and uses it to form part families. Part family formation is achieved by the *ClusterParts()* function. The rule implemented in this function is presented in section 4.5.2.1.

The algorithm proceeds to create a membership matrix for machines in the *Generate_U_Machines()* function. This ‘secondary’ membership matrix is not formed by the Fuzzy C-Means membership model; this membership matrix is formed by an affinity analysis of each machine with respect to individual part families. The more parts requiring the service of machine *x* in a family, the greater the affinity of the machine to that family and the higher the membership value. The method for forming this secondary membership matrix is described in section 4.5.2.2. The membership matrix *U_Machines* is then used to allocate machine sites to cells using the *ClusterSites()* function. The rule implemented in this function is presented in section 4.5.2.3.

After machine sites have been allocated to cells, the heuristic proceeds to formulate a set of new primary prototypes and formulate a new membership matrix for part family formation. The *PrototypeGeneration()* function and *Generate_U_Parts()* function is discussed further in section 4.5.2.4.

The process of forming primary and secondary clusters continues until the primary prototypes of the previous iteration match the updated primary prototypes of the current iteration, or the maximum iteration count is reached. The outputs of the algorithm are the clusters (machine sites and families) that were formed in the same iteration in which the algorithm was terminated. The resulting part families are then post processed to remove any duplicate members that may still exist as a result of the cluster formation rule for parts. The *PostProcessing()* function is presented in section 4.5.2.5.

The SFC heuristic is an original innovation of this research. The FCMdd algorithm and SFC heuristic were implemented in the Matlab® program development environment. The program code may be viewed on the accompanying CD.

4.5.2.1 Cluster Formation for Parts

The *ClusterParts()* function implements the crisp cluster formation rule: $\tilde{\mu}_{ik} = 1$ if $i = \arg\max_{1 \le i \le C} \{\mu_{ik}\}$ and $\tilde{\mu}_{ik} = 0$ otherwise.

In instances where the maximum membership value of a part is shared by more than one family, the part is allocated to all these families to allow the heuristic to more freely explore part family options. This is the only scenario under which parts are duplicated in families. An example application of part family formation is provided in Appendix B.2.

4.5.2.2 Affinity Analysis for Machine Membership Matrix Formation

The process of developing a membership matrix for machines is not based on the Fuzzy C-Means membership model, it is formulated by analysing the affinity of each machine with respect to part families. The affinity matrix is a $C \times N$ matrix where C is the number of cells that have been prescribed and N is the number of machines:

$$A = [a_{ik}], \qquad 1 \le i \le C, \qquad 1 \le k \le N \tag{38}$$

Here a_{ik} is the number of parts in cluster (family) i that require processing on machine k. The matrix A is populated from part family information and process routing data for parts stored in the machine-component incidence matrix. The elements of the membership matrix *U_Machines* are computed by equation 39:

$$\mu_{ik} = \frac{a_{ik}}{\sum_{j=1}^{C} a_{jk}} \tag{39}$$

An example calculation corresponding to the *Generate_U_Machines()* function is presented in Appendix B.3.

4.5.2.3 Cluster Formation for Machine Sites

The *ClusterMachines()* function implements the cluster formation rule: $\tilde{\mu}_{ik} = 1$ if $\mu_{ik} \ge L$ and $\tilde{\mu}_{ik} = 0$ otherwise. Here L is the minimum membership value of a machine with respect to a cell, which justifies the creation of a site for the machine in the cell. The manner in which μ_{ik} is calculated for machines makes it an indicator of the relative part load that each cell will impose on a machine. L can therefore be thought of as a threshold of relative part load that justifies the creation of a site for the machine in a cell. An example of the allocation of sites for machines in cells is presented in Appendix B.4.

4.5.2.4 New Prototype Generation and Updating the Part Membership Matrix

Updated part family prototypes are generated in each iteration of the SFC heuristic. These primary prototypes are generated by exploiting the relationship between primary prototypes and secondary clusters. Generated prototypes are used to create an updated membership matrix for parts. This is done by using equation 19. Examples for the *PrototypeGeneration()* function and *Generate_U_Parts()* functions are presented in Appendix B.5.

4.5.2.5 The Post Processing Function

The *PostProcessing()* function removes duplicate parts from families. Recall that duplicate parts are allowed to exist in families when maximum membership values are tied according to the cluster formation rule. The criteria used for removing duplicate parts from families was to remove the part from the family where the corresponding cell had the least available processes. If this criteria could not determine an appropriate family, the part was allowed to remain in the family with the least members.

4.6 Example - Site Allocation on a Factory Configuration Template

The machine-component matrix of Table A - 1 was used to simultaneously identify machines sites within cells and part families. The SFC heuristic was used with the Jaccard Dissimilarity measure. Three cells were chosen for implementation on a FCT; the silhouette analysis presented in section 4.3.6 is relevant.

If the parameter $L = 0.4$, the Clusters of Sites (*CSi*) and Part families (*PFi*) are:

CS1 = {S2, S3, S8, S9, S12, S15, S17};
CS2 = {S1, S4, S5, S6, S13, S17};
CS3 = {S7, S10 , S11, S14, S16, S18};
PF1 = {3, 12, 13, 14, 15, 16, 17, 18, 19, 21, 26};
PF2 = {2, 4, 5, 6, 7, 9, 10, 11};
PF3 = {1, 8, 20, 22, 23, 24, 25, 27, 28, 29, 30}.

Here *CS1* denotes the cluster of sites allocated to the first cell, etc. The prefix '*S*' denotes a site allocated to a cell for the corresponding machine number. For example, machine 2 has been allocated a site in the first cell. Based on $L = 0.4$ each machine has been allocated exactly one site that it may occupy in each cell with the exception of machine 17 which has a site in the first and second cells.

If $L = 0.2$, the clusters of sites and families are:

CS1 = {S2, S3, S5, S8, S9, S12, S13, S15, S17};

CS2 = {S1, S2, S4, S5, S6, S7, S8, S13, S17};

CS3 = {S7, S9, S10, S11, S12, S14, S16, S18};

PF1 = {3, 12, 13, 14, 16, 17, 18, 19, 21, 26};

PF2 = {2, 4, 5, 6, 7, 9, 10, 11, 15};

PF3 = {1, 8, 20, 22, 23, 24, 25, 27, 28, 29, 30}.

Based on *L = 0.2* machines 2, 5, 7, 8, 9, 12, 13, 17 have been allocated two sites each. For example machine 9 has a sites available for occupation in the first and third cells. Notice that the part family clusters have changed; part 15 is now in the second family.

Note that for the problems in this research the FCMdd algorithm and SFC heuristic were able to provide results rapidly, requiring a maximum of six iterations each.

4.7 Selection of Parameter *L* in FCT Creation

The parameter *L* is the relative part load on a machine with respect to a cell. The parameter may be selected based on the number of sites that are feasible for a given factory floor space. In the example of the previous section, 19 sites were generated across all cells for *L=0.4* and 26 sites were generated for *L=0.2*. The parameter *L* may be adjusted until an acceptable number of sites are generated. Note that *L* should never bet set very high in the SFC heuristic as it may lead to no sites being created from some machines. It is recommended that the parameter *L* be varied in the range of 0-0.5.

An alternative method of evaluating a solution is to assume that each site is occupied by a machine. This allows the creation of a machine-component incidence matrix where block diagonal structures can be identified. As an example, if every site is occupied by a machine in the solution corresponding to *L=0.2*, the resultant Machine Cells (*MCi*) will be: *MC1 = {2, 3, 5, 8, 9, 12, 13, 15, 17}*, *MC2 = {1, 2, 4, 5, 6, 7, 8, 13, 17}*, *MC3 = {7, 9, 10, 11, 12, 14, 16, 18}*. The corresponding block diagonalised matrix is presented in Table A - 2, Appendix A.1. From Table A - 2 is clear that there are 7 exceptional elements are outside of the block structures and 163 voids are within the block structures. Recall that exceptional elements are indicators of inter-cell part movement and voids are indicators of machine underutilization in a fixed CMS.

In the operation of a RCMS every site will not be occupied at all times. When sites are empty the number of part types performing inter-cell part movements can only increase in order to access the necessary machine at another site. The exceptional elements obtained here are therefore a lower bound on the minimum number of part types requiring inter-cell material handling movements at any time. It is desirable to make the lower bound as low as possible for a greater opportunity to eliminate inter-cell part moves by re-layout of machines. The re-layout of machines effectively reconfigures machine clusters.

The void count is the total number of instances for which parts will not use a machine allocated to a resident cell. An increasing void count, in the context of a fixed CMS indicates the increasing likelihood that machines will have low utilization. In the operation of a RCMS not every site will be occupied at all times. The void count, in the context of RCMS indicates the increasing likelihood that machine sites will have low utilization. The creation of additional sites for machines involves an investment risk since the creation of sites requires the outlay of finances for a larger factory floor space and the installation of infrastructure to support the machine that will be hosted.

In summary, it desirable to reduce exceptional elements to increase opportunities for eliminating material handling costs and it is desirable to reduce voids to reduce the risk of investing in sites that may be underutilized. Naturally, an increase in opportunity is accompanied by an increase in risk. Figure 4.3 shows that reducing the value *L* in the SFC heuristic reduces the number of exceptional elements. Figure 4.4 shows that reducing the value *L* in the SFC heuristic increases the number of voids.

In this example, at *L = 0.15*, the number of exceptional elements is zero, however, this is accompanied by a sharp increase to over 250 voids. This is an increase of more than 100 voids for the benefit of 7 less exceptional elements when compared to *L = 0.2*. In the example problem the solution generated for *L = 0.2* represents the maximum reasonable opportunity-risk combination that should be adopted.

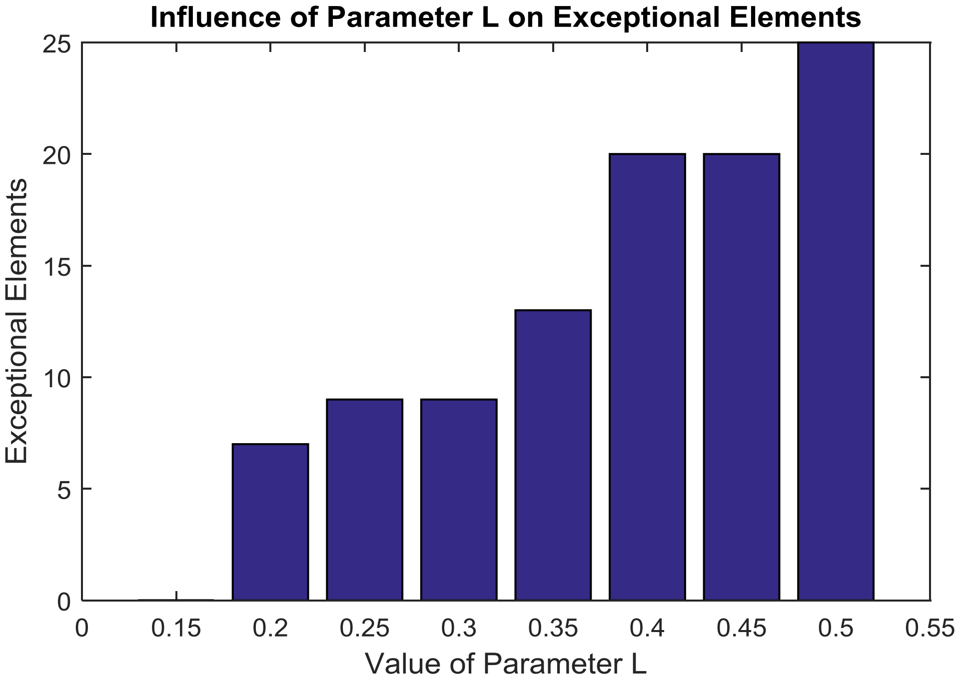

Figure 4.3: Influence of *L* on Exceptional Elements for the Example Problem

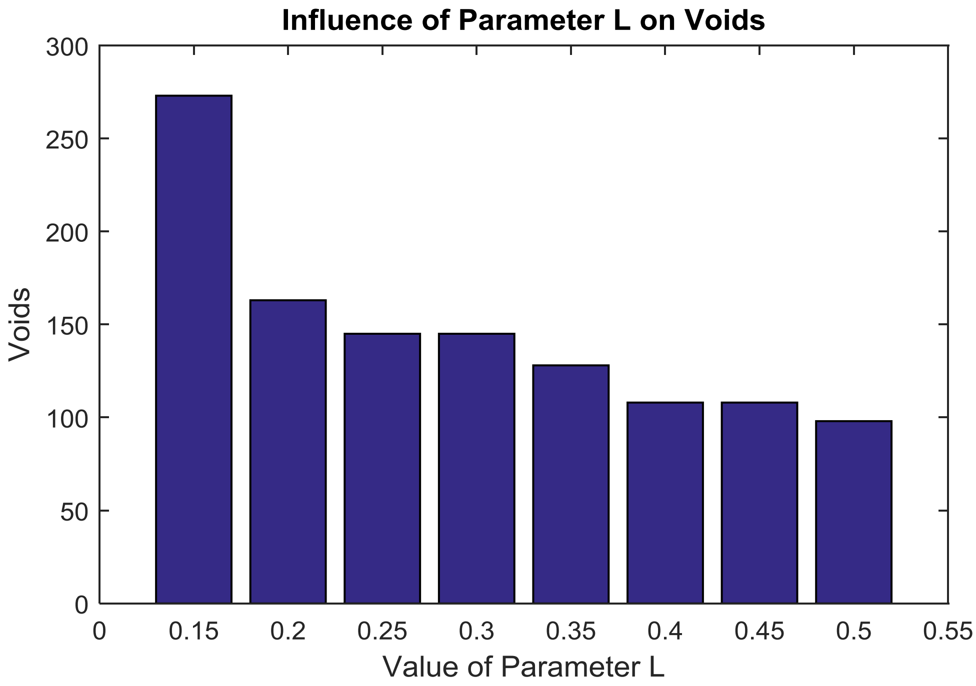

Figure 4.4: Influence of *L* on Voids for the Example Problem

4.8 Chapter Summary

In this chapter the concepts from fuzzy clustering theory were identified as useful for generating sites for machines in cells on a Factory Configuration Template. The concept was built into a Simultaneous Fuzzy Clustering heuristic that generates sites for machines in cells and part families simultaneously.

The SFC heuristic generates crisp clusters based on rules that are applied to a fuzzy partitioned membership matrix. In the heuristic primary clusters are used to generate secondary clusters and secondary clusters are then used to reformulate the primary clusters to obtain well matched solutions.

The membership concept from fuzzy clustering was used to identify if a site should be created for a machine in each cell based on the relative part load parameter *L*. The chapter concluded with a discussion on selecting an appropriate solution for various options of *L*.

5 Capacity Analysis and Site Arrangement in Manufacturing Cells

5.1 Chapter Introduction

The SFC heuristic presented in chapter 4 identified sites that should be created for machines in cells in order to impart reconfigurability to a CMS. This reconfigurability exists within the provisions allowed for re-layout by a Factory Configuration Template. The previous chapter assigned machine sites to cells, however, these sites must be arranged in a manner that supports unidirectional flow. The problems associated with a lack of planning for unidirectional flow in future cell configurations was discussed in section 2.8.

This chapter addresses the subject of site arrangements within cells to support unidirectional flow, in current and future manufacturing system configurations. The focus is on intra-cell site arrangement and not the arrangement of cells in relation to each other. The inter-cell layout design deals with the location of cells relative to each other. The location and orientation of cells with respect to each other may be studied as an optimization problem, however, it is not mandatory for demonstrating the use of a FCT and other methods developed in this research.

A factor that must be considered when developing an intra-cell cell layout is the possibility of duplicate machines being required in series or parallel along the path of flow within a cell. Duplicate machines may be required in parallel in order to improve the production capacity of a cell. A duplicate machine placed in series may eliminate the backward flow of parts. These matters will be taken under consideration in this chapter as intra-cell design methods are developed for unidirectional flow.

5.2 Capacity Considerations in Duplicate Site Identification

The SFC heuristic developed in section 4.5 has the ability to allocate a maximum of one site per machine per cell. The heuristic cannot allocate additional sites because it is based on the information provided in a zero-one machine component incidence matrix without consideration for machine capacity requirements. The output of the SFC heuristic must be combined with capacity verification procedures to ensure that the FCT has an appropriate number of sites at a system and cell level.

5.2.1 System Level Site Analysis

The minimum number of machines required by a system must be calculated in order to ensure that the capacity is sufficient to meet the demand for parts. The system level requirement for machines is different from the cell level requirement for machines and both types of requirements must be considered. The cell level requirement for machines is always greater than the system level requirement of machines. For example, at a system level, the total part load may dictate that 2 machines be made available; however, Cell 1 might require 1.2 machines and Cell 2 might require 0.8 machines. Since a faction of a machine cannot be allocated to a cell, Cell 1 effectively requires 2 machines and Cell 2 requires 1 machine. The total machine requirement at a cellular level is therefore 3 machines.

Machines are usually made available based on the system level requirement. These are the mandatory minimum machine numbers necessary to create sufficient production capacity. If additional capital is available a machine may be purchased to satisfy the cell level requirement. If no additional machine is made available, some parts may have to perform inter-cell movements.

When designing a FCT, sufficient sites must be made available across all cells in order to host the mandatory minimum number of machines. Moreover, the mandatory number of machines must be determined for each period (i.e. year 1, year 2, etc.) to ensure that the FCT is appropriate for the entire planning horizon. Depending on the parameter *L*, the SFC heuristic will create sites for the same machine type across multiple cells, however, a check must be performed to ensure that sufficient sites are made available.

The following steps may be followed in order to calculate the minimum mandatory number of machines to be made available in each period:

1. Determine the number of hours of processing time required on machine *i*, for each period *k*. This is calculated by summing the product of the anticipated volume of each part *p* and the cycle time of part *p* on machine *i*:

$$H_{ik} = \frac{\sum_{p=1}^{P} volume_{pk} \times minutes_{ip}}{60} \tag{40}$$

2. Determine the number of working hours available on machine *i* for each period *k*:

$$W_k = (hours/day) \times days \tag{41}$$

3. Determine the number of machines of type *i* required by the system in period *k*:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} \tag{42}$$

If $N_{ik} > 1$ for any of the periods *k*, a check must be performed to ensure sufficient sites have been made available for machine *i*. Note that $i \epsilon I$, where *I* is the set of all machine types, $p \epsilon \{1,\ldots,P\}$ is the set of all part types and $k \epsilon K$, where *K* is the set of periods (years, months, etc.) that are relevant. Machine availability is represented as a percentage; usually 70 % - 80 %.

Note that the manner in which the necessary number of machines does not consider the actual rate at which parts are capable of being produced by cells as dictated by the theory of constraints or the assignment of operators to cells. Nonetheless, this is the prescribed method for determining mandatory numbers of machines according to the *Handbook of Cellular manufacturing Systems* [82].

5.2.2 Example - Analysis of System Level Site Requirements

Assume that the following set of machine sites and part families have been determined by the SFC heuristic:

CS1 = {S1, S2, S3, S4};
CS2 = {S2, S4, S5, S6};
PF1 = {1, 2, 3, 4, 5};
PF2 = {6, 7, 8, 9, 10}.

Table 5.1 presents the anticipated annual part demand for parts 1-10. Table 5.2 presents the cycle time for each part on machine type 2; parts 4, 5, 9 and 10 do not use machine 2.

Table 5.1: Anticipated Annual Part Demand in Thousands (P=Part, Y=Year)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Y1	6	6	4	3	0	0	0	2	9	9
Y2	6	4	6	7	0	2	2	2	7	8
Y3	0	0	0	11	9	7	6	2	5	3
Y4	0	0	0	10	10	6	8	2	0	1

Table 5.2: Cycle Times for Parts on Machine 2 (P=Part, M=Machine)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
M2	30	25	26	-	-	27	28	28	-	-

Assuming a 350 day work year with 16 hours of working time per day; the total annual operating time is W_k = 5600 *hours/year*. In order to determine if a sufficient number of sites have been proposed for machine 2 the total hours of time required on the machine by parts must be calculated. The results are presented in Table 5.3 and a sample calculation is presented in Appendix C.1.

Table 5.3: Hours of Operating Time Required (Y=Year, M=Machine)

	Y1	Y2	Y3	Y4
M2	8167	10033	6883	7367

Assuming that machine availability is 70 %, the total number of machine type 2 required per year was calculated and is shown Table 5.4. A sample calculation is presented in Appendix C.2.

Table 5.4: Numbers of Machines Required Per Year (Y=Year, M=Machine)

	Y1	Y2	Y3	Y4
M2	3	3	2	2

The results of Table 5.4 reveal that three machines of type 2 are required. Although two machines are required in year 3 and 4, it is likely that three machines will be available due to purchases in the previous years. A comparison of this result with the number of sites proposed reveals that one additional site is needed for machine type 2, this could be created in the first or second cell. The choice requires an analysis of the cell level machine requirements.

The example here used machine 2 as the subject to demonstrate the importance of analysing the number of sites made available for machines throughout the system. This analysis must be performed for all machine types and tables such as Table 5.3 and Table 5.4 should be populated with the results of all machines.

5.2.3 Cell Level Site Analysis

Cell level site requirements must be analysed for each machine type to identify if a second site is required in the same cell. This analysis should be conducted even if the system level site analysis indicates that a sufficient number of sites are available. As a general rule, sites for machines should only be duplicated in cells where there is already one site.

Additional sites for machines, as dictated by capacity requirements usually implies that the extra machines will be placed in parallel along the path of flow within the cell. Adding machines in parallel is not supported by all cell shapes; the circular layout presented in Figure 2.2 does not lend itself to the placement of machines in parallel. A multi-row layout, as shown in Figure 5.1 supports the addition of machines in parallel. Multi-row layouts are characteristic of higher volume production systems and are not typical of a CMS. Nonetheless, the need may arise where machines in parallel are needed.

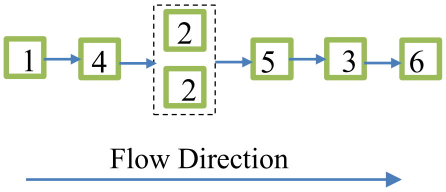

Figure 5.1: A Multi-Row Layout

If a multi-row layout is implemented with identical machines in parallel, methods for obtaining unidirectional flow as in a single row layout may still be applied. The duplicate machines are treated as a single entity as shown in Figure 5.1. An example will be presented that analyses if a cell requires machines (machine sites) in parallel.

5.2.4 Example – Analysis of Cell Level Site Requirements

In the example of section 5.2.2 it was shown that a minimum of three sites for machine 2 were required. Since the CFT consists of two cells, the creation of an additional site for machine 2 in one of the cells is required. In this section the cell level site requirements for machines will be analysed using the information provided in section 5.2.2.

Table 5.5 presents the calculations for the number of operating hours required in Cell 1 and Cell 2 on an annual basis. The calculations were performed using equations 40-42, using the number of parts in the corresponding family for the calculation, i.e. not all parts in the system. A sample calculation is shown in Appendix C.3.

Table 5.5: Hours of Operating Time per Cell for Machine Type 2 (Y=Year)

Cell 1				Cell 2			
Y1	**Y2**	**Y3**	**Y4**	**Y1**	**Y2**	**Y3**	**Y4**
7233	7267	0	0	933	2767	6883	7367

Assuming that machine availability is 70 %, the total number of machine type 2 required by cells per year has been calculated and is shown Table 5.6. A sample calculation is presented in Appendix C.4.

Table 5.6: Numbers of Machines Required per Cell per Year (Y=Year)

Cell 1				Cell 2			
Y1	**Y2**	**Y3**	**Y4**	**Y1**	**Y2**	**Y3**	**Y4**
2	2	0	0	1	1	2	2

The system level site requirements are three sites for three essential copies of machine 2. Table 5.6 reveals that there is merit in creating a duplicate site in Cell 1 and a duplicate site in Cell 2, i.e. a total of four sites in the system. If four sites are implemented the set of machine sites will be: *CS1 = {S1, S2, S2 S3, S4}* and *CS2 = {S2, S2, S4, S5, S6}*. If four sites are implemented with three machines, one site will be empty at all times; nonetheless, each site will be occupied at some point in time.

The example here used machine 2 as the subject to demonstrate the importance of analysing cell level site requirements. This analysis must be performed for all machine types. In situations where the system level analysis does not require additional sites, the cell level requirements must be analysed as opportunities may exist for cells to exchange machines and gain an inter-cell material handling reduction.

5.3 Site Arrangement in Cells

The capacity analysis of cells and determining if duplicate machine sites are required is a procedure that must be performed to determine if the cell layout shape is appropriate. Recall that cell shapes such as the circular layout do not allow machines to be placed in parallel while a multi-row layout can easily facilitate duplicate machine sites in parallel.

Once suitable cell layout shapes have been established the arrangement of sites within the layout may be optimised for unidirectional flow. In this research, if a multi-row layout has been adopted, the assumption is that duplicate machines in parallel are treated as a single entity as shown in Figure 5.1. This allows the multi-row layout to be treated as a single row layout when optimising the sequence of machine sites in cells.

5.3.1 Unidirectional Flow Optimization Model

In this research, unidirectional flow will be optimised using the data in a From-To matrix. The optimization model is as follows

The following notation is applicable to the model:

Sets:

$I = \{ 1, 2, \ldots, M\}$ is the index set, corresponding to machine types; i denotes a machine type and $i \in I$, note that $j \in I$ is a second index of the set of machine types.

$L = \{1,2,\ldots,N\}$ is the set of locations available in a linear sequence for each machine site; for example if the set of sites to be arranges are $\{S1,S3,S5,S7,S9\}$ then $L = \{1,2,3,4,5\}$. Index $\alpha \in L$ and $\beta \in L$.

Decision variables:

$x_{i\alpha}$ is a zero-one assignment variable indicating the assignment of the site corresponding to machine i to location α

$x_{i\beta}$ is a zero-one assignment variable indicating the assignment of the site corresponding to machine j to location β

Constants:

u_{ij} indicates instances of part flow between type machine i and type j according to a From-To matrix.

Note that the purpose of using two indices for the set of machines and two indices for the set of possible site locations is to enable an easier understanding and representation of the problem as mathematical program. Consider the expression:

$$x_{i\alpha} x_{j\beta} u_{ij} \tag{43}$$

If this expression is greater than zero, it indicates that there are u_{ij} instances of part flow between machine i and j, while the machines are in locations α and β respectively. Since α and β are indices referring to the same sequence of locations; if $\beta>\alpha$, this indicates that the flow is proceeding forward according to the sequence; e.g. flow from machine i in location $\alpha=1$ to machine j in location $\beta=3$ is considered flow in a forward direction. If $\alpha>\beta$ this indicates that the parts flow backward, i.e. part backtracking against the desired direction of intra-cell flow.

To minimize instances of back flow, the optimization problem can be written as

$$min \sum_{i=1}^{M} \sum_{j \neq i}^{M} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{\alpha-1} x_{i\alpha} x_{j\beta} u_{ij} \tag{44}$$

Subject to:

$$x_{i\alpha} = x_{j\beta}, \qquad \forall (i, \alpha) = (j, \beta) \tag{45}$$

$$\sum_{i=1}^{M} x_{i\alpha} = 1, \qquad \forall \alpha = 1, \dots, N \tag{46}$$

$$\sum_{\alpha=1}^{N} x_{i\alpha} = 1, \qquad \forall i = 1, \dots, M \tag{47}$$

$$x_{i\alpha} \in \{0,1\}; \; x_{j\beta} \in \{0,1\} \tag{48}$$

The first constraint requires that $x_{j\beta} = x_{i\alpha}$ for all $(j, \beta) = (i, \alpha)$; therefore $x_{i\alpha}$ is the decision variable and any algorithm chosen to solve this problem need only manipulate values of $x_{i\alpha}$. The second constraint allows only one machine i to be placed in location α, i.e. two machines cannot be placed in the same position in a linear sequence. The third constraint allows machines (machine sites) to be assigned to only one location in a linear sequence. The final constraints restrict $x_{i\alpha}$ and $x_{i\beta}$ to zero-one variables.

5.3.2 Tabu Search Algorithm for Unidirectional Flow Optimization

The Tabu Search (TS) algorithm provides a way of solving the unidirectional flow optimization model without having to deal with some of the complex features of the model such as the quadratic objective function. TS was originally created and formalized by Glover [83-85]. It is a metaheuristic method that employs local search for the optimization of mathematical models.

General Tabu Search Algorithm

```
//Preliminary Definitions
it_count = 1; //Initialize iteration counter
ITMAX= MaxIterations; //Define maximum number of iterations
MAX_L = MaxLength; // Define a maximum size for the Tabu List
T=Ø; //Define an empty Tabu List
N = NumberCandidates; //Define number of candidate solutions to generate per iteration

//Initialization
$S_o$ = InitialSolution(); //
S = $S_o$; // Define an initial solution
$S^*$ = S; //Set S as the best known solution
```

```
$f^*$ = ObjectiveFunction(S); //Determine the performance of S
T= T+S; // Add solution S to the Tabu List
Repeat
        it_count=it_count+1; // Update Iteration Counter

        // Generate N candidate solutions in the neighbourhood of S
        {S'} = Neighbourhood(N,S);
        {S'}= {S'}- T; // Discard candidate solutions that appear in the Tabu List
        S"= argmin[ObjectiveFunction(S')]; //  Evaluate and Select Best Candidate
        S=S"; //Change Current Active Search Point to Best Candidate Point
        if f">f* //If the performance of best candidate surpasses that of the best known solution
                $S^*$=S";//Set best known solution to best candidate
                f* = f";// Save the best known objective value
        end

        T=T+S"; // Add the Best Candidate to the Tabu List
        if size(T) = MAX_L; //If the Tabu List has reached its maximum size
                T=RemoveOldest(T) ; // Remove the solution that is the oldest in the Tabu List
        end
Until (it_count=ITMAX)
```

The Tabu Search algorithm presented here requires the initialization of an iteration counter and the definition of a maximum iteration count. The algorithm requires that an empty Tabu List be defined at the outset and a maximum length for the list to be defined. The Tabu List is a list of solutions for tweaks have has been conducted in an effort to find improved solutions in the 'neighbourhood'. The term 'neighbourhood' implies that the tweaks are minor enough for a modified solution to still be considered very similar to the original solution

An initial solution is generated by the *InitialSolution()* function. Solution representation and initialization are discussed in section 5.3.2.1. The initial solution S_o is defined to be the active solution S on which tweaks will be conducted in order to find improved solutions. The initial solution is then set as the best known solution S^* and added to the Tabu List. Solutions in the Tabu List will not be explored again by performing teaks. Prior to adding the initial solution to the Tabu List, the objective function value is computed and stored. The computation of objective function values is discussed in section 5.3.2.2.

The TS algorithm tweaks a single active solution S in each iteration of the repeating loop. N candidate solutions are generated by performing tweaks to the solution S. The manner in which these tweaks are formed is discussed in section 5.3.2.3. Candidate solutions that match solutions in the Tabu List are immediately discarded. The remaining candidate solutions are evaluated for their best objective function values. The best performing candidate *S''* is then set as the active solution *S*. If the objective function value of the best candidate outperforms that of the best known solution, it is set as the best solution and the best objective function value is updated. The active solution is then added to the Tabu List and if the Tabu List has exceeded its length, the oldest solution in the list is removed. The process of continuously updating the active solution and generating candidate solutions continues until the iteration counter reaches the maximum count. The algorithm can also be terminated when f^* converges onto a prescribed value however, this option was not implemented here.

The TS algorithm as described here was implemented in the Matlab® program development environment. The program code may be viewed on the accompanying CD.

5.3.2.1 Solution Representation and Initialization

The solution to the flow problem is represented as an ordered set, e.g. $S=\{1,3,2,4,6,5,9,8,10\}$. This implies that the site for machine 2 is at the third position in the sequence, etc. The initial solution is be generated by the *InitialSolution()* function by performing a random permutation without repetition of indices $1,\dots,M$. The method of representing a solution allows only one machine site to occupy one position in the sequence; it therefore ensures that the second and third constraints (equations 46 and 47) are satisfied.

5.3.2.2 Objective Function Calculation

The representation of the solution as an ordered set is easily converted to $x_{i\alpha}$ and $x_{j\beta}$ format. If machine i is in location α according to the set, fix $x_{i\alpha} = 1$ and $x_{i\alpha} = 0$ for every other α. Set $x_{j\beta} = x_{i\alpha}$ for all $(j,\beta) = (i,\alpha)$. The objective value value is then calculated using equation 44.Note that the manner in which the set representation is converted in values for variables $x_{i\alpha}$ and $x_{j\beta}$ ensures that the first and last constraints (equations 45 and 48) are enforced.

5.3.2.3 Generation of Candidate Solutions

Candidate solutions in the neighbourhood of *S* are generated by the *Neighbourhood()* function as follows:

1. Create a duplicate of *S*.
2. If *K* is the cardinality of the set *S*, randomly select two positions in the interval 1 to *K*.
3. Swap the elements in *S* that are stored at the two positions.
4. Repeat steps 1-3 *N* times, to create *N* candidate solutions.

5.3.3 Sample Results

The Tabu Search algorithm was applied to the problem given in Table 2.3 and Table 2.4. For this problem the best sequence in which sites for machines should be arranged is given in Table 5.7 and a plot of the best objective value per iteration is shown in Figure 5.2

Table 5.7: TS Results – Best Arrangement of Machine Sites

Location	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cell *x*	S2	S3	S1	S4	S5	S7	S6	S8	S10	S9

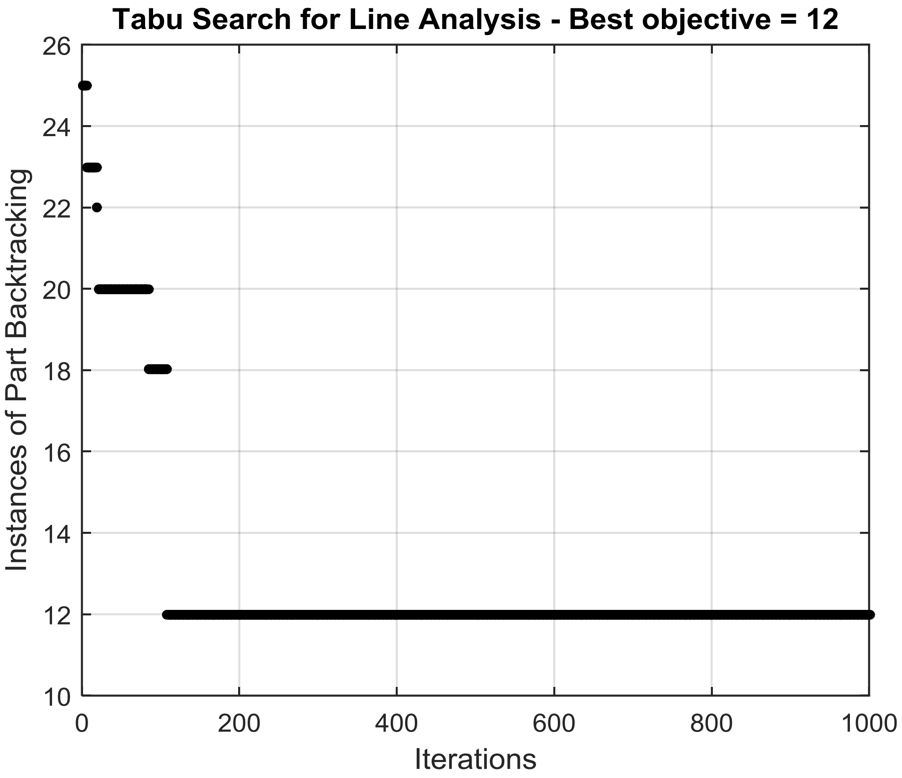

Figure 5.2 Tabu Search Algorithm – Best f^* per Iteration

Ten candidate solutions were generated per iteration and the length of the Tabu List was forty. Figure 5.2 shows that the number of instances of part backtracking moves could have been as high as 30, however, with the optimised sequence in which machine sites are arranged, the instances of part backtracking movements is 12. No further improvement was found although the search continued for 1000 iterations. Note that in a RCMS all the parts that were used in the analysis will not be active in the system at all times. The machines may not be at their respective sites at all times either. A part cannot backtrack to a machine that is not present. The backtracking count presented here is therefore represent an upper bound on the number of instances of part backtracking. It is desirable that the backflow in the worst case scenario be as low as possible.

5.4 Site Arrangement with Duplicate Sites in Series

When parts in the same family do not follow the same sequence of operations, the number of part backtracking movements is higher. The re-visitation of machines is another factor that may lead to a part backtracking to a machine that was visited previously. For example, a part may require a *face milling* operation on a milling machine. The same part may then visit other machines and then return to a milling machine for a *slotting* operation to be performed.

Backtracking part moves may be reduced by arrangement of machines (machine sites) in an optimised sequence; this was demonstrated in section 5.3.3. For a given set of parts and process sequences it may be impossible to reduce the amount of part backtracking any further without the introduction of additional machines in series along the path of flow. If machines are available further downstream according to the direction of flow, there will be no need for parts to backtrack. The following sections discuss the modification of the unidirectional flow optimization model in order to explore if an advantage can be gained by considering duplicate machines and machine sites in series.

5.4.1 Modifications to the Unidirectional Flow Optimization Model

The model presented in 5.3.1 does not explicitly consider multiple instances of the same machine type. If a maximum of ζ copies of each machine may be added to the system then constraint 47 must be modified as follows:

$$\sum_{\alpha=1}^{N} x_{i\alpha} \leq \zeta, \qquad \forall i = 1, \ldots, M \tag{49}$$

In the previous model, every machine type was to be allocated to one site and one site was to be allocated to a single machine. The modified constraint now allows each machine type to be allocated to more than one site. This allows exploration to determine if any benefit may derived from duplicate machines in series. If some benefit may be derived then there is value in creating duplicate sites in series to host the additional machines.

When there are multiple machines available, the challenge arises which requires determining which From-To pairs are valid for the objective function. Consider the definition of the objective function of equation 44. When duplicate sites are available, machine i may be available at $\alpha = 7$ and $\alpha = 6$, machine j might be available at $\beta = 3$ and $\beta = 1$. The coefficient u_{ij} obtained using the From-To matrix is no longer valid. The new coefficient $U_{i\alpha j\beta}$ defines the flow from machine type i in location α to machine type j in location β. A new objective function may is defined as:

$$min \sum_{i=1}^{M} \sum_{j \neq i}^{M} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{\alpha-1} x_{i\alpha} x_{j\beta} U_{i\alpha j\beta} \tag{50}$$

Where:

$$U_{i\alpha j\beta} = \sum_{p=1}^{TotalParts} f_p(i,\alpha,j,\beta), \quad for\ x_{i\alpha} = x_{j\beta} = 1 \tag{51}$$

$$U_{i\alpha j\beta} = 0, \qquad for\ x_{i\alpha} = 0\ or\ x_{j\beta} = 0 \tag{52}$$

Here $f_p(i,\alpha,j,\beta)$ is a function that determines the number of instances in which part p will move from machine i to machine j while i is in location α and j is in location β. This function will henceforth be called the Alpha-Beta Flow Function and will be discussed in the next section. The complete updated model is presented in Appendix D.1.

5.4.2 The Alpha-Beta Flow Function

The Alpha-Beta Flow Function f, is a function that that determines the number of instances in which part p will move from machine i to machine j while i is in location α and j is in location β. Prior to defining this function, other definitions must be considered.

Let P_p be the ordered set of processes required by part p. Recall that S' is an ordered set indicating the location of each machine type in a sequence in the TS algorithm. A Process-Machine Mapping Function g is defined as follows:

$$g: P_p \times S' \to Q_p \tag{53}$$

Here g defines how each processes of part p is mapped to the machines at the various positions as defined by S'. Here Q_p the ordered set of matched processes and machine locations, the elements of Q_p are of the form (i,α) and (j,β) . Because Q_p is an ordered set, each adjacent set of elements represents one From-To instance. The Alpha-Beta Flow Function operates on the set Q_p in order to determine the number of instances δ, for which part p will move from machine i to j while in locations α and β respectively:

$$f(i,\alpha,j,\beta): Q_p \to \delta \tag{54}$$

The working of functions g and f will be demonstrated with an example.

5.4.3 Example - Implementation of Function *f* and Function *g*

Consider the TS algorithm applied to solving the unidirectional flow optimization model; at an intermediate iteration the candidate solution $S' = \{1,2,3,4,5,3,2,4,1,2\}$. S' is an ordered set, which implies that machine type $i=1$ is available at the first and ninth locations, machine type $i=2$ is available at the second and tenth locations, etc. Part p requires processes 1-7 on machines as follows: $P_p = \{2,4,3,5,3,4,1\}$. P_p is an ordered set, which implies that the first process is on machine type $i=2$, the second process is on machine type $i=4$, etc.

The From-To relationships for this problem are more complex since there are multiple instances of the same machine type in S^*. Possible mapping of processes to the machines at various locations are listed in Table 5.8.

Table 5.8 Mapping of Machine Locations to Manufacturing Process of Part p

Process	1	2	3	4	5	6	7
Machine Type (i)	2	4	3	5	3	4	1
Location of Machines (α)	2,7,10	4,8	3,6	5	3,6	4,8	1,9

In this instance there are 96 possible ways in which the processes could be mapped to the machines at various locations (3×2×2×1×2×2×2=96). The challenge of setting up a Process-Machine Mapping Function g, is that the function should map processes to machines at the various locations in an optimal manner. The function g may be setup as a mathematical programming problem, however, the number of alternative mappings is relatively small for the application of a fully-fledged optimization model and solver; an enumerative approach was adopted instead. Function g operates as follows:

1. enumerate all possible process to machine mappings for machines available at various locations;
2. evaluate each alternative based on number of machines used and part backtracking moves;
3. select the mapping that has the highest machine utilization and lowest backtracking moves in this order of priority.

To illustrate the enumeration approach for implementation of function g, the 96 alternatives for the mapping of processes to machines in this example are presented in Table D - 1. The characteristics of each solution with respect to machine utilization and part backtracking movements are listed in Table D - 2 (see Appendix D.2). The best solution to the example problem is $Q_p = \{(2,2),(4,4),(3,3),(5,5),(3,6),(4,8),(1,9)\}$. The first element is $(i,\alpha)=(2,2)$, implying that the first process is performed on machine 2 which is in location 2. The second element is $(j,\beta)=(4,4)$, implying that the second process is performed on machine 4 which is in location 4, etc.

For the set Q_p, the Alpha-Beta Flow Function f is defined as follows:

$$f(i,\alpha,j,\beta) = 0 \;\; if \;\; (i,\alpha),(j,\beta) \notin Q_p \qquad (55)$$

$$f(i,\alpha,j,\beta) = \; \delta \; if \;\; (i,\alpha),(j,\beta) \in Q_p \qquad (56)$$

Here δ is the number of times the sequence $\{(i,\alpha),(j,\beta)\}$ appears in Q_p. For example, the sequence $\{(5,5),(3,6)\}$ appears once in Q_p, therefore $f(5,5,3,6) = 1$.

In the set Q_p there is only one instance of $\{(i,\alpha),(j,\beta)\}$ where $\beta > \alpha$, that is $\{(4,4),(3,3)\}$. The objective function of equation 50 sums flow in the instance where $\beta > \alpha$ therefore, for part p and solution S', $f(4,4,3,3) = 1$ is the contribution to the objective function.

5.4.4 Tabu Search Algorithm with Duplicate Machines

The generic structure of the Tabu Search algorithm, presented in section 5.3.2 is applicable to the solution of the unidirectional flow optimization model with duplicate machines in series. Modifications are only required to the following functions: *InitialSolution()*, *ObjectiveFunction()* and *Neighbourhood()*. The modifications are discussed in the subsections that follow.

5.4.4.1 Initialization Function

A candidate solution is represented as an ordered set, e.g. $S'=\{1,3,2,4,6,5,9,8,10\}$. The initial solution is be generated by the *InitialSolution()* function by performing a random permutation with repetition of machine indices $1,..,M$. The length of the permutation is $M+X$ where X is the number of additional machines in the sequence that may be explored to minimize part backtracking. For consistency with the constraint equations, the cardinality of the set of locations L must equal $M+X$. i.e. $|L|=M+X$. With respect to constraint equation 49, $\zeta=X+1$ for all machine types i, $i=1,\dots,M$.

5.4.4.2 Objective Function

The objective function is no longer calculated in the manner shown in section 5.3.2. Under a given solution S', the set Q_p must be created by the process-machine mapping function g for every part p. Alpha-Beta Flow Function f is applied to every Q_p to determine instances of part backtracking for machines located at various positions in the ordered set S'. For a given pair of machines and locations the instances of part backtracking are computed by equations 51 and 52. The total part backtracking for all pairs of machines and locations is calculated by equation 50.

5.4.4.3 Generation of Candidate Solutions

In order to optimize the sequence of machines and explore alternative duplicate machines, the *Neighbourhood()* function performs one of two possible operations to generate a new candidate:

Operation one: swap the position of two elements in the sequence S', as described in 5.3.2.3;
Operation two: replace duplicate elements in the sequence with new, randomly selected elements.

Each candidate S' is created by subjecting the active solution S to one of the two procedures, but not both at the same time. Tabu Search is meant to explore in the neighbourhood of the active solution S; applying both exploratory mechanisms is likely to create a drastically different candidate that may no longer be considered as located in the neighbourhood of S. Operation one is selected when random variable $x=1$ and operation two is selected when $x = 0$. The probability

$Pr(x=1) = p$ and probability of $Pr(x=0) = 1-p$, with the statistical distribution being a Bernoulli distribution.

To illustrate the working of operation two, consider the solution 6, 3, 7, **6**, 2, 1, 5, **1**, **2**, 4. A candidate solution is generated as follows:

1. Remove duplicate elements: 6, 3, 7, **[ ]**, 2, 1, 5, **[ ]**, **[ ]**, 4;
2. Randomly select elements in the range *1-M*, for as many duplicate elements that are required: e.g. 1,1,3;
3. Insert duplicate elements in empty positions: 6, 3, 7, [1], 2, 1, 5, [1], [3], 4.

5.4.5 Sample Results

The Tabu Search algorithm for unidirectional flow optimization with duplicate machines was applied to example problem presented in Table 2.3. The search took 19.43 seconds on a computer with an Intel® Core™ i7-4600 CPU operating at 2.7 GHz with 8.00 GB of ram.

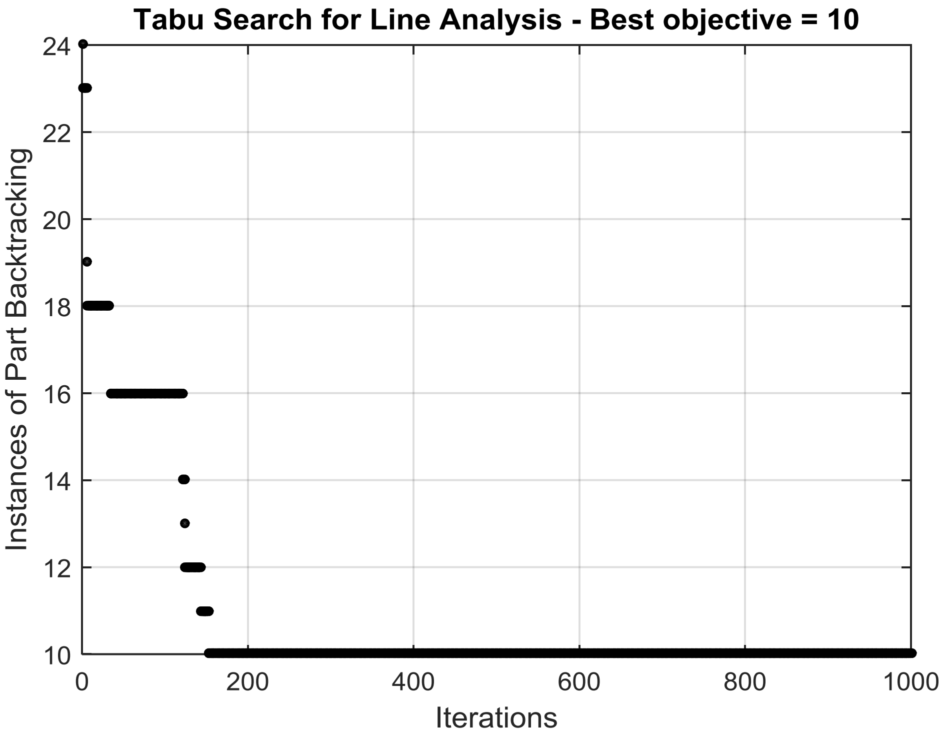

Figure 5.3 Tabu Search Algorithm – Best f^* per Iteration

Ten candidate solutions were generated per iteration and the length of the Tabu List was forty. The two operations for creating candidate solutions were performed with parameter $p = 0.5$. One additional copy of machine 9 placed in series reduces the number of part backtracking movements from 12 down to 10. No further improvement was found after approximately 180 iterations although the search was allowed to continue for 1000 iterations as shown in Figure 5.3. The best arrangement for machines/sites is given in Table 5.9.

Table 5.9: TS Results – Best Arrangement of Machine Sites

Location	1	2	3	4	5	6	7	8	9	10	11
Cell *x*	S2	S3	S1	S4	S5	S7	S6	S8	S9	S10	S9

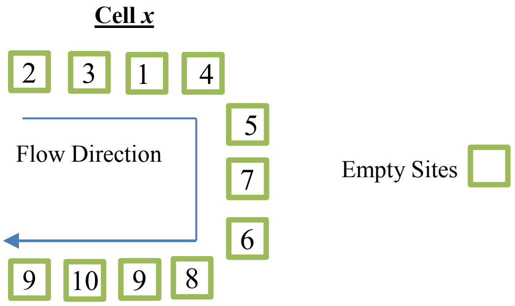

Figure 5.4 Arrangement of Machine Sits on FCT According to TC Result

Figure 5.4 shows how the sites may be arranged of a Factory Configuration Template based on the results of the Tabu Search. Note that for the given FCT, the instances of part backtracking may still be as high as 12 when no duplicate copy of machine 9 is available. Nonetheless, the additional site for machine 9 in series creates the opportunity to reduce this number although one copy of machine 9 may be available. A model for the intra-cell re-layout of machines will be presented in the next chapter; this model that enables additional sites to be exploited by identifying opportunities to eliminate part backtracking moves.

5.5 Method for Factory Configuration Template Creation

In Chapter 4-5 and subsections that have been presented thus far, methods have been proposed that address individual elements of Factory Configuration Template design. These individual methods must be combined to create the FCT in the following sequence:

1. Create a machine-component incidence matrix for all parts and machines for which the RCMS must be designed, including parts that are active in different periods.
2. Perform a silhouette analysis on part clusters generated by the FCMdd algorithm for various values of C in order to determine a suitable number of part families and cells.
3. Apply the FCMdd algorithm to generate a set of medoids and a membership matrix for parts for the chosen value of C. Use the membership matrix and medoids as a seed for the SFC heuristic. Apply the SFC heuristic to the machine-component incidence matrix in order to identify part families and sites for machines in cells. Generate multiple solutions for part and machine site clusters for various values of parameter L. It is recommended that experimentation begin from $L = 0.5$ down to $L = 0.1$ in steps of 0.1 or 0.05.
4. Analyse machine and site clusters for various values of L in terms of the number of sites required or the number of exceptional elements and voids in the solution. A solution may be chosen based on the number of sites that a given factory floor space can accommodate.

A solution may also be chosen based on the combination of risk and opportunity. Recall that a lower number of exceptional elements indicates increased opportunity to eliminate inter-cell part flow by reconfiguration. The number of voids indicates an increasing risk that some sites may have low utilization.

5. Determine the number of machines of each type that are required by the system in order to meet production capacity requirements. Compare the number of machines required at the system level to the number of sites proposed across all cells by the SFC. This comparison reveals if a sufficient number of sites have been made available. If additional sites are required, the cells in which the sites should be created are identified by an analysis of cell level site requirements.
6. Determine the number of each machine type that is required at the cell level. The number of machines required at the cell level is used to identify the cells in which sites in parallel should be created. Additional sites are only created in cells that already have one site as determined by the SFC heuristic. Note that the sites are added in parallel in order to enable a higher volume of flow through the cell.
7. Select an appropriate layout shape for each cell based on the need for additional machine sites in parallel.
8. Apply the TS algorithm to all part families and relevant machines in order to determine the optimal sequence in which machine sites should be arranged in cells. The optimal sequence is the one that supports unidirectional flow. Determine the mandatory instances of part backtracking that would be required by part families in order to complete all manufacturing processes in the prescribed order.
9. Apply the TS algorithm with X additional machines in order to determine if the mandatory instances of part backtracking can be reduced further by the possibility of additional machines and sites in series. If an advantage can be gained that offsets the investment in additional sites, accept this solution.

After steps 1-9 are complete, sufficient information is available that determines the number of sites that should be created for each machine in each cell. Information is also available on the shapes of cells and the order in which sites should be arranged within cells. This information is sufficient for a Factory Configuration Template to be created with the essential information for reconfiguration planning models to be applied. This nine step method is applied in the RCMS study of Chapter 7.

In an actual RCMS implementation, the FCT requires the following information to be complete:

1. Specification of machine sizes and footprint in order to determine the floor area dedicated to each site.

2. Specification of machine sizes so that lanes for machine movement can be created to support hassle free reconfigurations.
3. Specification of mandatory minimum safe distances between machines and specification of space required by machine operators in order for sites to be arranged in a manner that creates a safe working environment.
4. Specification of the relative orientation of cells with respect to each other and physical distances between cells.

This list is non-exhaustive and other types of information may be added to the FCT according to the requirements of individual manufacturers. Once the FCT is complete, the floor space design must be implemented for the arrangement of machines and sites on the factory floor to support future re-layout of the system.

5.6 Chapter Summary

In Chapter 4 the SFC heuristic was presented that generated sites for machines in multiple cells. The SFC heuristic had the ability to create a maximum of one site per machine type per cell. In this chapter methods were presented for the duplication of machine sites in parallel in order to facilitate future enhancements to cell capacity in order to meet the required the production volumes. A Tabu Search algorithm was also developed that optimises the arrangement of sites in cells in order to support unidirectional flow. The Tabu Search algorithm was then extended to propose the creation of additional machine sites in series, in order to further eliminate the backward flow of parts.

The chapter concluded with a description of how individual methods and algorithms should be integrated in order to complete the design of a Factory Configuration Template. The FCT can include various levels of information; the methods proposed thus far identify sites for machines in cells, the shapes of cells and the arrangement of sites within the cell. This information is sufficient for reconfiguration planning models to be developed that propose periodic changes to the layout of the RCMS in response to changing production requirements. The reconfiguration planning models are presented in the next chapter.

6 Reconfiguration Planning Models

6.1 Chapter Introduction

The concept of a Factory Configuration Template was proposed in Chapter 3 in order make the re-layout of manufacturing cells practically feasible. The problems associated with performing reconfigurations without an initial system design for re-layout were presented in section 2.8. The FCT is a design for the factory floor space that defines sites for machines; machines can be moved in and out of these sites in order to gain the advantages of increased machine utilization and less complex and lower cost part flows.

Chapter 4 and Chapter 5 presented the techniques used for creating the FCT. Recall that the FCT indicates the reservation of space within a cell for a particular machine type, it does not indicate the presence of an actual machine. The question then arises: “how are available machines to be positioned in the sites available across multiple periods?” This question is answered by developing and solving appropriate assignment models. This chapter presents models that are used to assign machines to the sites that have been reserved for them in a manner that considers both current and future production requirements of the system

6.2 The Reconfiguration Planning Task

The process of determining how the layout of a RCMS should evolve over multiple periods is referred to as reconfiguration planning. The series of changes that are to be implemented should be planned in a way that considers both the current and future requirements of the system in order to propose configurations that are optimal over a longer term.

In a RCMS there are machine that are classified as fixed or mobile. In Chapter 3, fixed machines were defined as machines that did not have multiple sites available and could be assigned to the default sites on the FCT without any further analysis. The process of reconfiguration planning therefore deals with the assignment of mobile machines to the sites available on the FCT over multiple periods. The multi-period assignments will determine how the configuration of the RCMS evolves in response to change.

Reconfiguration planning deals with the allocation of machines to sites and the development of assignment models are necessary to generate optimised plans for the re-layout of the system. Both inter-cell and intra-cell machine assignment models are developed in this chapter. Inter-cell assignment models deal with the allocation of machines among cells. In the case where more than one site is available for a machine in a cell, an intra-cell assignment model is developed that determines how a machine should be allocated between multiple sites in the same cell. The inter-cell and intra-cell multi-period assignment models are designed to achieve the following objectives:

- minimize the capacity surplus/deficit in cells;
- maximize machine utilization;
- minimize inter-cell material handling movements;
- minimize the cost of system reconfiguration;
- minimize the complexity of part flow in cells
- minimize the cost of operating a RCMS.

Many of these objectives are interrelated and not explicitly represented in a mathematical model. A RCMS only becomes economically viable when the cost of reconfiguration is offset by other types of costs such as inter- and intra-cell material handling costs. The economic trade-off will be investigated in the models that are developed.

The reconfiguration planning task requires a significant amount of information to be available at the outset in order to set up and solve inter- and intra-cell machine assignment models. The following information is required for each period to be considered in the reconfiguration plan:

- a relevant Factory Configuration Template;
- the types of parts and demand volumes for each period;
- the types of machines and machine numbers that will be made available in each period;
- the process cycle times for each part on each machine;
- the sequences in which each process will be performed for each part;
- the cost per part for an inter-cell material handling movement ;
- the cost per part for an intra-cell backtracking movement ;
- the cost per inter-cell machine relocation;
- the cost per intra-cell machine relocation;
- the budget available for reconfiguration per period;
- the total working hours per period.

This chapter proceeds with the development of two models for inter-cell machine assignment, presented in sections 6.3 and 6.4 respectively. A model for intra-cell machine assignment is presented in section 6.5. Examples of reconfiguration plans are presented to demonstrate the ability and output of each model.

6.3 Capacity Based Machine Assignment Model

6.3.1 Model Formulation

The model presented in this chapter was formulated under the following assumptions:

1. The costs of inter-cell and intra-cell material handling movements per part are parameterised as a fixed rate per time period and are not directly correlated to distance travelled within the factory.
2. The cost of reconfiguration (machine relocation) has been parameterised as a fixed rate per machine relocation and is not directly correlated to distance of relocation within the factory.
3. A reconfiguration cost is brought into effect for the gain and removal of a machine from a cell. The cell from which the machine is removed incurs a cost, which is the cost of uninstalling the machine and restarting the cell under a new configuration. The cell gaining a new machine incurs a cost. This is the cost of installing the machine and restarting the cell under a new configuration.
4. The cost of gaining/losing a machine is averaged to be equal for the cell that gains a machine and the cell that loses a machine.
5. The implementation of a FCT allows machines to be rearranged without introducing an additional cost for rearranging material handling equipment such as conveyors, robots and Automated Guided Vehicles.

The following notation is applicable to the model:

Sets:

$I = \{1, 2, \ldots, M\}$ is the set of machine types; i denotes a machine type and $i \in I$.

$J = \{1,2,\ldots,N\}$ is the set of cells; j denotes a single cell and $j \in J$.

$K= \{1,2, \ldots,T\}$ is the set of periods; k denotes a single period of time and $k \in \mathrm{K}$.

Decision variables:

x_{ijk} is a integer variable that signifies the number of machines of type i allocated to cell j in time period k. Note that all $x_{ijk} = 0$ for k=0.

Parameters:

H_{ijk} is the operating hours on machine type i required by parts belonging to cell j in period k.

W_k is the number of working hours prescribed for the factory in period k.

r_k is the cost of a cell gaining/losing a machine in period k.

R_k is the budget available for reconfiguration (machine re-layout) in time period k.

N_{ik} is the maximum number of machines of type i available for assignment in period k.

X_{ij} is the number of sites available for machine i in cell j according to the FCT.

Mobile machines may be allocated to cells according to various objectives, the model presented here allocates machines to cells in each period, based on the machine capacity required by a cell. The capacity, represented as hours of machine time required by a cell, is determined by equation 40.

The objective function, shown in equation 57, is designed is to minimize the sum of surplus and deficit hours of capacity for all cells, across multiple periods, through the appropriate assignment of machines.

$$min \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \left| W_k x_{ijk} - H_{ijk} \right| \qquad (57)$$

Subject to:

$$\sum_{i \in I} \sum_{j \in J} r_k \left| x_{ijk} - x_{ij(k-1)} \right| \leq R_k, \qquad \forall k \in K \qquad (58)$$

$$\sum_{j \in J} x_{ijk} = N_{ik}, \qquad \forall i \in I, k \in K \qquad (59)$$

$$x_{ijk} \leq X_{ij}, \qquad \forall i \in I, j \in J, k \in K \qquad (60)$$

$$x_{ijk} \in Z^+ \qquad (61)$$

The first constraint restricts machine re-layout by a reconfiguration budget R_k. A cost r_k is incurred for an increase or decrease in machine count, the reconfiguration cost is therefore borne by the cell that loses a machine and the cell that gains a machine. Reconfiguration cost and budget change in each period; this requires that a constraint of this type to be implemented for every $k \epsilon K$. The second constraint allows a maximum of N_{ik} machines of type i to be assigned to cells in period k; a constraint of this type is necessary for each machine type i in every period k. This constraint ensures that no additional machines are allocated than what is available. The third constraint restricts the number of machines of type i assigned to cell j to a maximum of X_{ij} in all periods. The maximum number of machines that a cell can accept is determined by the number of sites that have been made available on the FCT. Note that X_{ij} effectively places an upper bound on each variable x_{ijk}. The final constraint restricts the decision variables to positive integer values.

6.3.2 Conversion of Capacity Based Model to Linear Integer Form

The model presented in equations 57-61 is not in a form that can be solved by a software program. The absolute value terms in the objective function and constraints may be eliminated by variable substitution in order to convert the model to a linear integer program. The following variable substitutions are necessary:

$$u_{ijk} - v_{ijk} = W_k x_{ijk} - H_{ijk}, \qquad \forall i \in I, j \in J, k \in K \qquad (62)$$

$$p_{ijk} - q_{ijk} = x_{ijk} - x_{ij(k-1)}, \qquad \forall i \in I, j \in J, k \in K \qquad (63)$$

The variable substitution introduces four additional variables to the model as follows:

- u_{ijk} is an integer variable indicating the hours of excess capacity with respect to machine type i, in cell j in period k.

- v_{ijk} is an integer variable indicating the hours of deficit capacity with respect to machine type i, in cell j in period k.
- p_{ijk} is an integer variable indicating the number of machines of type i added to cell j in period k.
- q_{ijk} is an integer variable indicating the number of machines of type i removed from cell j in period k.
- y_{ijk} is a zero-one variable that ensures that either p_{ijk} or q_{ijk} will assume a positive value while the while the other remains at zero.

The capacity based assignment model in linear integer form after variable substitution is as follows:

$$min\, F = \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} u_{ijk} + v_{ijk} \tag{64}$$

Subject to:

$$-u_{ijk} + v_{ijk} + W_k x_{ijk} = H_{ijk}, \qquad \forall i \in I, j \in J, k \in K \tag{65}$$

$$\sum_{i\in I}\sum_{j\in J} r_k \left(p_{ijk} + q_{ijk}\right) \leq R_k, \qquad \forall k \in K \tag{66}$$

$$p_{ijk} - q_{ijk} - x_{ijk} + x_{ij(k-1)} = 0, \qquad \forall i \in I, j \in J, k \in K \tag{67}$$

$$p_{ijk} \leq M y_{ijk}, \qquad \forall i \in I, j \in J, k \in K \tag{68}$$

$$q_{ijk} \leq M\left(1 - y_{ijk}\right), \qquad \forall i \in I, j \in J, k \in K \tag{69}$$

$$\sum_{j\in J} x_{ijk} = N_{ik}, \qquad \forall i \in I, k \in K \tag{70}$$

$$x_{ijk} \leq X_{ij}, \qquad \forall i \in I, j \in J, k \in K \tag{71}$$

$$y_{ijk} \in \{0,1\};\ x_{ijk}, p_{ijk}, q_{ijk}, u_{ijk}, v_{ijk} \in Z^+ \tag{72}$$

The objective function minimizes the sum of surplus and deficit hours of machine capacity in manufacturing cells. Note that in the optimal solution either u_{ijk} or v_{ijk} will assume a value of zero while the other will assume a positive integer value. The first constraint enforces the variable substitution that was performed in order to eliminate the absolute value from the objective function. The second constraint enforces the reconfiguration budget. Variables p_{ijk} and q_{ijk} were introduced into the model in order to eliminate the absolute value of equation 58. The third, fourth and fifth constraints enforce the variable substitution. Note that variable y_{ijk}, which features in the fourth and fifth constraints, is a zero-one variable that ensures the condition that either p_{ijk} or q_{ijk} will assume a positive value while the other remains at zero. This was necessary since p_{ijk} and q_{ijk} have objective function coefficients of zero. The constant M, which ensures the correct functioning of conditional constraints four and five, is an arbitrarily large integer. The only requirement on the constant M, is that it be larger than the largest possible number of machines entering or leaving a cell. The sixth constraint allows a maximum of N_{ik} machines of type i to be assigned to cells in time period k. The seventh constraint allows a maximum of X_{ij} machines of type i in cell j in all time periods, this ensures that the reconfiguration occurs within the specification of the FCT. The final constraint restricts all variables to positive integer values, with the exception of y_{ijk} which is restricted to zero or one.

The variable substitution, that was performed in order to convert an integer programming model with absolute values to a linear integer program, increases the number of decision variables by a factor of six.

The model takes into consideration the initial placement/installation cost of machines when first purchased. The reconfiguration budget R_k must therefore include initial placement cost in addition to the budget made available for rearranging the positions of existing machines on a factory floor.

The model treats the cost of adding/removing machines to/from cells in a uniform manner, i.e. the cost of adding/removing a machine is fixed at r_k for a time period k, irrespective of machine type or distance moved. This is suitable for a factory where the distance between manufacturing cells is similar for all pairs of cells and the cost of relocating each machine type is approximately the same. If a machine has a higher reconfiguration cost due to its size, weight, or other technical factor, this is dealt with by the parameterization of reconfiguration cost as r_{ik} as opposed to r_k.

6.3.3 Solution by Branch and Bound with Cutting Planes

The capacity based assignment model is a model of the form:

$$min\, f^T X \tag{73}$$

$$A.X \leq b \tag{74}$$

$$Aeq.X = beq \tag{75}$$

$$X \in \mathbb{Z}^+ \tag{76}$$

Where f is a vector of objective function coefficients, A and Aeq are arrays of coefficients corresponding to the left hand side of constraints, b and beq are the vectors corresponding to the right hand side values of constraints and X represents a vector of n decision variables.

Since the model was possible to represent in a standard form, the Matlab® *intlinprog* solver was used to solve it. This is the only commercial solver that was used in this research. All other algorithms were developed specifically for this research in the Matlab® program development environment.

The *intlinprog* solver implements a Branch and Bound or Branch and Cut algorithm to solve integer and mixed integer programs. The Branch and Bound algorithm was originally developed by Land and Doig [86] and later extended by researchers such as Balas et al [87] to include cutting planes. The Branch and Bound algorithm with cutting planes (Branch and Cut) for minimization problems is described here briefly.

Branch and Bound With Cutting Planes

```
//Initialisation
P=Ø; // Define an empty set of problems
$p_o$ = InitialProblem; //Define the initial problem as the original ILP
$Z_{best}$ = ∞; //Define best objective function value
UpperBound=$Z_{best}$; // Define the upper bound on the problem
BestSolution = null; // Define the best solution as null
P=P+$p_o$; // Add the initial problem to the set of problems
Repeat
    p* = SelectProblem(P); // Select a problem from among active problems
    [solution, feasible ] = SolveRelaxation(p*); //Solve the LP relaxation of the problem

    if feasible = True // If the solution is feasible

        [F*] = ObjectiveValue(solution); // Determine the objective value of the solution

        if f<ZBest // If the objective value is lower than the best known solution

            Integer= isinteger(solution); // Determine if the solution is integer

            if Integer =  True // If the solution is integer

                BestSolution=solution; // Set the solution as the best solution
                ZBest=F*; // Set the objective value as the best value found

                // Update the upper bound on the problem
                UpperBound=ZBest;
            else
                //Add cutting planes to the LP relaxation
                p*= AddCuttingPlanes(p*,solution);
                if CuttingPlanesFound(p*)
                    // Solve the LP relaxation again
                    goto SolveRelaxation(p*);
                end
```

```
                end
            end
        end

        [P] = Branch(P)// Branch on the existing problems to create new sub-problems
        // Identify active problems and establish a lower bound
        [P, LowerBound]= ActiveProblems(P);
Until (P=Ø or UpperBound-LowerBound<ε) //Repeat until there are no active problems or the difference between upper and lower bounds is less than epsilon
```

At initialization a data structure *P* is defined to hold the set of problems. Traditionally the Branch and Bound algorithm is conducted with a tree type data structure where each node holds a candidate problem. A candidate problem is a problem with a feasible region that is more restricted relative to the original problem. Initially the best objective value and upper bound are set to infinity and the best solution is set to null. The *intlinprog* solver has the ability to apply heuristics to obtain an initial integer feasible solution and an upper bound, however, this option was not used. The upper bound on future solutions is the objective function value of the best integer feasible solution that has been found. The original problem p_o is added to the data structure *P* that contains all problems.

In each iteration of the algorithm a problem is selected from the list of active problems. The problem is then solved as a linear programming problem by relaxing the integer restriction on variables. The primal-simplex algorithm was the option used with the *intlinprog* solver for all linear programming relaxations. The explanation of the primal-simplex algorithm does not add any value to the objectives of this research; for a description of the primal-simplex algorithm see Vanderbei [88]. If the solution to the relaxed problem is feasible the objective function value is computed. If the variables of the feasible solution are all integer the solution is compared to the best known solution. If the objective value is lower than that of the best known solution it replaces the best solution and the upper bound is updated. If the solution is feasible but with variables that are not integer, cutting planes may be added to problem p^* to create a new problem with a tighter lower bound. Cutting planes are added by introducing additional linear constraints to the problem. If cutting planes have been found and added to p^* then the problem is solved again as a linear programming relaxation. If no cutting planes are found the algorithm proceeds to the branch function.

The branch function creates two candidate problems from a selected parent problem by partitioning the feasible region. The list of active problems is then updated; problems that are fathomed are 'pruned' from the set of active problems *P*. A problem is considered to be fathomed if the outcome has been determined without solving it. If a problem has a lower bound that is higher than the established upper bound, this problem is considered to be fathomed and is pruned. The lower bound on a problem is the objective value that was obtained by solving the linear programing relaxation of the parent problem from which the new problem was derived.

Note that the *intlinprog* solver is able to operate with or without the use of cutting planes. Cutting planes were only used in situations where solutions were not found in reasonable time. The *intlinprog* solver has multiple types of cutting planes that may be implemented. Types of cuts that were used to solve integer programming models in this research included Gomory cuts, developed by Gomory [89]; flow cover cuts, by Gu et al [90]; strong CG cuts, by Letchford and Lodi [91]; zero-half cuts, by Caprara and Fischetti [92] and clique cuts, by Atamtürk et al [93].

6.3.4 Example - Application of Capacity Based Assignment Model

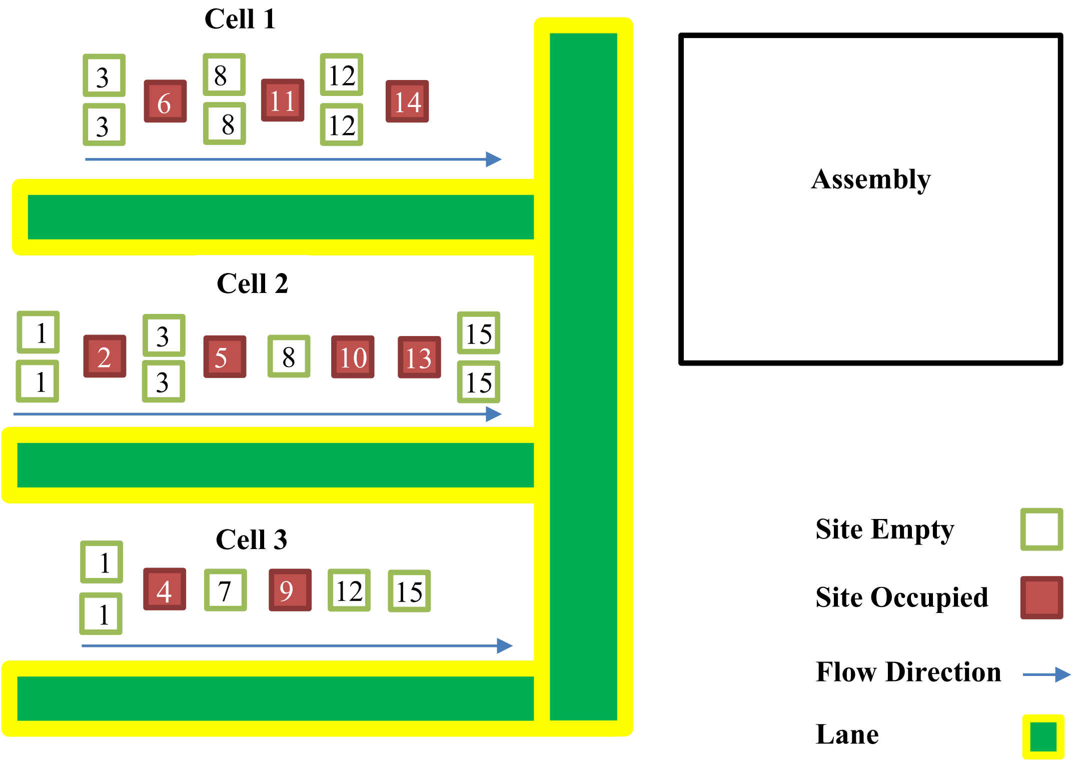

Figure 6.1: Example Factory Configuration Template

Consider the Factory Configuration Template shown in Figure 6.1. A reconfiguration plan must be developed for this RCMS, which will determine how the layout and clusters of cells will evolve in an optimal manner over three periods. Machines 1, 3, 8, 12 and 15 are mobile machines and the assignment of these machines to sites will be determined by the capacity based assignment model. Machines 2, 4, 5, 9, 10, 11, 13 and 14 have one site available on the FCT and have already been assigned to the respective sites as shown in Figure 6.1. These machines may be excluded from the assignment model.

Table 6.1: Operating Hours Required per Machine per Cell per Period

	Period 1 – Operating Hours			Period 2 – Operating Hours			Period 3 – Operating Hours		
Machine	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**
1	0	6735	5560	0	3982	3768	0	0	6375
2	0	3504	0	0	3231	0	0	3036	0
3	5454	3700	0	4825	6394	0	5425	3700	0
4		0	3895	0	0	3900	0	0	4137
5	0	3938	0	0	3938	0	0	3938	0
6	3928	0	0	3050	0	0	3351	0	0
7	0	0	4126	0	0	3831	0	0	3217
8	3716	3286	0	3716	4579	0	6973	2798	0
9	0	0	3541	0	0	4125	0	0	3736
10	0	3573	0	0	3573	0	0	3573	0
11	3442	0	0	3996	0	0	3256	0	0
12	6539	0	2360	5237	0	3099	5721	0	5601
13	0	3259	0	0	3615	0	0	3551	0
14	3292		0	3502		0	4284	0	0
15	0	3154	0	0	4214	3020	0	6081	5781

Table 6.2: Reconfiguration Expense and Budget

	Period 1	Period 2	Period 3
r_k	R 5000.00	R 5500.00	R 6000.00
R_k	R 110000.00	R 30000.00	R 30000.00

Table 6.3: Machines Available per Period

	Machines Available		
Machine	**Period 1**	**Period 2**	**Period 3**
1	3	3	3
2	1	1	1
3	3	3	3
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1
8	2	2	3
9	1	1	1
10	1	1	1
11	1	1	1
12	3	3	3
13	1	1	1
14	1	1	1
15	1	2	3

The hours of machine capacity, H_{ijk}, required by each cell in each period is presented in Table 6.1. In this abbreviated example, individual part demands are not shown, however, the hours may be calculated from part cycle times and volumes using the procedure presented in section 5.2.

Each reconfiguration period is six months long (180 days) and each working day is 24 hours, the total working hours per period is therefore $W_k = 4320\ hours$. The reconfiguration of the RCMS is restricted by a budget, the cost per gain/loss of a machine and the total budget per period is listed in Table 6.2. The machines that are to be made available to the system per period are listen in Table 6.3.

Solution

The capacity based assignment model minimises the sum of excess and deficit hours of capacity per machine per cell per period. Some excess capacity will always exist if sufficient machines are made available; consider that 2.5 machines cannot be made available, if 2.5 machines are required then 3 machines must be made available. The excess machine capacity E_{ik}, represented as excess hours, is calculated by equation 77:

$$E_{ik} = W_k N_{ik} - \sum_{j \in J} H_{ijk}, \qquad \forall i \in I, k \in K \tag{77}$$

The calculation for this problem is presented in Table E - 1, Appendix E.1. This calculation is significant because it is an indicator that the capacity based assignment model can never achieve an objective value below the figure of 41007 hours. This can be viewed as a type of basic lower bound on the model. A portion of the reconfiguration budget (R 50000.00) is used for the layout of the ten fixed machines in the first period. This leaves a balance of R 60000.00 for the layout of mobile machines in the first period.

The capacity based assignment model for this problem is not shown here due to its size; for five transient machines allocated to three sells in three periods, the model consists of 270 variables and 243 constraint equations. A program was written in the Matlab® program development environment to automatically populate the *A* and *Aeq* arrays and *f*, *b* and *beq* vectors of equations 73-75, using the essential information provided in Table 6.1 - Table 6.3. The program code is available on the accompanying CD. The model and solution are also available in a Microsoft® Excel spreadsheet on the accompanying CD.

Figure 6.2 shows the convergence of the Matlab® *intlinprog* solver when applied to the capacity based assignment model for this problem. The figure shows that the Branch and Bound method was able to determine the best solutions without using cut generation.

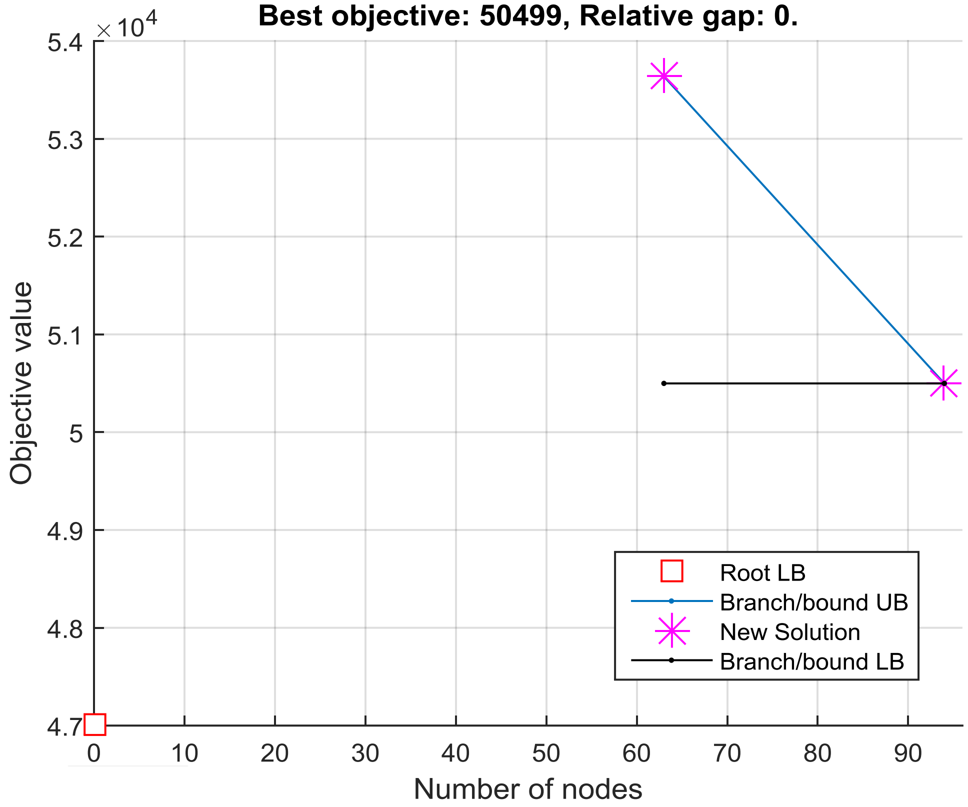

Figure 6.2: Convergence of *intlinprog* solver for the Capacity Based Assignment Model

The best solution of 50499 hours was found after the algorithm explored +90 nodes (candidate problems). The assignment of mobile machines to various cells as determined by the model is shown in Table 6.4. The table is effectively a reconfiguration plan for three periods; the placement of a machine is indicated in green and the removal of a machine is indicated in red.

Table 6.4: Three Period Reconfiguration Plan

	Period 1			Period 2			Period 3		
Machine	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**
1	0	2	1	0	1	2	0	1	2
3	2	1	0	1	2	0	2	1	0
8	1	1	0	1	1	0	2	1	0
12	2	0	1	2	0	1	2	0	1
15	0	1	0	0	1	1	0	2	1

In the first period the machines are to be installed in their respective cells to form the initial manufacturing system configuration. R 60000.00 will be spent on the initial machine placement/installation of mobile machines. In the second period (6-12 months), the following configuration changes are proposed:

- one copy of machine type 1 to be relocated from cell 2 to cell 3;
- one copy of machine type 3 to be relocated from cell 1 to cell 2;
- one new copy of machine type 15 to be added to the system and placed in cell 3;
- a total of R 27500.00 will be spent on reconfiguration.

In the third period (12-18 months), the following configuration changes are proposed:

- one copy of machine type 3 to be relocated from cell 2 to cell 1;
- one new copy of machine 8 to be added to the system and placed in cell 1;
- one new copy of machine 15 to be added to the system and placed in cell 2;
- a total of R 24000.00 will be spent on reconfiguration.

Table 6.5 presents results on the assignment of machine capacity to cells. The objective function value for the problem with mobile machines only, yielded a result of $F = 50499$ hours of surplus and deficit capacity. The total surplus and deficit capacity in each period is calculated by applying equations 78 and 79 to the values of variables u_{ijk} and v_{ijk} in the optimal solution.

$$U_k = \sum_i \sum_j u_{ijk} \tag{78}$$

$$V_k = \sum_i \sum_j v_{ijk} \tag{79}$$

The table shows the total surplus and deficit capacity per period F_k.

Table 6.5: Capacity Allocation Results

	Period 1	Period 2	Period 3	Total
U_k (hours)	12576	14090	19087	45753
V_k (hours)	1240	764	2742	4746
$F_k = U_k + V_k$	13816	14854	21829	50499
$\sum E_{ik}$ (hours)	11336	13326	16345	41007

Recall that excess capacity is introduced as the number of machines to be purchased must be rounded up to an integer value during calculations. Across the three periods 41007 hours of excess capacity was introduced by purchasing excess machine capacity. A total of 45753 hours of excess capacity existed in cells after the assignment, this is due to excess capacity existing in some cells while other cells experienced a capacity deficit. This occurs because machines cannot be split between cells and are assigned using integer values. The total capacity deficit existing in cells is 4746 hours; the parts that require the missing capacity will perform inter-cell material handing movements in order to access the required machines. Note that the following relationship exists between surplus and deficit capacity:

$$\sum_{i\in I}\sum_{j\in J} v_{ijk} = \sum_{i\in I}\sum_{j\in J} u_{ijk} - \sum_{i\in I} E_{ik}, \qquad \forall\, k \in K \tag{80}$$

That is, the surplus capacity purchased subtracted from the surplus capacity via machine assignment equals the surplus deficit. It is therefore possible to minimize the capacity deficit only and obtain the same result.

6.4 Cost Minimization Assignment Model

6.4.1 Model Formulation

The model presented in section 6.3 may be modified to demonstrate an economically advantageous trade-off between reconfiguration cost and the cost of inter-cell part movements. The capacity based assignment model is converted into a cost minimization model by removing the restriction on reconfiguration budget and adding the cost of reconfiguration to the objective function. A cost C_{ijk} is incurred in the objective function for each hour of deficit capacity experienced in cell j with respect to machine i in period k. The assumption is that when a capacity deficit exists parts will make an inter-cell movement to another cell in order to access the required capacity. The model is as follows:

$$min\ \text{Cost} = \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} r_k\left(p_{ijk} + q_{ijk}\right) + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} C_{ijk} v_{ijk} \tag{81}$$

Subject to:

$$-u_{ijk} + v_{ijk} + W_k x_{ijk} = H_{ijk}, \qquad \forall i \in I, j \in J, k \in K \tag{82}$$

$$p_{ijk} - q_{ijk} - x_{ijk} + x_{ij(k-1)} = 0, \qquad \forall i \in I, j \in J, k \in K \tag{83}$$

$$u_{ijk} \le M y_{ijk}, \qquad \forall i \in I, j \in J, k \in K \tag{84}$$

$$v_{ijk} \le M\left(1 - y_{ijk}\right), \qquad \forall i \in I, j \in J, k \in K \tag{85}$$

$$\sum_{j\in J} x_{ijk} = N_{ik}, \qquad \forall i \in I, k \in K \tag{86}$$

$$x_{ijk} \le X_{ij}, \qquad \forall i \in I, j \in J, k \in K \tag{87}$$

$$y_{ijk} \in \{0,1\};\ x_{ijk}, p_{ijk}, q_{ijk}, u_{ijk}, v_{ijk} \in Z^{+} \tag{88}$$

The objective function minimizes the sum of reconfiguration cost and the sum of the cost of deficit hours of capacity in cells. The first constraint ensures that excess and deficit hours of capacity are represented as positive values. The second constraint ensures that the loss or gain of a machine is represented as a positive value. The excess capacity u_{ijk} has a zero coefficient in the objective function in this model since no cost is incurred for cells that possess more capacity than required. Constraints three and four ensure that u_{ijk} or v_{ijk} assumes a positive value but not both variables at the same time. The zero-one variable y_{ijk} is used in conjunction with a constant M to implement the conditional constraints. M is an arbitrarily large constant; the only requirement on the constant M is that it be larger than the largest possible capacity surplus or deficit. The fifth constraint allows a maximum of N_{ik} machines of type i to be assigned to cells in time period k. The sixth constraint allows a maximum of X_{ij} machines of type i in cell j in all time periods, this ensures that the reconfiguration occurs within the allowance of the FCT. The final constraint restricts all variables to positive integer values, with the exception of y_{ijk} which is restricted to zero or one.

6.4.2 Estimation of Cost Coefficients

The cost C_{ijk} of an hour of deficit capacity in a cell is difficult to determine precisely, however an acceptable approximation can be calculated as follows:

$$C_{ijk} = \left\lceil \frac{60}{\bar{t}_{ij}} \right\rceil c_k \tag{89}$$

The cost is based on the number of parts that will perform an inter-cell material handling movement for each hour of deficit capacity experienced in a cell. The number of parts affected by the deficit is determined by dividing 60 minutes by the average process cycle time $\bar{t}_{ij}$ of a parts on machine i in cell j. The product of the number of parts by the cost per inter-cell par movement c_k yields the estimated cost per hour capacity deficit.

6.4.3 Example Application of Cost Minimization Assignment Model

The cost minimization assignment model can also be represented in the standard form shown in equations 73-76. The model was also solved using the Matlab® *intlinprog* solver. The problem used in this example is the same as the problem of section 6.3.4 with the additional assumption that the cost of an hour deficit capacity, C_{ijk}, is R 10.00, R 12.00 and R 14.00 in periods 1, 2 and 3 respectively.

The cost minimization assignment model for this problem is not shown here due to it's size; for five transient machines allocated to three sells in three periods, the model consists of 270 variables and 240 constraints. A program was written in the Matlab® program development environment to automatically populate the A and Aeq arrays and f, b and beq vectors of equations 73-76, using the essential information. The program code is available on the accompanying CD. The model and solution are available in a Microsoft® Excel spreadsheet on the accompanying CD.

Figure 6.3 shows the convergence of the Matlab® *intlinprog* solver when applied to the cost minimization assignment model for this problem. The best cost solution of R 167284.00 was found after the algorithm explored +350 nodes (candidate problems) without the need for cutting planes. The cost consists of machine relocation between cells and the cost of inter-cell part movements due to a deficit in the required number of operating hours on the various types of mobile machines.

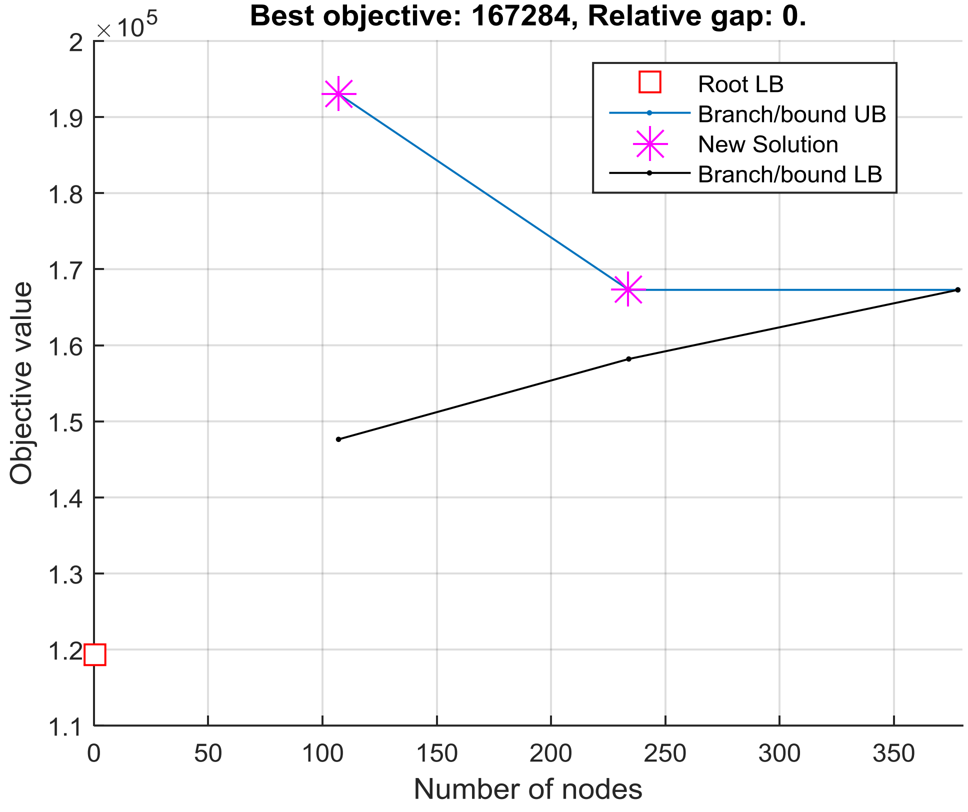

Figure 6.3: Convergence of *intlinprog* solver for the Cost Minimization Assignment Model

The assignment of mobile machines to various cells as determined by the model is shown in Table 6.6. A comparison between the reconfiguration plan of Table 6.6 and that of Table 6.4 reveals that the cost minimization assignment model has proposed two less machine relocations as compared to the capacity based assignment model.

Table 6.6: Alternative Three Period Reconfiguration Plan

	Period 1			Period 2			Period 3		
Machine	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**	**Cell 1**	**Cell 2**	**Cell 3**
1	0	2	1	0	1	2	0	1	2
3	2	1	0	2	1	0	2	1	0
8	1	1	0	1	1	0	2	1	0
12	2	0	1	2	0	1	2	0	1
15	0	1	0	0	1	1	0	2	1

The capacity results pertaining to the cost minimization assignment model were determined by applying equations 78 and 79 to the values of variables u_{ijk} and v_{ijk} in the optimal solution. The surplus and deficit capacity results are shown in Table 6.7. A comparison between Table 6.5 and Table 6.7 reveals that the cost minimization model has a larger deficit of 6315 hours as compared to 4746 hours in the previous solution. The choice between the reconfiguration plans depends on what a manufacturer might deem as less inconvenient; two additional machine relocations or an additional 1569 hours of deficit capacity in cells.

Table 6.7: Capacity Allocation Results

	Period 1	Period 2	Period 3	Total
U_k *(hours)*	12576	15659	19087	47322
V_k *(hours)*	1240	2333	2742	6315
$F_k = U_k + V_k$	13816	17992	21829	53637
$\sum E_{ik}$	11336	13326	16345	41007

6.5 Intra-Cell Machine Assignment Model

The capacity based assignment model and the cost minimization model are classified as inter-cell machine assignment models. This is due to the fact that these models are able to assign machines among cells. In the case where there are duplicate machine sites in parallel, the inter-cell reconfiguration models are sufficient since it usually does not matter which site in parallel is occupied.

In a cell where there are duplicate sites in series, an inter-cell assignment model does not specify exactly which sites should be occupied. When considering machines arranged in series, the site that is occupied by a mobile machine is significant since the choice of site can be used to eliminate some instances of part backtracking. In this section an intra-cell machine assignment model is developed that assists with this decision. Before proceeding with the model formulation, the terms *first site* and *second site* must be defined. The first site is the site that appears at an earlier location in a sequence of machines and the second site appears later in a sequence of machines. In Figure 6.4 the first site with respect to machine type 4 is at the second location and the second site is at the fifth location in the sequence.

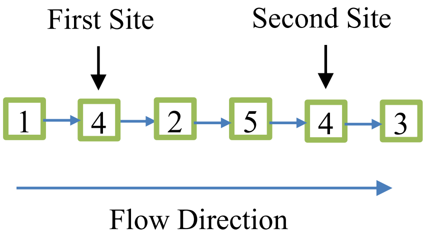

Figure 6.4: Illustration of First and Second Sites for Machine Type 4

6.5.1 Model Formulation

A manufacturing cell may have multiple sites in series reserved for a single machine type; the sites are created for a machine that is used to negate the intra-cell backtracking movements of parts, as demonstrated in section 5.4.5. The model presented in this section is used to assign a single machine to sites within the cell when two possible sites are available. The problem of assigning multiple mobile machines to multiple sites in parallel is a topic for future research.

The following notation is applicable to the intra-cell machine assignment model:

Sets:

$K = \{1, 2, \dots, T\}$ is the index set of time periods; k denotes a single time period and $k \in K$.

Decision variables:

x_k is a zero-one variable indicating the assignment of the machine to the first site.

y_k is a zero-one variable indicating the assignment of the machine to the second site.

a_k is a zero-one variable indicating that the machine in the first site has been gained by reconfiguration in period k.

b_k is a zero-one variable indicating that the machine in the first site has been removed by reconfiguration in period k.

p_k is a zero-one variable indicating that the machine in the second site has been gained by reconfiguration in period k.

q_k is a zero-one variable indicating that the machine in the second site has been removed by reconfiguration in period k.

Parameters:

α_k is the volume of parts that will perform an intra-cell backtracking movement if a machine is not available in the first site in period k.

β_k is the volume of parts that will perform an intra-cell backtracking movement if a machine is not available in the second site in period k.

c_k is the cost per part, per intra-cell backtracking movement in period k.

r_k is the cost of relocating a single machine within the cell in period k.

N_k is the maximum number of machines of the specified type that have been made available to the cell in period k; $N_k \in \{0, 1, 2\}$.

The objective of the model is to minimize the cost of backtracking flow and the cost of relocating a machine to negate the undesirable flow. The assignment models of sections 6.3 and 6.4 will propose the movements of machines in and out of cells. The number of machines of a given type that have been allocated to the cell in period k is indicated by the variable N_k. The challenge in representing the objective function is to impose a reconfiguration cost when a machine is relocated internally to the cell and not to impose a reconfiguration cost when sites x_k or y_k gain a machine from another cell. The cost of inter-cell machine relocation has already been taken into consideration by the previous inter-cell assignment models. The necessity to impose a reconfiguration cost only when a machine is relocated within the cell has resulted in the creation of additional variables a_k, b_k, p_k and q_k.

The first summation of the objective function (equation 90) represents the backtracking flow cost if the machine in question is not made available in the first site, i.e. if $x_k = 0$ the cost $c_k\alpha_k$ is imposed. The second summation of the objective function represents the backtracking flow cost if the machine in question is not made available in the second site. Note that if $x_k = 0$ and $y_k = 0$, no backtracking cost is incurred at all since the parts will be performing an inter-cell part movement. This cost is already accounted for by the inter-cell machine assignment models. The third summation represents a reconfiguration cost if a machine is removed from the second site and placed in the first site. The fourth summation represents a reconfiguration cost if a machine is removed from the first site and placed in the second site. A reconfiguration cost is therefore imposed if a machine is moved internally. If a machine is lost/gained in either site to/from another cell, no reconfiguration cost is incurred in this model since inter-cell machine relocation costs are already accounted for.

$$\begin{aligned} min \text{ Cost} = \sum_{k\in K} c_k\alpha_k y_k(1-x_k) + \sum_{k\in K} c_k\beta_k x_k(1-y_k) \\ + \sum_{k\in K} r_k(a_k q_k) + \sum_{k\in K} r_k(p_k b_k) \end{aligned} \tag{90}$$

Subject to:

$$x_k - x_{(k-1)} - a_k + b_k = 0, \qquad \forall\, k \in K \tag{91}$$

$$y_k - y_{(k-1)} - p_k + q_k = 0, \qquad \forall\, k \in K \tag{92}$$

$$a_k + b_k \leq 1, \qquad \forall\, k \in K \tag{93}$$

$$p_k + q_k \leq 1, \qquad \forall\, k \in K \tag{94}$$

$$x_k + y_k = N_k, \qquad \forall\, k \in K \tag{95}$$

$$x_k, y_k, a_k, b_k, p_k, q_k \in \{0,1\} \tag{96}$$

The objective function does not add a charge for both a machine loss and gain at sites. In the inter-cell machine assignment models the assumption is that the cost of restarting a cell after reconfiguration is included in the cost per machine loss/gain. When an inter-cell machine relocation occurs, two cells must start-up under a new configuration, therefore a cost is charged to the cell that gains a machine and the cell that loses a machine. With respect to an intra-cell machine swap between sites in series, a cost is charged to the objective function once since only one cell is affected.

The first constraint of the model is a balance equation which will enforce $a_k = 1$ if a machine is gained in the first site and $b_k = 1$ if a machine is removed from the first site. Likewise, the second constraint is also a balance equation that will cause p_k and q_k to indicate the gain or loss of a machine at the second site. The third and fourth constraints allow either a loss or gain to be reflected in each site in each period *k*. The third and fourth constraints ensure the correct functioning of the first and second constraints. The fifth constraint ensures that all machines that have been made available to the cell are allocated to the positions available. The final constraint restricts all variables to zero-one values.

6.5.2 Notes on the Intra-Cell Machine Assignment Model

The model presented here has two intuitive variables x_k and y_k; however four additional variables a_k, b_k, p_k and q_k have been added to the model in order to correctly account for the cost of reconfiguration. The additional variables increase the difficulty of finding a solution substantially, the number of possible solutions is determined by the following equation:

$$Solutions = 2^{6T} \tag{97}$$

Where *T* is the total number of periods for which the model will be attempted to be solved. As an example, for *T*=5 periods the total number of possible solutions is 1073741824.

In many instances the solution to the problem of intra-cell machine assignment model is intuitive since the process sequence of parts may unanimously favour a machine placement in one site over the other. Moreover, if two machines of the same type are available both sites will be filled and the choice does not require the application of the model. The intra-cell machine assignment model was presented here to provide a complete set of tools for planning the reconfiguration of a RCMS, however, reconfiguration planners should examine the problem for a solution by inspection before proceeding to solve the model with an algorithm.

6.5.3 Implicit Enumeration Algorithm for Quadratic Zero-One Models

The model presented in section 6.5.1 is a type of quadratic multi-period assignment model. The model can be represented in standard form by the grouping of linear and quadratic terms in the objective function:

$$min\, Z = C^T X + \frac{1}{2} X^T Q X + constant \tag{98}$$

$$A.X \leq b \tag{99}$$

$$Aeq.X = beq \tag{100}$$

$$X^n \in \{0,1\} \tag{101}$$

Here, *C* is a vector of coefficients to linear objective function terms, *Q* is an *n*×*n* symmetric matrix of coefficients to quadratic terms and *X* is a vector of decision variables; *A* and *Aeq* are arrays of coefficients corresponding to the left hand side of constraints, *b* and *beq* are the vectors corresponding to the right hand side values of constraints.

The method of Implicit Enumeration is suitable for solving zero-one quadratic models. It is a variation of the Branch and Bound algorithm that exploits the zero-one characteristic of decision variables. In a conventional Branch and Bound algorithm with integer variables, the integer restriction on variables is relaxed and candidate problems are solved as linear programs by the primal-simplex algorithm. In Implicit Enumeration the linear constraints are relaxed while the zero-one restriction is maintained. Variables are set using rules in order to solve the relaxed version of the problem.

An Implicit Enumeration solver for zero-one quadratic programs was developed specifically for this research in the Matlab® program development environment. The program code is available on the accompanying CD. The Implicit Enumeration algorithm for minimization is as follows:

Implicit Enumeration Algorithm

```
P=Ø; // Define an empty set of problems
p_o = InitialProblem; //Define the initial problem as the original problem
Z_best = ∞; //Define best objective function value
UpperBound=Z_best; // Define the upper bound on the problem
LowerBound=LowerBound(p_o); // Define the initial lower bound using equation 102
BestSolution = null; // Define the best solution as null
P=P+p_o; // Add the initial problem to the set of problems
Repeat
        p* = SelectProblem(P); // Select a problem from among active problems
        solution = SolveRelaxation(p*); //Solve the relaxation of the problem

        feasible= FeasibilityCheck(solution); //Determine if the solution is feasible

        if feasible = True // If the solution is feasible

                f = ObjectiveValue(solution); // Determine the objective value of the solution

                if f<ZBest // If the objective value is lower than the best known solution

                                BestSolution=solution; // Set the solution as the best solution
                                ZBest=f; // Set the objective value as the best value found

                                // Update the upper bound on the problem
                                UpperBound=ZBest;
                end
        end

        P = Branch(P); // Branch on the existing problems to create new problems
```

```
        // Identify active problems and establish a lower bound
        [P, LowerBound]= ActiveProblems(P);
Until (P=Ø or UpperBound-LowerBound=0) //Terminate on condition
```

The initialization procedures for the Implicit Enumeration algorithm are similar to that of the Branch and Bound algorithm presented in section 6.3.3. The exception is that the lower bound (LB) on the original problem is established by the *LowerBound()* function. The *LowerBound()* function implements the following calculation:

$$LB = \sum_{i=1}^{n} min\left\{c_i + \frac{1}{2}q_{ii} + \frac{1}{2}\sum_{j \neq i} min\left(q_{ij}, 0\right), 0\right\} \tag{102}$$

Here, the negative contribution of each decision variable to the objective function, is added to LB (effectively $x_i = 1$) and a positive contribution is omitted (effectively $x_i = 0$). The nested summation considered that the contribution of a variable x_i to the objective function also depends on the variable x_j, since $q_{ij}x_ix_j = q_{ij}$ if both $x_i = 1$ and $x_j = 1$. The term $\underline{q_{ii}}$ is independent of any other decision variable and is separated from the nested summation.

New problems with a smaller feasible region are created in each iteration of the algorithm. The *SelectProblem()* function chooses the next problem to solve. The approach adopted here is that the most recent problem added to *P* is the one that is solved next. This is analogous to a Depth First Search (DFS) on a Branch and Bound tree.

In Implicit Enumeration the definitions of *free variables* and *fixed variables* are essential:

Fixed Variables: variables that have been fixed to either a zero or one value by branching.

Free Variables: variables that have not been fixed to hold a value

Variables are fixed by the *Branch()* function in order from $x_1,..,x_n$. A variable may be fixed to $x_i = 0$ or $x_i = 1$. Every branching procedure fixes one variable at a time to produce two new candidate problems with an additional x_i that is fixed. In the implementation developed in this research every candidate problem has a set of fixed variables and the task is to determine the values of free variables.

Free variables are explored to develop trial solutions by the *SolveRelaxation()* function. The relaxed problem is generated by dropping the linear constraints and setting free variables in a manner that minimises the objective function value. Variable x_i is set in a solution by first completing the following calculations:

$$l_i = c_i + \frac{1}{2}q_{ii} + \sum_{j \neq 1} min(0, q_{ij}) \tag{103}$$

$$u_i = c_i + \frac{1}{2}q_{ii} + \sum_{j \neq 1} max(0, q_{ij}) \tag{104}$$

Here l_i and u_i are the lowest and highest possible contributions of the variable x_i to the objective function. The variable x_i is set by the following rules: if $l_i > 0$ then $x_i = 0$; if $u_i < 0$ the $x_i = 1$.

In Implicit Enumeration, the constraints of the model are temporarily dropped by the *SolveRelaxation()* function. The feasibility of a solution checked by the *FeasibilityCheck()* function after the relaxed problem is solved. The feasibility check is achieved by substituting the solution into the constraints listed as equations 99 and 100. If the solution does not violate the constraints it is feasible. The objective value is computed for all feasible solutions. If the objective value is less than that of the best solution, the current solution is set as the best known solution.

The *Branch()* function, which has already been discussed, is executed prior to punning the set of active problems. If a problem has a lower bound that is higher than the established upper bound, this problem is considered to be fathomed and is pruned. The lower bound on a problem is the objective value that was obtained by solving the relaxation of the parent problem from which the new problem was derived.

The Implicit Enumeration algorithm will continue as long as there a problems to be explored. The algorithm will terminate when the list of problems is empty or the difference between upper and lower bounds is zero.

6.5.4 Example Application of Intra-Cell Machine Assignment Model

The problem documented in Table 6.8 is used to demonstrate the application of the intra-cell machine assignment model. The table documents the backtracking flow values α_k and β_k if the machine in question is not available at the first or second site. Flow volumes established by examining the sequence in which parts require processes and the volumes in which parts are to be produced. The example presented here is abbreviated and a comprehensive demonstration of how α_k and β_k values are determined is presented in the study of Chapter 7.

The cost per intra-cell machine relocation and the cost per part for a backtracking movement are also listed in Table 6.8. At the point of designing a RCMS the capital cost of equipment used to facilitate the backtracking movements can be included into the cost per part for a backtracking move. This will generate solutions that minimize equipment requirements along with other cost bearing factors that are associated with part backtracking.

Table 6.8: Intra-Cell Machine Assignment Problem

	k=1	*k=2*	*k=3*	*k=4*
α_k	4000	3500	2000	1500
β_k	0	3000	3500	1500
r_k	R 3100.00	R 3200.00	R 3300.00	R 3400.00
c_k	R 1.00	R 1.10	R 1.20	R 1.30
N_k	1	2	1	1

Table 6.8 also specifies the number of copies of the machine type that are to be made available in each period. Values of N_k are determined by first applying the inter-cell machine assignment models in order to determine the machines that are allocated to each cell in each period.

In this example problem, the intra-cell machine assignment model consists of 24 variables and 20 constraints. Although the number of variables and constraints are relatively small, the various vectors and matrices that form the model are still large and cumbersome and are not presented here. A program was written in the Matlab® program development environment to automatically populate the *A*, *Aeq* and *Q* arrays and the *b*, *beq* and *C* vectors of equations 98-101. The program code is available on the accompanying CD. The model and solution are also available in a Microsoft® Excel spreadsheet on the accompanying CD.

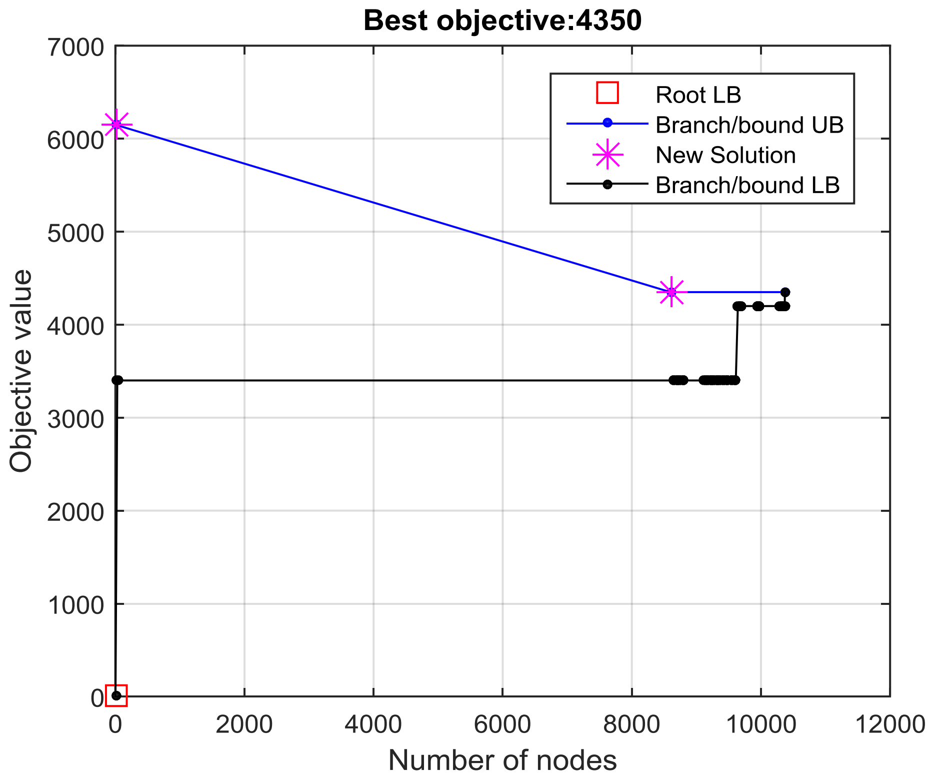

Figure 6.5: Convergence of the Implicit Enumeration Algorithm

The Implicit Enumeration algorithm for the example problem converged after exploring 11765 nodes (candidate problems). The best solution found had a cost of R 4350.00. The convergence plot of the Implicit Enumeration algorithm is shown in Figure 6.5. Note that the root lower bound, which is the lower bound on the original problem, is always zero for the intra-cell machine assignment model when computed with equation 102. It is also easy to establish an upper bound on the model quickly by generating a feasible solution. Feasible solutions may be generated by simply assigning an available machine to the first site and to the second site if an additional machine is available.

Table 6.9: Best Solution Determined by the Implicit Enumeration Algorithm

	x_k	a_k	b_k	y_k	p_k	q_k
k=1	1	1	0	0	0	0
k=2	1	0	0	1	1	0
k=3	0	0	1	1	0	0
k=4	0	0	0	1	0	0

The Implicit Enumeration algorithm for the example problem converged after exploring 11765 nodes (candidate problems). The best solution found had a cost of R 4350.00. The convergence plot of the Implicit Enumeration algorithm is shown in Figure 6.5. Note that the root lower bound, which is the lower bound on the original problem, is always zero for the intra-cell machine assignment model when computed with equation 102. It is also easy to establish an upper bound on the model quickly by generating a feasible solution. Feasible solutions may be generated by simply assigning an available machine to the first site and to the second site if an additional machine is available.

Table 6.9 presents the best solution obtained by the Implicit Enumeration algorithm. A machine is available in the first site for periods $k = 1$ and $k = 2$ and a machine is available at the second site for periods $k = 2$, $k = 3$ and $k = 4$. All machines gains/losses at the respective sites were derived from the inter-cellular movement of machines therefore no reconfiguration cost is charged to the objective function. The cost as determined by the objective function is constituted of part backtracking costs only. This example verifies that intra-cell machine assignment model is consistent with the inter-cell machine assignment models.

6.6 Chapter Summary

This chapter presented two models for the inter-cell assignment of machines and an intra-cell machine assignment model for machine re-layout within a cell. The models are all based on the principle that a machine may only be allocated to a cell if a site is available for it.

A capacity based assignment model was developed, that assigns machines based on the objective of minimizing the surplus and deficit capacity in cells. This model allocates machines to cells within the constraint of a reconfiguration budget. If the reconfiguration budget is generous this model has the blind spot of possibly relocating a machine to a cell in order to gain a miniscule advantage with respect to deficit capacity. A cost minimization assignment model was developed that assigns machines to cells based on the optimal trade-off between reconfiguration cost and the cost of an hour of deficit capacity. Although this model has the ability to find the best economic trade-off, it has the blind spot of preventing a reconfiguration for a miniscule cost saving. The best reconfiguration plan can be determined by comparing the results of both models since each covers the other's deficiency.

The chapter concluded with a model for intra-cell machine assignments. This model has the ability to locate machines in a manner that eliminates part backtracking costs at an optimal trade-off to intra-cell machine relocation costs. This was demonstrated with an example.

7 RCMS Design and Reconfiguration Planning Study

7.1 Chapter Introduction

In this chapter the methods and models developed in Chapters 4-6 will be applied as a study to the design and evolution of a RCMS. The Factory Configuration Template will be developed through the methods that have been researched and found to be suitable. A multi-period reconfiguration plan will then be developed that determines how the configuration of cells will evolve in response to changing production requirements. The purpose of the study is to illustrate the integrated use of individual methods at various stages of the FCT design and reconfiguration planning process. The study will also demonstrate that the RCMS can provide a better solution than a traditional, fixed CMS. The advantage of the RCMS reconfiguration planning models developed in this research, over other RCMS/DCMS models available in literature is that the plans developed here are generated, based on initial design of the factory floor space. This will enable the practical implementation of the plans.

7.2 RCMS Design Problem

A tier one manufacturing company wishes to implement a Reconfigurable Cellular Manufacturing System that will fulfil manufacturing requirements over a six year period. The manufacturer is a small make-to-order business that manufactures products according to the requirements of a larger manufacturing company. During the period of six years the manufacturer is contracted to produce three different products that shall henceforth be known as Product A, Product B and Product C. The product quantities listed in Table 7.1 are to be used in the designed.

Table 7.1: Guideline Quantities for Machine Capacity Calculations

	Product A	Product B	Product C
Year 1	4000	0	0
Year 2	3500	3000	0
Year 3	2000	4500	0
Year 4	0	3200	1500
Year 5	0	1500	3000
Year 6	0	0	5000

The products are assembled from part types and in related quantities as listed in Table 7.2. The relationships between individual parts and the machines that will be used to manufacture them are listed in Table 7.3. The table lists the sequence in which part are to visit machines; the process cycle time per machine is also listed. The cost of inter- and intra-cell machine relocation are listed in Table 7.4. The cost of an inter-cell flow per part and the cost of intra-cell backtracking flow per part is listed. The reconfiguration budget per period is also presented. Note that the cost of unidirectional intra-cell part flow is not presented because this is the desired type of part flow and does not feature in any of the cost minimization models.

Table 7.2: Product Bill of Components

	Product A							
Part Number	1	2	4	6	7	11	14	16
Quantity	1	1	2	1	1	1	2	1
	Product B							
Part Number	3	9	10	13	18	19	20	
Quantity	2	2	1	1	1	1	1	
	Product C							
Part Number	5	8	10	12	15	18	17	
Quantity	1	2	1	1	1	1	2	

Table 7.3: Part Route Sheet

Part Number	Sequence of Machines	Process Cycle Time (minutes per operation)
1	7-8-11-12	7-9-10-10
2	5-1-6-7-9	7-7-7-8-9
3	6-7-8-9	7-7-7-9
4	1-4-6-9	7-10-7-7
5	10-1-7-9-12	10-7-8-8-8
6	5-1-4-6-9	8-9-9-8-9
7	2-8-12-10	10-8-9-10
8	2-6-8-5	8-7-10-8
9	1-4-12-5	9-8-7-7
10	2-3-8-5	9-8-7-10
11	2-8-11-12-10	9-10-8-7-10
12	2-3-4-6-8	10-9-10-7-9
13	1-2-3-4-12	9-8-9-10-9
14	2-8-12-10	10-8-9-9
15	10-1-7-9-12	9-8-8-9-8
16	5-1-3-6-7	7-9-10-8-7
17	1-7-11-12	8-7-9-10
18	2-3-4-12	10-10-9-9
19	6-7-8-9-11	8-7-9-8-9
20	10-2-6-8-9-11	9-9-7-7-8-10

Table 7.4: Finances Governing the RCMS Design Problem

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
r_k (Inter-Cell)	R 4000.00	R 4200.00	R 4400.00	R 4600.00	R 4800.00	R 5000.00
r_k (Intra-Cell)	R 3100.00	R 3200.00	R 3300.00	R 3400.00	R 3500.00	R3600.00
c_k (Inter-Cell)	R 1.50	R 1.60	R 1.70	R 1.80	R 1.90	R 2.00
c_k (Intra-Cell Backtracking)	R 1.00	R 1.10	R 1.20	R 1.30	R 1.40	R 1.50
R_k* (Budget)	R 48	R10	R40	R 10	R 40	R 10

* Budget amounts are represented in thousands of Rand

The cost of inter-cell machine relocation r_k is based on the cost of uninstalling a machine in one cell and relocating it in another cell. In practice, the cost of lost production due to the downtime of both cells may be added to this cost. The cost of intra-cell machine relocation is based on uninstalling a machine and reinstalling it in the same cell. Inter- and intra-cell material handling costs are based on the cost of labour used to perform this task. In practice, the cost of material handling equipment such as dollies may be added to this cost.

The annual operating hours of the manufacturing facility is W_k = 7200 hours made up of 360 working days per year and 20 operating hours per day. Machine availability is assumed to be 80 %. Additional considerations to be factored into the FCT design and reconfiguration plan are:

- the preferred shape for cells is the U-Shaped layout;
- budgeted amounts for reconfiguration in one year may not be carried over to the next;
- all parts are to proceed to a common assembly cell that is based on manual assembly processes, the cell does not have to be designed or reconfigured.

7.3 Factory Configuration Template Design

The FCT is a design for a RCMS; it determines the sites to be allocated to machines and the arrangement of sites on a factory floor. If the FCT is implemented on the factory floor the RCMS is considered to be designed and initialized for future reconfigurations. A nine step procedure was specified in section 5.5 for the creation of a FCT with sufficient information to proceed with reconfiguration planning; these steps will now be applied to the problem at hand.

Step One

The first step is to create a machine-component incidence matrix for all parts and machines for which the RCMS must be designed. The part route sheet presented in Table 7.3 was used to create the machine-component incidence matrix shown in Table F- 1, Appendix F.1. Although each part is active in different periods, all parts are still represented in this matrix.

Step Two

The second step is to perform a silhouette analysis on part clusters generated by the FCMdd algorithm for various values of C. Although the silhouette analysis may not ultimately determine the number of part families and cells to implement, it provides an indicator if the number of naturally existing part clusters in the data set. Recall that the data dimensions are the zero-one column vectors derived from the machine-component incidence matrix.

The silhouette plot is presented in Figure 7.1; this plot was generated by application of the Jaccard dissimilarity measure and the FCMdd algorithm to the clustering of parts. According to Figure 7.1 the case of $C = 2$ has the least outlying parts. Although $C = 5$ has the highest average silhouette value, there are two outlying parts. Moreover, five cells and families is excessive for 12 machines and 20 parts. The RCMS design will proceed with two part families and two cells. Note that the Lance & Williams dissimilarity measure was also tested on this problem and was shown to provide no advantage in solutions over the Jaccard dissimilarity measure.

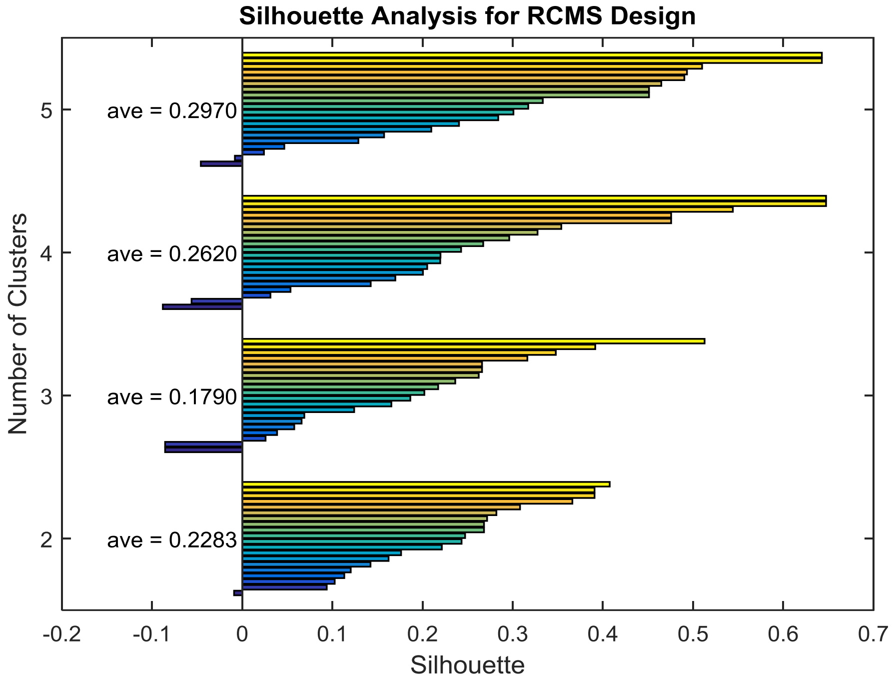

Figure 7.1 Silhouette Analysis for FCT Design for RCMS

Step Three

The third step is to apply the SFC heuristic and generate multiple solutions for part and machine site clusters for various values of parameter *L*. Table 7.5 lists the parameters that were used with the FCMdd algorithm and the SFC heuristic. Recall that the FCMdd algorithm seeds the SFC heuristic with an initial set of part family prototypes and a membership matrix.

Table 7.5: FCMdd Algorithm and SFC Heuristic Parameters

FCMdd Algorithm		**SFC Heuristic**	
Binary Dissimilarity Measure	Jaccard	***L***	0.2-0.5
Maximum Iterations	10	**Maximum Iterations**	50
m	2		
Number of Clusters	2	**Number of Clusters**	2

Both the FCMdd algorithm and the SFC heuristic converged in three or less iterations for all values of L. Table 7.6 lists the exceptional elements and voids obtained for various values of parameter *L*. Recall that *L* is the relative part load for which a machine site can be created in a cell. The method of obtaining exceptional elements and voids as indicators was discussed in section 4.7.

Table 7.6: **Exceptional Elements (EE) and Voids for Various Values of *L***

L	EE	Voids
0.5	25	43
0.4	15	66
0.3	6	85
0.2	0	99

Step Four

The fourth step is to analyse the solutions for various values of *L*. A lower number of exceptional elements indicates increased opportunity to reduce inter-cell part flows within the provisions for machine re-layout made by the FCT. An increasing number of voids indicates an increasing risk that some sites created on the FCT may never be used since these sites were created for low part loads. These loads may be too low to attract a machine when an assignment model is applied. Table 7.6 reveals a high number of exceptional elements for $L = 0.4$ to 0.5. The choice of solution was restricted to $L<0.3$. The problem at hand has zero exceptional elements at $L = 0.2$. This solution was selected for implementation. The part family and cell site clusters for this solution are:

$CS1 = \{S1, S2, S3, S4, S5, S6, S7, S8, S9, S12\};$

$CS2 = \{S1, S2, S6, S7, S8, S9, S10, S11, S12\};$

$PF1 = \{2, 4, 6, 8, 9, 10, 12, 13, 16, 18\};$

$PF2 = \{1, 3, 5, 7, 11, 14, 15, 17, 19, 20\};$

Step Five

The fifth step is to analyse the system level site requirements. This is to ensure that the number of sites made available throughout the system are sufficient to host the mandatory number of machines that will be purchased.

Table 7.7: Mandatory Number of each Machine Type Required per Annum (M=Machine, Y = Year)

	Y1	Y2	Y3	Y4	Y5	Y6
M1	1	1	1	1	1	1
M2	1	1	1	1	1	1
M3	1	1	1	1	1	1
M4	1	1	1	1	1	1
M5	1	1	1	1	1	1
M6	1	1	1	1	1	1
M7	1	1	1	1	1	1
M8	1	1	1	1	1	1
M9	1	1	1	1	1	1
M10	1	1	1	1	1	1
M11	1	1	1	1	1	1
M12	1	1	1	1	1	1

Mandatory numbers of machines will be purchased so that the system has sufficient capacity to meet production requirements. In order to determine the mandatory number to purchase, part volumes to be produced in each year must be determined. Individual part volumes, based on the demand for Product A, B and C, are presented in Table F- 2 , Appendix F.2. A sample calculation is also presented.

The hours H_{ik} of processing time required on each machine based on part volumes and process cycle times is presented in Table F- 3, AppendixF.3. The values of H_{ik} were used to determine the mandatory number of machines of each type required to create sufficient machine capacity. The numbers are listed in Table 7.7; sample calculations are provided in Appendix F.4. Since a maximum of one unit of each machine type is required in all years there is no need to duplicate machine sites in parallel. The sets of cell sites *CS1* and *CS2* remain the same.

Step Six

The sixth step is to analyse the cell level site requirements. At a system level there are sufficient sites to host all machines, nonetheless to demonstrate the method a cell level verification will be performed. The cell level analysis of machine requirements may reveal additional opportunities of reduce the inter-cell flow of parts by introducing sites in parallel. This was demonstrated in the example of section 5.2.4.

The hours H_{ijk} of processing time required on each machine at the cell level are presented in Table F- 5, Appendix F.5. Sample calculations are also shown. The values of H_{ijk} were used to determine the number of machines N_{ijk} of each type required by each cell in each period. The results are presented in Table 7.8 and a sample calculation is presented in Appendix F.6.

Table 7.8: Machines Required per Cell per Year (C=Cell, M = Machines, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
M1	1	0	1	0	1	0	1	1	1	1	0	1
M2	0	1	1	1	1	1	1	1	1	1	1	0
M3	1	0	1	0	1	0	1	0	1	0	1	0
M4	1	0	1	0	1	0	1	0	1	0	1	0
M5	1	0	1	0	1	0	1	0	1	0	1	0
M6	1	0	1	1	1	1	1	1	1	1	1	0
M7	1	1	1	1	1	1	0	1	0	1	0	1
M8	0	1	1	1	1	1	1	1	1	1	1	0
M9	1	0	1	1	1	1	0	1	0	1	0	1
M10	0	1	0	1	0	1	0	1	0	1	0	1
M11	0	1	0	1	0	1	0	1	0	1	0	1
M12	0	1	1	1	1	1	1	1	1	1	1	1

The results of Table 7.8 reveals that each cell requires a maximum of one machine of the various types across all years. The sites made available in clusters *CS1* and *CS2* are sufficient. If one site was allocated to a machine in a cell by the SFC heuristic, but the case arose where the cell level analysis reveals that the cell could benefit from a second site, this additional site may be created in parallel. Note that additional sites should only be created for machines that already have a single site in the cell. Further analysis of the table reveals multiple instances of both cells requiring the same machine at the same time. For example, a site is available for machine type 9 in *CS1* and *CS2*; however, only one mandatory machine of this type will be made available. If additional capital is available to place a machine at both sites this would reduce the inter-cell part flow volume.

Step Seven

The seventh step of the CFT design process is to select a shape for cells. In the design requirements specified in section 7.2, U-shaped cell layouts were the preferred layout shape. The system and cell level site analysis revealed that additional sites were not required in parallel. The U-shaped layouts may therefore be implemented.

Step Eight

The eighth step is to apply the unidirectional flow optimization model and the TS algorithm to determine the optimal sequence in which machine sites should be arranged in cells. In step eight the possibility of duplicate machine sites in series is not considered. The sequence in which operations are to be performed on individual parts is listed in Table 7.3.

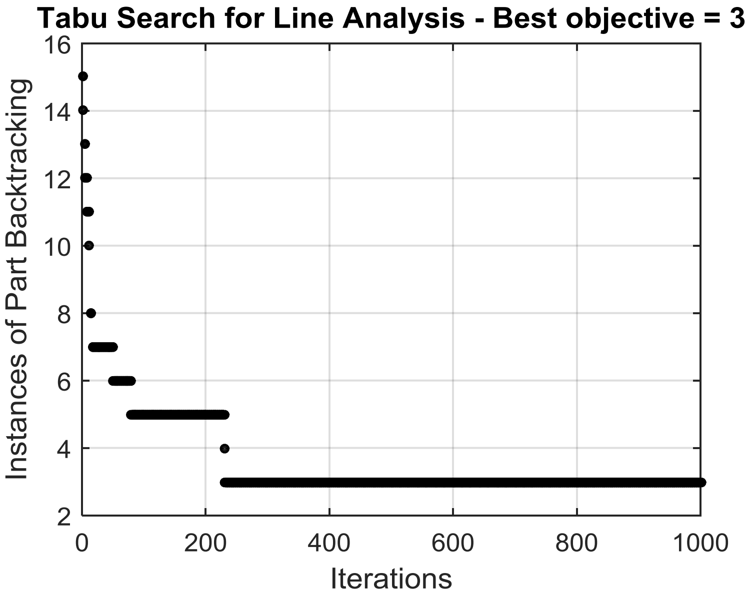

Figure 7.2 TS Algorithm – Best f^* per Iteration – Cell 1

The TS algorithm was applied to *CS1* and *CS2* in order to determine the best arrangement of sites to support unidirectional part flow in Cell 1 and Cell 2 respectively. Figure 7.2 shows the performance of the best solution per iteration for the TS algorithm applied to cell 1. The best solution was found after +200 iterations with a minimum of three instances of part backtracking. Although the algorithm was allowed to continue for 1000 iterations no further improvement was found. The best arrangement of sites for cell 1 is presented in Table 7.9. If this sequence of sites is implemented on the FCT, parts 2, 6 and 16 from the family *PF1* will perform intra-cell backtracking moves in order to have processes completed in the prescribed order.

Table 7.9: TS Results – Best Arrangement of Machine Sites

Location	1	2	3	4	5	6	7	8	9	10
Cell 1	S1	S2	S3	S4	S6	S8	S12	S7	S9	S5
Cell 2	S2	S6	S1	S7	S8	S9	S11	S12	S10	

The best arrangement of sites for Cell 2 is also presented in Table 7.9. If this sequence of sites are implemented on the FCT, parts 5, 15 and 20 from the family *PF2* will perform intra-cell backtracking moves in order to have processes completed in the prescribed order. Figure 7.3 shows that no improvement was found although the TS was allowed to search for more than 800 iterations after this solution was found.

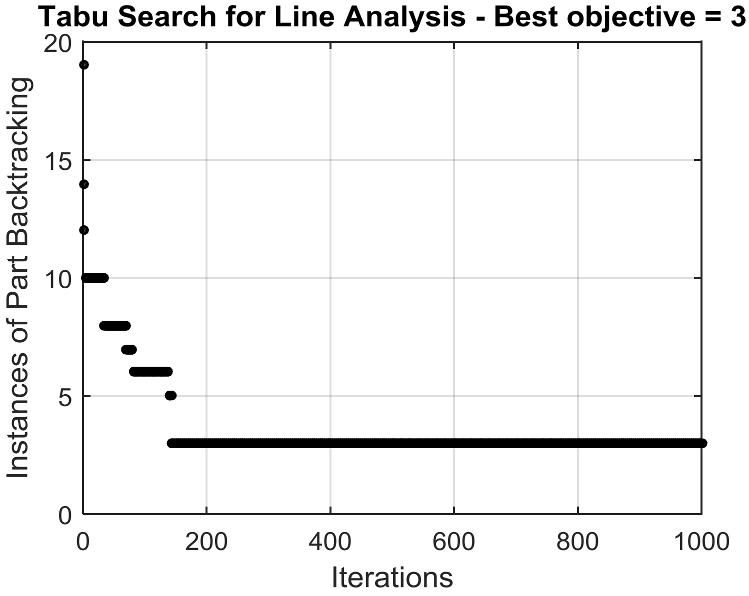

Figure 7.3: TS Algorithm – Best f^* per Iteration – Cell 2

Step Nine

In the ninth step the possibility of additional sites in series is investigated in order to further eliminate cases of the backward flow of parts. The TS algorithm was applied with $X=1$, i.e. one additional site in series. The results for both cells are shown in Table 7.10.

Table 7.10: TS Results – Best Arrangement of Machine Sites with *X=1*

Location	1	2	3	4	5	6	7	8	9	10	11
Cell 1	S5	S1	S2	S3	S4	S12	S6	S8	S5	S7	S9
Cell 2	S10	S2	S1	S6	S7	S8	S9	S11	S12	S10	

Under the proposed arrangement of sites with an additional site in series for machine type 5, all instances of part backtracking have been eliminated in Cell 1. Figure 7.4 shows that the best solution was obtained in under 150 iterations.

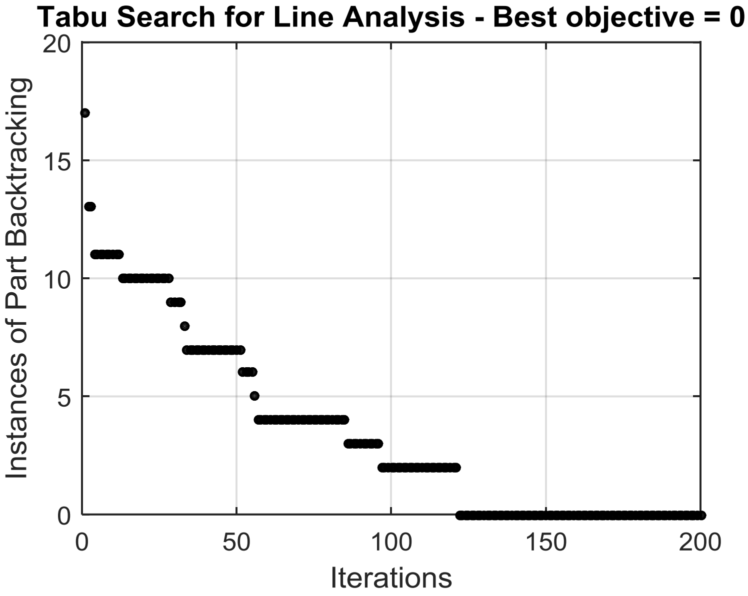

Figure 7.4: TS Algorithm – Best f^* per Iteration – Cell 1

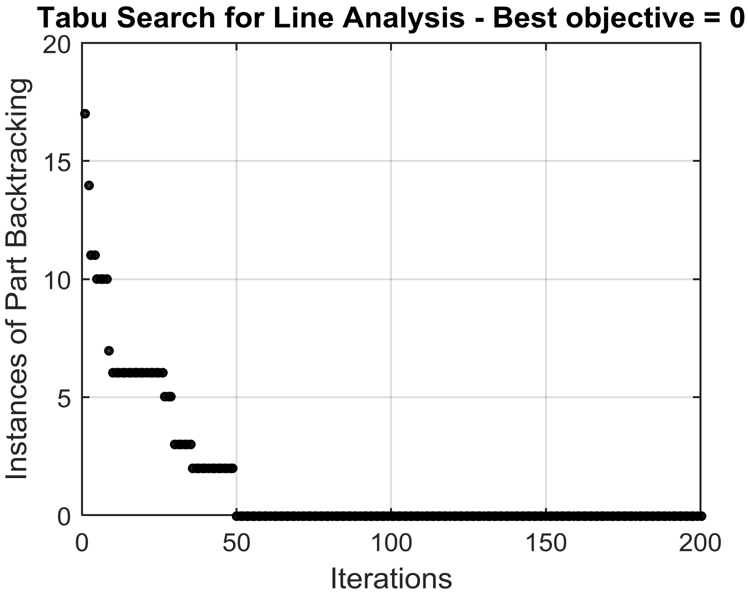

Figure 7.5: TS Algorithm – Best f^* per Iteration – Cell 2

With respect to Cell 2, an additional site for machine 10 in series has also eliminated all instances of part backtracking. Figure 7.5 shows that the best solution was found in under 50 iterations.

Note that the TS algorithm as implemented in the Matlab® development environment, was only programmed to terminate after the prescribed iteration count was complete. The algorithm could be upgraded to terminate upon finding a zero solution, however, this does not add any value to the objectives of this research.

When comparing the results of the TS algorithm for the case of no duplicate sites and the case of duplicate sites in series the interpretation in the RCMS context is important. In a RCMS every site is not occupied by a machine at all times. For the case of no duplicate sites, the result is an upper bound on all possible instances of part backtracking. For the case of one duplicate site in series, the result is a lower bound on all possible instances of part backtracking. The actual instances of part types backtracking will be between 0 - 3.

The Factory Configuration Template

The FCT was created by implementing the sequences of sites, as determined in Table 7.10. The FCT with the necessary information to begin reconfiguration planning is shown in Figure 7.6.

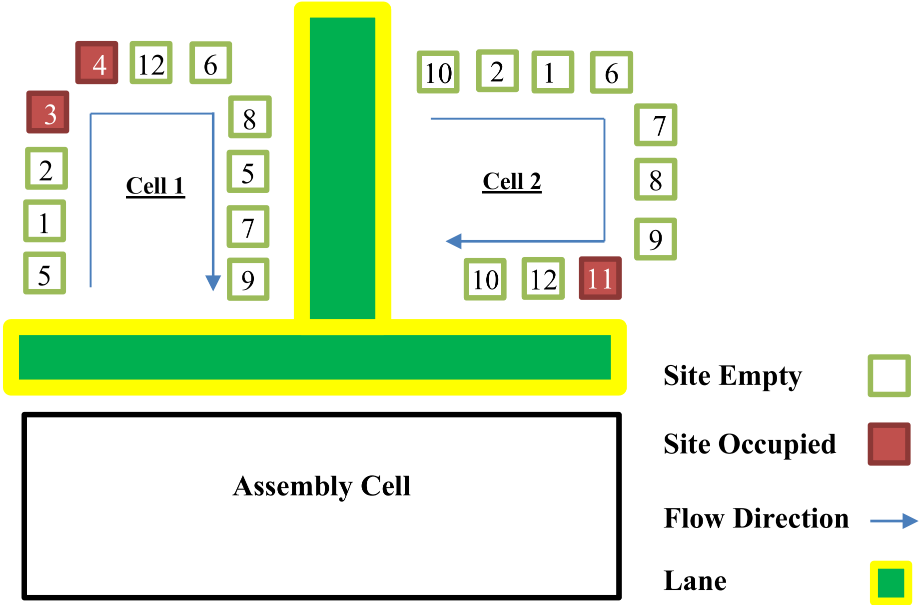

Figure 7.6: Proposed Factory Configuration Template

According to the FCT machine types 1, 2, 6, 7, 8, 9 and 12 may be placed in either Cell 1 or Cell 2. These machines must be strategically positioned to eliminate inter-cell part flow. The assignment of these machines to potential sites in each year must be determined by solving the capacity based assignment model or the cost minimization assignment model. Note that there are no default inter-cell material flows by the design of the FCT since the calculation of exceptional elements based on every site filled was zero. Actual inter-cell material flows will arise once machines have been allocated to sites since the mandatory number of machines to be made available is less than the sites available.

Machine types 5 and 10 have two potential sites each, however, the respective sites are both in the same cell. Machine types 5 and 10 are therefore used for the elimination of part backtracking flow within the respective cells and the location of these machines in each year must be determined by solving the intra-cell machine assignment model.

The number of sites X_{ij} reserved for each machine in each cell is listed in Table 7.11. These values are required by the inter-cell machine assignment models. Machines 3, 4 and 11 have only one space available on the FCT these machines may be allocated to their respective sites without the need for further reconfiguration planning. The placement of fixed machines is indicated by occupied sites on the FCT.

Table 7.11: Sites X_{ij} available for each machine in each cell (C= Cell, M = Machine)

	C1	C2		C1	C2
M1	1	1	**M7**	1	1
M2	1	1	**M8**	1	1
M3	1	0	**M9**	1	1
M4	1	0	**M10**	0	2
M5	2	0	**M11**	0	1
M6	1	1	**M12**	1	1

7.4 Budget Update Prior to Reconfiguration Planning

Prior to solving the inter-cell machine assignment models it is necessary to amend the budget allocated for factory layout. Fixed machines 3, 4 and 11 are placed in their respective cells without having to include these machines in any further analysis. Mobile machines 5 and 6 are also allocated to respective cells without being included in the inter-cell assignment models, since these machines will move within one cell. In the first year the costs of placing these machines is subtracted from the factory layout budget; this leaves a budget of R 28000 remaining in year 1 as shown in Table 7.12. These amounts will be used in the annual reconfiguration budget of the capacity based assignment model.

Table 7.12: Amended Layout/Reconfiguration Budget*

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Factory Layout Budget	R 48	R10	R40	R 10	R 40	R 10
Placement of Machines 3,4,11	R 12	R 0	R 0	R 0	R 0	R 0
Placement of Machines 5, 10	R 8	R 0	R 0	R 0	R 0	R 0
Balance Remaining	R28	R 10	R 40	R 10	R 40	R 10

* Amounts are listed as thousands of Rand

7.5 Inter-Cell Machine Assignment Models

7.5.1 Note - Instances of Inter-Cell Part Moves

Prior to applying the inter-cell machine assignment models, the volumes of parts requiring operations on each machine in each cell was determined, the results are presented in Table F- 6, Appendix F.7. These volumes were calculated to enable an evaluation of the instances of inter-cell part flow that will result if a machine is not assigned to a cell according to the inter-cell machine assignment models.

In this chapter one inter-cell part movement is counted for each machine required by a part, which is not within its resident cell. Upon closer inspection, it may be possible for the part to visit multiple machines that are not present in the resident cell by a single inter-cell move, i.e. combine inter-cell visits. Combined visits are only possible if the sequence in which processes must occur is favourable. Combined visits are not considered here and all calculations on instances of inter-cell moves therefore represent the worst case scenario of one inter-cell move per part requiring a missing machine.

7.5.2 Capacity Based Assignment Model

The capacity based assignment model was used to assign machines to cells in multiple periods. The number of processing hours required of each machine in each cell in each period, i.e. H_{ijk} has been calculated and is listed in Table F- 5, Appendix F.5. The total annual operating hours, W_k is 7200 hours per year and was provided in the problem definition. The cost of removing a machine from a cell with the intention of installing it in another cell, i.e. r_k, was also presented in the problem definition. Recall that the cost r_k is charged by the model when a machine is uninstalled and it is charged again when the machine is installed in another cell. The cost of an inter-cell reconfiguration is therefore $2r_k$; in practice this must be taken into consideration when estimating this cost coefficient.

The machine numbers available for assignment to sites within the FCT was calculated and is listed in Table 7.7. The number of sites reserved for each machine in each cell, X_{ij}, is listed in Table 7.11. These parameters were implemented in the capacity based assignment model as shown in section 6.3. The model was solved with the Matlab® *intlinprog* solver. The model and solution are available in a Microsoft® Excel spreadsheet on the accompanying CD.

The Branch and Bound algorithm converged after exploring 1801836 nodes (candidate problems) as shown in Figure 7.7. The root lower bound on the objective function was 182311 hours of under and overcapacity in cells.

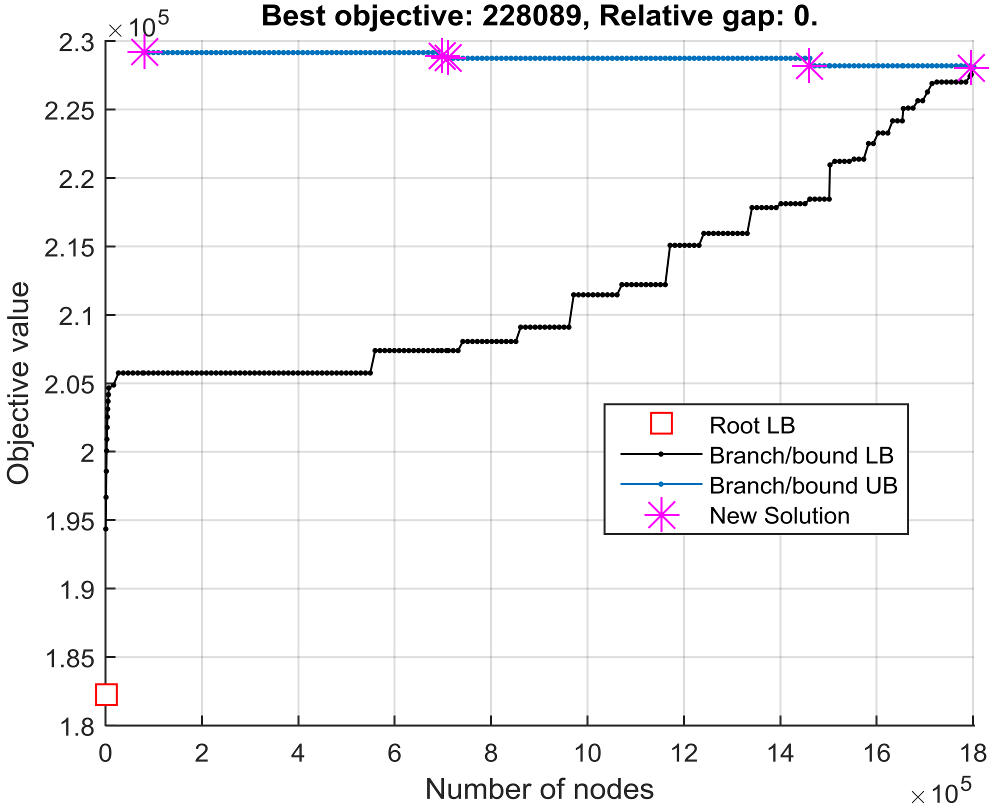

Figure 7.7: Convergence of the *intlinprog* solver for the Capacity Based Assignment Model

The Branch and Bound algorithm converged with a best feasible solution of 228089 hours with no relative gap between the upper and lower bound on the model. Table 7.13 shows the assignment of machines to cells as determined by the *intlinprog* solver. The elements of the table shaded in green indicate the gain of a machine in a cell, red indicates that a machine was removed from a cell. In the first year machines will be allocated to sites to create an initial manufacturing system configuration. Configuration changes are proposed for the second, third and fifth year. No configuration changes are proposed for the fourth and sixth year of operation.

Table 7.13: Six Period Reconfiguration Plan (C=Cell, M = Machine)

	Year 1		Year 2		Year 3		Year 4		Year 5		Year 6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	1	0	1	0	1	0	1	0	0	1	0	1
M2	0	1	0	1	1	0	1	0	1	0	1	0
M6	1	0	1	0	0	1	0	1	1	0	1	0
M7	1	0	0	1	0	1	0	1	0	1	0	1
M8	0	1	0	1	0	1	0	1	1	0	1	0
M9	1	0	1	0	0	1	0	1	0	1	0	1
M12	0	1	0	1	1	0	1	0	0	1	0	1

Table 7.14 presents a summary of the total hours of overcapacity and under capacity exhibited by cells in each year based on the proposed assignment of machines. This table was populated from values of u_{ijk} and v_{ijk} in the final solution; the values per year were calculated by equations 78 and 79. A cell will exhibit hours of under capacity if necessary machines are not allocated to the cell. The same cell will also exhibit hours of overcapacity when more machines of other types are allocated to it than what is required. Hours of overcapacity are generally exhibited since a fraction of a machine cannot be assigned to a cell, the whole machine must be assigned to the cell. The hours of under capacity indicate that parts will travel to cells that exhibit hours of overcapacity in order to access machines. The total under capacity is therefore an indicator of inter-cell part flow.

Table 7.14: Summary of Over and Under Capacity per Year

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Total
Over Capacity (hours)	33798	31989	32432	37207	37375	32399	205200
Under Capacity (hours)	467	7325	6768	3954	3625	750	22889
Total Over/Under Capacity	34265	39314	39200	41161	41000	33149	228089

Based on these machine assignments the volumes of parts that will perform inter-cell part movements are listed in Table 7.15. The volume of inter-cell part flow that was determined by examining in Table F- 6, Appendix F.7. For example, in year 1, the table indicates that 8000 parts require the services of machine 7 in Cell 1 and 4000 parts require the services of machine 7 in Cell 2. The assignment model allocated machine 7 to Cell 1 in year 1, therefore 4000 parts will perform an inter-cell part movement, originating from Cell 2 and flowing to Cell 1.

Table 7.15: Instances* of Inter-Cell Part Flow Originating from each Cell per Year (C=Cell, M=Machine)

	Year 1		Year 2		Year 3		Year 4		Year 5		Year 6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	0	0	0	0	0	0	0	6	4.5	0	0	0
M2	0	0	9	0	0	12.5	0	3.2	0	1.5	0	0
M6	0	0	0	12	10	0	4.5	0	0	6	0	0
M7	0	4	7	0	4	0	0	0	0	0	0	0
M8	0	0	3	0	4.5	0	9.2	0	0	6	0	0
M9	0	0	0	12	8	0	0	0	0	0	0	0
M12	0	0	12	0	0	10	0	6	9	0	5	0
Total	0	4	31	24	26.5	22.5	13.7	15.2	13.5	13.5	5	0

*Instances of inter-cell flow are represented as thousands.

The total instances of inter-cell part flow that may be anticipated if the current RCMS configuration is implemented is 168900 parts at a cost of R 290620.00. The calculation is shown in Table F- 9, Appendix F.8. Table 7.16 shows the planned use of the budget made available for the layout of the manufacturing system. A total of R 110000.00 will be spend on the initial layout and subsequent re-layout if the proposed solution is adopted. The total expected cost of this solution is R 400620.00. The budget remaining may be used for intra-cell reconfiguration.

Table 7.16: Planned Budget Expenditure on Inter-Cell Machine Re-layout*

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 5	Total
Balance Brought Forward	R 28	R 10	R 40	R 10	R 40	R 10	R 138
Planned Use	R 28	R 8.4	R 35.2	R 0	R 38.4	0	R 101.6
Remaining	R 0	R 1.6	R 4.8	R 10	R 1.6	R 10	R 28

* Amounts are presented as thousands of Rand

7.5.3 Cost Minimization Assignment Model

The cost minimization assignment model determines a best economic trade-off between the inter-cell relocation of parts versus the inter-cell relocation of machines. The model does not have the ability to precisely determine the number of parts performing an inter-cell movements, however, the model has the ability to determine the hours of deficit capacity with respect to machine *i* in cell *j* in time period *k*. The hours of deficit capacity are related to inter-cell part movements by means of a cost coefficient C_{ijk}. The values of these coefficients are presented in Table F- 10, Appendix F.9. Sample calculations are also presented. All other model parameters are common to the capacity based assignment model described in section 7.5.

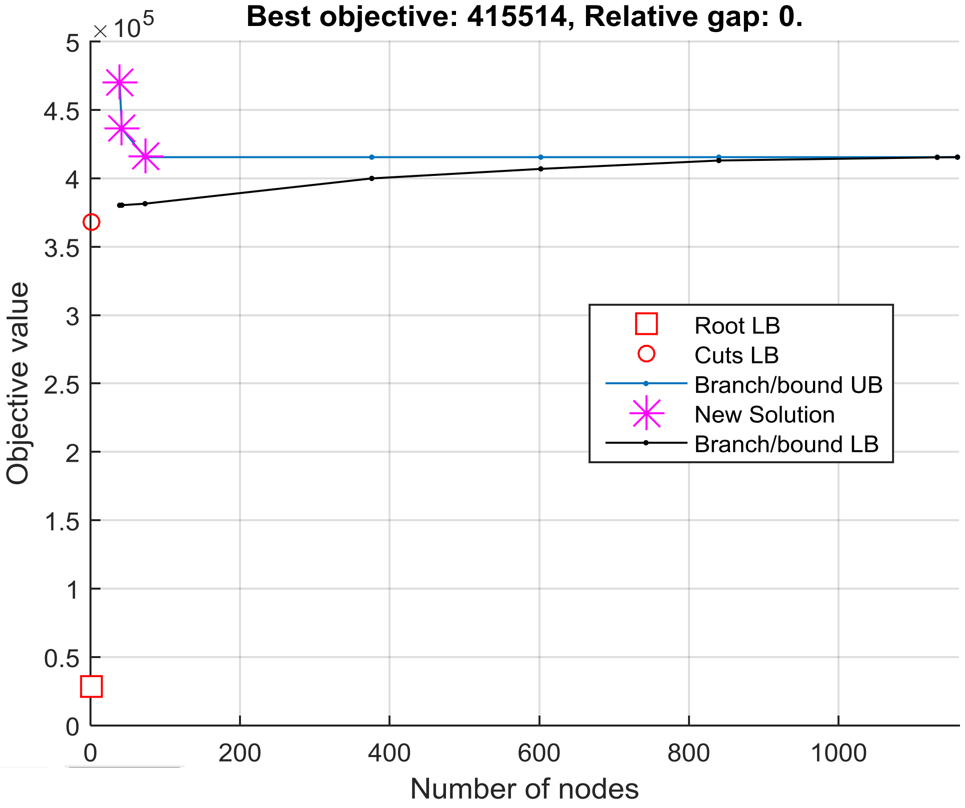

Figure 7.8: Convergence of the *intlinprog* solver for the Cost Based Assignment Model

The model was solved with the Matlab® *intlinprog* solver. The model and solution are available in a Microsoft® Excel spreadsheet on the accompanying CD. The Branch and Bound algorithm with cutting planes converged after exploring 1159 nodes as shown in Figure 7.8. The root lower bound on the objective function was R 28000.00, estimated by relaxing the integer restriction on the variables and solving the original problem as a linear program using the primal-simplex algorithm. The solver applied 23 Gomory cuts, 81 flow cover cuts, 24 strong CG cuts, 29 zero-half cuts, and 24 clique cuts. The lower bound after cut generation was R 368137.48. During experimentation, without cut generation, the *intlinprog* solver would explore up-to 10 million nodes (candidate problems) without convergence between upper and lower bounds. The use of cutting planes greatly reduced the computational effort required to solve this model. The algorithm converged with a best feasible solution at a cost of R 415514.09 with no relative gap between the upper and lower bound.

Table 7.18 shows the assignment of machines to cells as determined by the *intlinprog* solver. In first year machines will be allocated to sites to create an initial manufacturing system configuration. Configuration changes then are proposed for the third and fifth year only. No configuration changes are proposed for the second, fourth and sixth year of operation. A comparison of this reconfiguration plan to that of Table 7.13 reveals that the cost based solution has one less machine relocation. In this solution machine type 7 never occupies the site available for it in Cell 1.

Table 7.17: Alternative Six Period Reconfiguration Plan (C=Cell, M = Machine)

	Year 1		Year 2		Year 3		Year 4		Year 5		Year 6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	1	0	1	0	1	0	1	0	0	1	0	1
M2	0	1	0	1	1	0	1	0	1	0	1	0
M6	1	0	1	0	0	1	0	1	1	0	1	0
M7	0	1	0	1	0	1	0	1	0	1	0	1
M8	0	1	0	1	0	1	0	1	1	0	1	0
M9	1	0	1	0	0	1	0	1	0	1	0	1
M12	0	1	0	1	1	0	1	0	0	1	0	1

Table 7.18 was populated from values of u_{ijk} and v_{ijk} in the final solution; the values per year were calculated by equations 78 and 79. When compared to Table 7.14 a discrepancy occurs between the two solutions in the first year. The cost based assignment model produces an extra 533 hours of overcapacity in one cell and 533 hours of capacity deficit in the other. The cost based solution is therefore expected to produce a larger volume of inter-cell part movements.

Table 7.18: Summary of Over and Under Capacity per Year

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Row Total
Over Capacity (hours)	34331	31989	32432	37207	37375	32399	205733
Under Capacity (hours)	1000	7325	6768	3954	3625	750	23422
Total Over/Under Capacity	35331	39314	39200	41161	41000	33149	229155

Based on these machine assignments, the volumes of parts that will perform inter-cell part movements are listed in Table 7.19. The volume of inter-cell part flow that was determined by examining in Table F- 6, Appendix F.7. A comparison between Table 7.19 and Table 7.15 reveals that 4000 additional parts will perform an inter-cell part movement if this reconfiguration plan is implemented.

Table 7.19: Instances of Inter-Cell Part Movement Originating from each Cell per Year

	Year 1		Year 2		Year 3		Year 4		Year 5		Year 6	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
M1	0	0	0	0	0	0	0	6	4.5	0	0	0
M2	0	0	9	0	0	12.5	0	3.2	0	1.5	0	0
M6	0	0	0	12	10	0	4.5	0	0	6	0	0
M7	8	0	7	0	4	0	0	0	0	0	0	0
M8	0	0	3	0	4.5	0	9.2	0	0	6	0	0
M9	0	0	0	12	8	0	0	0	0	0	0	0
M12	0	0	12	0	0	10	0	6	9	0	5	0
Total	8	0	31	24	26.5	22.5	13.7	15.2	13.5	13.5	5	0

The total instances of inter-cell part flow that may be anticipated if the current RCMS configuration is implemented is 172900 parts at a cost of R 296620.00. The calculation is shown in Table F- 11, Appendix F.10. Table 7.20 shows the planned use of the budget made available for the layout of the manufacturing system.

Table 7.20: Planned Budget Expenditure on Inter-Cell Machine Re-layout*

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 5	Total
Balance Brought Forward	R 28	R 10	R 40	R 10	R 40	R 10	R 138
Planned Use	R 28	R 0	R 35.2	R 0	R 38.4	R 0	R 101.6
Remaining	R 0	R 10	R 4.8	R 10	R 1.6	R 10	R 36.4

* Amounts are presented as thousands of Rand

A total of R 101600.00 will be spent on the initial layout and subsequent reconfiguration if the proposed reconfiguration plan is adopted. The total cost of this solution is R 398220.00. The budget remaining after allocating machines to cells may be used for intra-cell reconfiguration.

Note the discrepancy between the actual cost of the solution R 398220.00, and the cost estimated by the objective function, which is R 415514.09. The percentage error of 4.2 % occurs because of the manner in which the cost coefficients C_{ijk} were estimated with respect to the cost of a deficit hour of capacity. Some small error will always exist as C_{ijk} is approximated on average process cycle times; the percentage error in this example is relatively small.

7.5.4 Selection of an Inter-Cell Reconfiguration Plan

Table 7.21 presents a summary of the important characteristics of each solution. The solution proposed by the cost minimization assignment model has a saving of R 2400.00 if selected above the solution proposed by the capacity based assignment model. On the contrary the capacity based solution has a predicted saving of 4000 inter-cell part movements at the inconvenience of one additional machine relocation. The cost difference is small, relative to the total cost of the solutions and should not dictate the choice.

Table 7.21: Summary of Capacity and Cost Based Reconfiguration Plans

	Capacity Based Solution	Cost Based Solution
Inter-Cell Reconfiguration Cost	R 110000.00	R 101600.00
Inter-Cell Material Handling Cost	R 290620.00	R 296620.00
Total Solution Cost	R 400620.00	R 398220.00
Inter-Cell Part Flow	168900 instances	172900 instances
Inter-Cell Machine Movements	9 instances	8 instances

The choice between the solutions may depend on the point of view as to which would cause the greater inconvenience, i.e. 4000 inter-cell part movements versus one inter-cell machine relocation. From a management perspective, 4000 inter-cell perspective part movements would be more complex to coordinate than one machine relocation. The 4000 inter-cell part movements also imply the need for additional buffering as well as increased work-in-process and longer manufacturing lead times. On the contrary, relocating a machine will present some risk of damage that management may be deemed as an unnecessary risk.

The primary difference between each solution is the assignment of machine 7 in the first year. Two other factors that may be taken under consideration when determining the assignment is to examine the total part-load of each cell and the number of machines within each cell. In this context the total part-load refers to the total number of parts to be manufactured in the cell corresponding to its own part family. Considering part-load, the machine should be placed in the cell with a lower load, so as to reduce the number of parts arriving from other cells at the cell with the higher load. In the case being examined here, both cells have a part load of 20000 each in the first year (see Table F-12, Appendix F.11); therefore the part-load cannot be a deciding factor.

Considering machine numbers, the machine should be placed so as to balance the number of machines in each cell, thereby enabling an even balance of machine operators per cell. This could simplify the management and supervision of each cell. It may also boost worker morale as the workload among personnel would be more even. In the case being examined here, if the capacity based solution is adopted Cell 1 will possess 7 machines and Cell 2 will possess 5 machines. If cell the cost based solution is adopted both cells will possess six machines each.

Based on the advantages of having an even number of machines per cell, as well as minimizing risk by eliminating a machine relocation, the cost based solution will be adopted here. Another advantage of this solution may be observed by examining The Branch and Bound algorithm converged with a best feasible solution of 228089 hours with no relative gap between the upper and lower bound on the model. Table 7.13 shows the assignment of machines to cells as determined by the *intlinprog* solver. The elements of the table shaded in green indicate the gain of a machine in a cell, red indicates that a machine was removed from a cell. In the first year machines will be allocated to sites to create an initial manufacturing system configuration. Configuration changes are proposed for the second, third and fifth year. No configuration changes are proposed for the fourth and sixth year of operation.

Table 7.13, machine 7 occupies the site available in Cell 1 for the first year, thereafter the site remains unused. The alternative reconfiguration plan in Table 7.17 reveals that machine 7 does not occupy the site available in Cell 1 at all. This low site utilization was anticipated since the solution by the SFC heuristic for $L = 0.2$ was adopted; this is a very low admission criteria for a site to be created in a cell.

The advantage of the cost based solution is that the site for machine 7 in Cell 1 may be eliminated from the FCT, thereby reducing the amount of space required to implement the solution. The cost based solution will be accepted and the revised FCT is shown in Figure F- 1, Appendix F.12.

7.6 Intra-Cell Machine Assignment Model

The intra-cell machine assignment model assigned those machines that have multiple sites within a single cell to the exact site they will occupy in each year. The model should only be solved after machines have been assigned to cells by inter-cell machine assignment models. The FCT has two sites available for machine type 5 in Cell 1 and two sites available for machine type 10 in Cell 2. Since one copy of each machine is available, two separate intra-cell machine assignment models will be solved here.

7.6.1 Cell 1 – Machine Type 5

In order to set-up the intra-cell machine assignment model for machine 5, it is necessary to first determine the α_k and β_k flow coefficients. These coefficients are determined by examining the dependency of parts belonging to family *PF1* on machine 5. Consider Table 7.22 which contains the route sequence and of parts and the volumes in which they are anticipated to be produced year on year. The table was constructed from the information available in Table 7.3 and Table F- 2. Consider also, that the sequence in which machine sites are arranged on the FCT is: 5-1-2-3-4-12-6-8-5-7-9.

Table 7.22: Route Sequence and Volumes of Parts Depending on Machine Type 5 (Y=Year)

Part	Sequence of Machine Visitation	Ideal Site	Part Volumes*					
			Y1	Y2	Y3	Y4	Y5	Y6
2	5-1-6-7-9	First	4	3.5	2	0	0	0
6	5-1-4-6-9	First	4	3.5	2	0	0	0
8	2-6-8-5	Second	0	0	0	3	6	10
9	1-4-12-5	Second	0	6	9	6.4	3	0
10	2-3-8-5	Second	0	3	4.5	4.7	4.5	5
16	5-1-3-6-7	First	4	3.5	2	0	0	0

*Part volumes are represented in thousands

By examining the required sequence of machine visitation of each part in comparison to the sequence in which sites are arranged as per the FCT, it is apparent that parts 2, 6 and 16 would benefit from machine 5 being placed in the first site and parts 8, 9 and 10 would benefit from machine 5 being placed in the second site. If machine 5 is not placed at the ideal site, the associated part would have to perform an intra-cell backtracking movement against the preferred direction of part flow in order to access the machine. Parts that would benefit from the machine being placed at the first site are associated with the α_k coefficient and parts that benefit from the machine being placed at the second site are associated with the β_k coefficient.

The association of part volumes with flow coefficients requires an analysis of other mobile machines on which the parts depend. Mobile machine numbers are highlighted in red text in Table 7.22. Consider part 16 which visits machine 1 after visiting machine 5. If machine 1 is not present in the cell, the part would have to perform an inter-cell material movement. In this instance the location of machine 5 would have no influence on part backtracking. In the case where machine 1 is not present in the cell, the volume of part 16 would not contribute to the α_k coefficient. The example of part 16 highlights the significance of examining the assignment of machines to cells, as documented in Table 7.17, in order to accurately determine the value of the coefficients. The following list describes when each of the parts listed in Table 7.22 will contribute to a coefficient:

- Parts 2, 6 and 16 to will contribute to the α_k coefficient when machine 1 is present in the cell, that is year's 1, 2, 3 and 4.
- Parts 8 and 10 will contribute to the β_k coefficient when either machine 8 is present in the cell, which is year's 5 and 6.

- Part 9 will contribute to the β_k coefficient when machine 12 is present in the cell that is year's 3 and 4.

The years in which the volume of each part will not contribute to its respective flow coefficient are shaded in grey in Table 7.22. Table 7.23 documents the α_k and β_k coefficients for the intra-cell machine assignment model of cell 1. The coefficients indicate that the machine should be placed in the first site in year 1 and year 2 and be relocated to the second site in the third year, however, this will be verified by solving the model.

Table 7.23: Flow Coefficients for the Intra-Cell Machine Assignment Model (Cell 1)

	Year 1	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
α_k	12000	10500	6000	0	0	0
β_k	0	0	9000	6400	10500	15000

The cost coefficients for intra-cell machine relocation r_k and intra-cell backtracking c_k were presented in Table 7.4. The model was solved by the Implicit Enumeration algorithm presented in section 6.5.3. A custom implementation of this algorithm was specifically created for solving the. The algorithm converged after exploring 1779 nodes (candidate problems) as shown in Figure 7.9. Individual variable values in the best solution are presented in Table F- 13, Appendix F.13. The model and solution are available in a Microsoft® Excel spreadsheet on the accompanying CD.

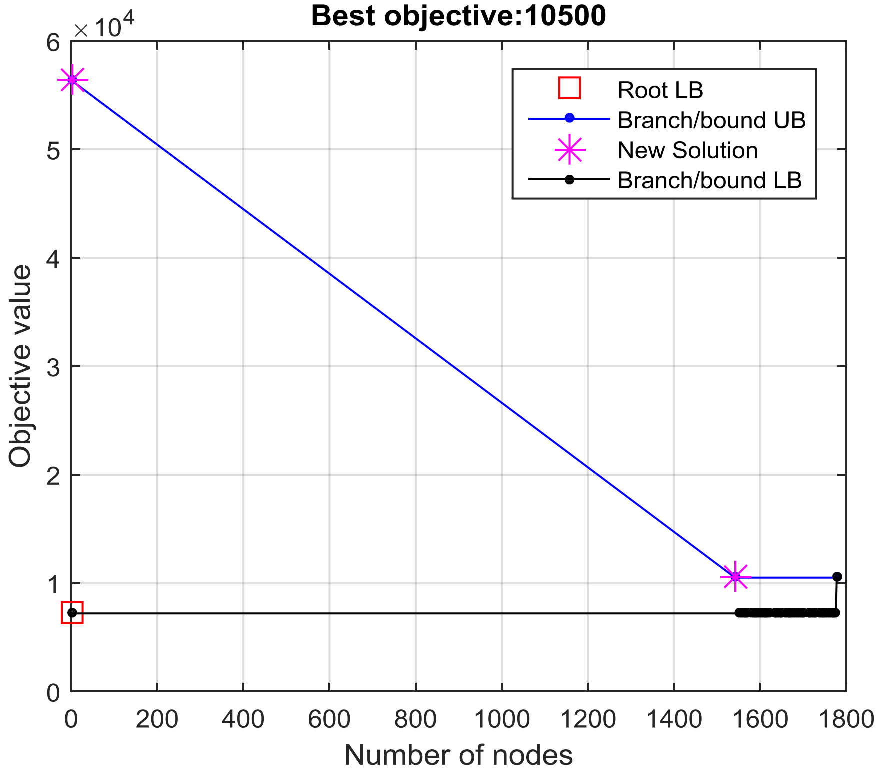

Figure 7.9: Convergence of the Implicit Enumeration Algorithm – Cell 1

The algorithm converged with a best solution cost of R 10500.00. The results verify that machine 5 should occupy the first site in year 1 and year 2 and the second site from year 3 onward. The cost of the solution is made up of the cost of intra-cell backtracking by parts 2, 6 and 16 in the third year, a total of 6000 instances at a cost of R 7200.00. Note that in practice, at the point of designing the system the capital cost of equipment can be integrated into the cost per part per backtracking move. The other cost contributing to the objective function value is the cost of machine relocation within the cell in the third year at R 3300.00.

Recall that a cost is not charged when the machine is placed in the first or second site from an external source, i.e. from other cells or during the placement of machines in the first year. A cost is charged by the objective function only when a machine is swapped between two sites internally. This is because the cost of all inter-cell machine movements are accounted for by the capacity based assignment model and the cost minimization assignment model.

7.6.2 Cell 2 – Machine Type 10

The sequence in which machine sites are arranged in Cell 2, according to the FCT is: 10-2-1-6-7-8-9-11-12-10. Table 7.24 lists the parts that depend on machine 10, which as two available sites in Cell 2. The table was constructed from the information available in Table 7.3 and Table F- 2. The following list describes when each of the parts listed in Table 7.24 will contribute to the α_k and β_k coefficients of the intra-cell machine assignment model:

- Parts 5 and 15 will contribute to the α_k coefficient when machine 1 is present in the cell; that is year's 5 and 6.
- Part 20 will contribute to the α_k coefficient when machine 2 is present in the cell; that is year's 1 and 2.
- Parts 7, 11 and 14 will contribute to the β_k coefficient when machine 8 is present in the cell; that is year's 1, 2, 3 4.

Table 7.24: Route Sequence and Volumes of Parts Depending on Machine 10 (Y=Year)

Part	Sequence of Machine Visitation	Ideal Site	*Part Volumes					
			Y1	Y2	Y3	Y4	Y5	Y6
5	10-1-7-9-12	First	0	0	0	1.5	3	5
7	2-8-12-10	Second	4	3.5	2	0	0	0
11	2-8-11-12-10	Second	4	3.5	2	0	0	0
14	2-8-12-10	Second	8	7	4	0	0	0
15	10-1-7-9-12	First	0	0	0	1.5	3	5
20	10-2-6-8-9-11	First	0	3	4.5	3.2	1.5	0

*Part volumes are represented in thousands

Table 7.25 documents the α_k and β_k coefficients for the intra-cell machine assignment model of Cell 2. The cost coefficients for intra-cell machine relocation and part backtracking were presented in Table 7.4.

Table 7.25: Flow Coefficients for the Intra-Cell Machine Assignment Model (Cell 2)

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
α_k	0	3000	0	0	6000	10000
β_k	16000	14000	8000	0	0	0

The model was solved by the Implicit Enumeration algorithm presented in section 6.5.3. The algorithm converged after exploring 18665 nodes (candidate problems) as shown in Figure 7.10. Individual variable values in the best solution are presented in Table F- 14, Appendix F.13. The model and solution are available in a Microsoft® Excel spreadsheet on the accompanying CD.

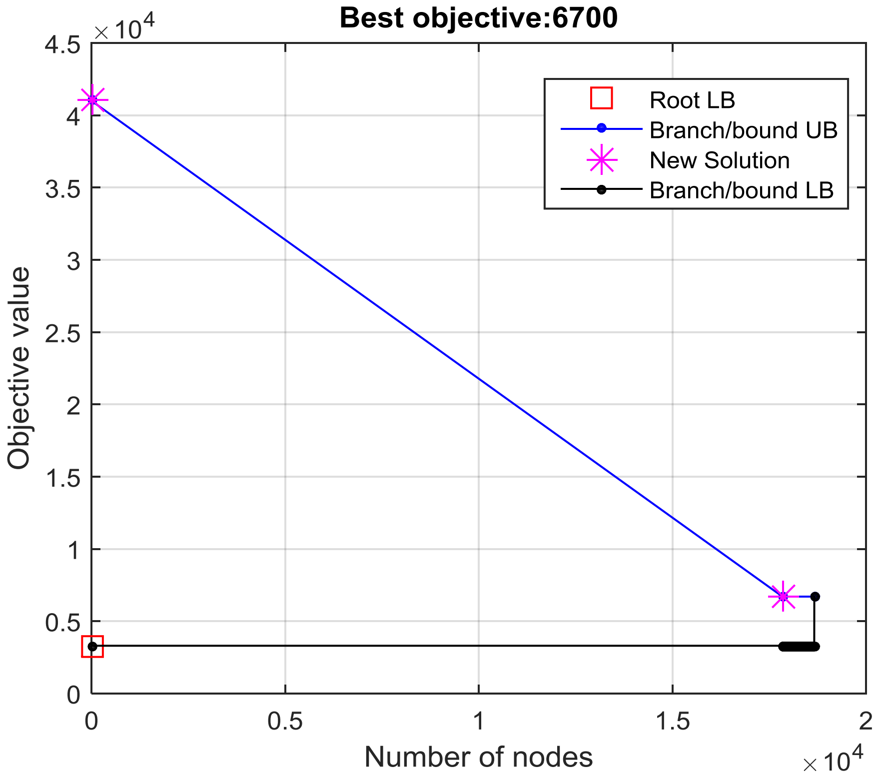

Figure 7.10: Convergence of the Implicit Enumeration Algorithm – Cell 2

The algorithm converged with a best solution cost of R 6700.00. The model assigned machine 10 to the second site in years 1, 2 and 3. The machine is then relocated to the first site in years 4, 5 and 6. The machine is relocated in the fourth year due to the lower cost of reconfiguration, although no part flow advantage is gained since $\alpha_4 = 0$. In practice, this is a decision that may be reviewed and the machine may be relocated in the fifth year when the relocation becomes advantageous.

The cost of the solution is made up of the cost of intra-cell backtracking by part 20 in the second year; a total of 3000 instances at a cost of R 3300.00. The other cost contributing to the objective function value is the cost of machine relocation within the cell in the fourth year at R 3400.00.

7.6.3 Update of Reconfiguration Expenditure

Table 7.26 presents the balance of the budget brought forward after the planned spending on inter-cell re-layout is accounted for. The cost of intra-cellular re-layout is subtracted from the balance brought forward to reveal if any budget deficit occurs. In this instance a budget surplus of R 34100.00 exists after all reconfiguration (re-layout) costs are accounted for; furthermore there is no occurrence of over expenditure in any year.

The reconfiguration plans proposed by the cost minimization assignment model and the intra-cell machine assignment model may be approved and implemented from a financial planning perspective.

Table 7.26: Planned Budget Expenditure on Intra-Cell Machine Re-layout*

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 5	Total
Balance Brought Forward	R 0	R 10	R 4.8	R 10	R 1.6	R 10	R 36.4
Planned Use	R 0	R 0	R 3.3	R 3.4	R 0	R 0	R 6.7
Remaining	R 0	R 0	R 1.5	R 6.6	R 16	R 10	R 34.1

* Amounts are presented as thousands of Rand

7.7 Combined Reconfiguration Plans

The configurations of manufacturing cells to be implemented over six years based on the anticipated product demand is presented in Table 7.27.

Table 7.27: Configuration of Manufacturing Cells ([ ] = Empty Site)

Location	Sequence of Machines										Site Utilization
	1	2	3	4	5	6	7	8	9	10	
Cell 1 – Year 1	5	1	[ ]	3	4	[ ]	6	[ ]	[ ]	9	6/10
Cell 1 – Year 2	5	1	[ ]	3	4	[ ]	6	[ ]	[ ]	9	6/10
Cell 1 – Year 3	[ ]	1	2	3	4	12	[ ]	[ ]	5	[ ]	6/10
Cell 1 – Year 4	[ ]	1	2	3	4	12	[ ]	[ ]	5	[ ]	6/10
Cell 1 – Year 5	[ ]	[ ]	2	3	4	[ ]	6	8	5	[ ]	6/10
Cell 1 – Year 6	[ ]	[ ]	2	3	4	[ ]	6	8	5	[ ]	6/10
Cell 2 – Year 1	[ ]	2	[ ]	[ ]	7	8	[ ]	11	12	10	6/10
Cell 2 – Year 2	[ ]	2	[ ]	[ ]	7	8	[ ]	11	12	10	6/10
Cell 2 – Year 3	[ ]	[ ]	[ ]	6	7	8	9	11	[ ]	10	6/10
Cell 2 – Year 4	10	[ ]	[ ]	6	7	8	9	11	[ ]	[ ]	6/10
Cell 2 – Year 5	10	[ ]	1	[ ]	7	[ ]	9	11	12	[ ]	6/10
Cell 2 – Year 6	10	[ ]	1	[ ]	7	[ ]	9	11	12	[ ]	6/10

The table was generated by combining the proposed plans for inter- and intra-cell reconfiguration. If there are no significant changes to the product demand and the types of parts that will enter and leave the system over the six year period these proposed configurations may be implemented. If changes occur, the models may be solved again with updated information. The next section will address how changes may be accounted for in order to determine adjusted cell configurations.

An examination of Table 7.27 reveals that across the six years both cells are expected to have six out of ten (60 %) of sites being used at any time. The solution as presented here includes 172900 inter-cell part movements, 9000 inter-cell part backtracking movements, 8 inter-cell machine relocations and 2 intra-cell machine relocations.

A total budget of R 158000.00 was made available at the outset for the placement of machines at sites across the six years. A budget surplus of R 34100.00 remained after all reconfigurations were accounted for. This implies that the total cost of factory layout and subsequent re-layout is R123900.00. The cost of all inter-cell material handling movements totals R 296620.00 and the cost of all intra-cell part backtracking movements totals R 10500.00. The total solution cost is therefore R 431020.00. Note that this figure includes the total cost of factory layout but does not account for the cost of the preferred type of part flow which is unidirectional flow contained within individual cells.

7.8 Strategies for Planning with Updated Information

7.8.1 Sensitivity of Solutions to Cost

In reconfiguration planning uncertainty increases with respect to parameter estimation over an extended number of periods. Changes with respect to the cost coefficients listed in Table 7.4 have the ability to change the optimal solution. Sensitivity analysis is useful for determining the degree of change to cost parameters for which the current solution remains optimal. The Cost Based Assignment model, which is a pure integer programming model will be examined here.

In linear programs the relationship between cost parameter changes in the objective function and the optimal solution can be easily determined. In integer programming the relationship between cost parameter changes and the optimality of a solution is disjointed. Sensitivity analysis was therefore performed by perturbing cost coefficients and re-solving the model, as opposed to exploiting the primal-dual relationships used in linear programming. For the Cost Based Assignment model, the costs of cells gaining/losing a machine could be decreased by as much as 32% or increase by as much as 68 % with the current machine assignments remaining optimal. The cost of an hour capacity deficit could be decreased by as much as 40 % or increase by as much as 47% with the current machine assignment remaining optimal.

Changes to cost parameters outside of these ranges or changes to other parameters are handled by the strategies presented in the next section.

7.8.2 Generation of Updated Reconfiguration Plans

During the lifecycle of the RCMS, changes to product/part demand and the types of parts in the system may change. New process technology may become available and the types of machines that are made available to the system may also change. As the system begins to generate a profit, management may opt to purchase additional machines. The following strategies may be used in order to generate manufacturing system configurations that are optimal based new information.

Case 1: Machine types and part types remain the same but product demand, available numbers of machines, or estimated costs change:

1. Re-compute the hours of machine capacity required by each cell (i.e. H_{ijk}).
2. Update cost parameters and machine numbers.
3. Solve the inter-cell machine assignment models; use the current state of the manufacturing system as the initial system configuration in these models. This may be achieved by fixing relevant variables by adding constraints to the model. For example, the constraint $x_{121} = 1$ will ensure that strictly one copy of machine type 1 will be placed in Cell 2 in year 1.
4. Use the results of the assignment models to estimate new flow α_k and β_k coefficients for the intra-cell machine assignment models.
5. Solve the intra-cell machine assignment models; fix the current location of relevant machines as the initial location in this model. This may be achieved by fixing relevant variables by adding constraints to the model. For example, the constraint $x_1 = 1$ will ensure that the machine is placed at the first site in the sequence of machines in year 1.

Case 2: Product design changes and a new part type is added to the system:

1. Add the new part to the family with the greatest similarity. The Jaccard similarity index presented in equation 1, combined with SLC, ALC or CLC clustering may assist with this decision.
2. Follow procedures 1-5 as listed in *Case 1* to determine the final set of system configurations.

Case 3: A new machine type is added to the system:

1. Determine if a site should be created in each of the manufacturing cells. A site may be added to the cell if the corresponding part family has sufficient need of the machine. The need may be determined by examining the number of part types and part volumes that require the services of the machine.

2. Determine the location of a new site for the machine within the relevant cell. This is determined by identifying sites that are no-longer in use and at an appropriate location in the sequence of sites, to best accommodate the needs of the part family. Once viable sites have been identified they may be repurposed by relevant physical alterations to be able to accommodate the new machine.
3. Follow procedures 1-5 as listed in *Case 1* to determine the final set of system configurations.

No example of adjusting the planned system configurations according to updated information will be presented here.

7.9 Comparison of Reconfigurable and Fixed CMS

In order to validate the concept of reconfigurable manufacturing cells, a comparative analysis will be conducted here on the performance of reconfigurable system versus a system with a fixed layout. A fixed cellular manufacturing system design has been created for the problem of section 7.2. The fixed system was designed using the SFC heuristic, together with the Jaccard dissimilarity coefficient in order to determine the clusters of machine cells and part families. One machine of each type was made available, as in the case of the RCMS. The line analysis was conducted using the Tabu Search Algorithm presented in section 5.3.2. The group analysis, line analysis and other calculations pertaining to the design of the fixed CMS is presented in Appendix G.1-G.3. The methods used for the RCMS were also used for the fixed CMS since they have the flexibility for single or multi-period cell formation.

Table 7.28 presents a cost comparison of RCMS and fixed CMS solutions. The layout cost of the fixed CMS was based on the placement of 12 machines at a cost of R 4000.00 per machine in the first year. The cost of inter-cell part flow for the fixed CMS is presented in Table G- 4 and the cost of intra-cell part backtracking flow is presented in Table G- 8.

The cost difference between the fixed and reconfigurable designs is R 241930.00. The cost of the RCMS solution is 67 % of the cost of the CMS with a fixed layout, which is a cost saving of 33 %. The reason why the RCMS outperforms the traditional type of CMS is because the RCMS has the ability to relocate machines to negate inter-cell part flow.

Table 7.28: Cost Comparison of Reconfigurable and Fixed CMS

	Reconfigurable CMS	**Fixed CMS**
Factory Layout Cost	R 123900.00	R 48000.00
Inter-cell part flow cost	R 296620.00	R 526200.00
Intra-cell part back flow cost	R 10500.00	R 71750.00
Total Cost of Solution	R 431020.00	R 645950.00

Table 7.29: Space Requirements and Flow Characterises of RCMS and Fixed CMS

	Reconfigurable CMS	Fixed CMS
Space required	20 sites	12 sites
Inter-cell flow	172900 instances	302500 instances
Intra-cell back flow	9000 instances	66500 instances

Table 7.29 lists the space requirements and flow characteristics of fixed and reconfigurable CMS. The inter-cell flow volumes for the fixed CMS are calculated in Appendix G.2 and the intra-cell flow volumes are calculated in Appendix G.3. According to Table 7.29 the RCMS requires 8 more sites for machines placement than the traditional fixed CMS, this implies a larger factory with a larger cost of space. This is a factor that must be taken under consideration in practice when determining the economic benefits of the RCMS. The results of Table 7.27 revealed that only 60 % of the sites were used at any moment in time. Unutilized sites may be used to set-up buffers within the cell. The static CMS requires more buffers than the RCMS due to its undesirable part flow characteristics. When the placement of buffers within the cell is considered, the space requirements of a RCMS and a fixed CMS may be similar, however, the investigation of this is a topic for future research.

The fixed CMS has 1.75 times inter-cell part flow of the RCMS and 7.38 times the intra-cell part backtracking flow. This makes it much more difficult to manage from the perspective of job scheduling and flow control. The fixed CMS is also expected to have significantly more work-in--process and longer manufacturing lead times than the RCMS.

Table 7.30: GT Performance Measures for Reconfigurable CMS

	Group Efficiency	Group Efficacy	Weighted Average Efficacy	Grouping Measures	Exceptional Elements	Voids
Year 1	0.83	1.55	0.68	0.65	2	14
Year 2	0.71	1.71	0.5	0.37	15	37
Year 3	0.68	1.79	0.46	0.29	18	40
Year 4	0.72	1.65	0.51	0.38	12	29
Year 5	0.72	1.65	0.51	0.38	12	29
Year 6	0.85	1.58	0.7	0.68	1	12

Measures of performance in the context of Group Technology, were calculated and documented in Table 7.30 and Table 7.31, sample calculations are provided in Appendix F.14 and Appendix G.4 respectively. A comparison between measures of performance would reveal that year on year, the RCMS generally outperforms the static CMS with higher values (except in the instance of the void count in which case a lower value is preferred).

Table 7.31: GT Performance Measures for Fixed CMS

	Group Efficiency	Group Efficacy	Weighted Average Efficacy	Grouping Measures	Exceptional Elements	Voids
Year 1	0.71	1.75	0.5	0.36	8	20
Year 2	0.7	1.73	0.49	0.34	16	38
Year 3	0.7	1.73	0.49	0.34	16	38
Year 4	0.7	1.68	0.49	0.35	13	30
Year 5	0.7	1.68	0.49	0.35	13	30
Year 6	0.7	1.85	0.49	0.35	7	18

Note that these GT measures of performance do not include part volumes in the calculations. In the third year the fixed CMS outperforms the RCMS in measures such as group efficiency, weighted average group efficacy and others. When examining the machine-component incidence matrices the primary difference between the fixed CMS configuration and the reconfigurable CMS configuration is that in the fixed case, machine 6 is in Cell 1 and machine12 is in Cell 2.

The part load calculation on each machine presented in Table F- 6 is applicable to both fixed and reconfigurable CMS since the both solutions have identical part families. An inspection of the table in the third year reveals that the higher load on machine 6 comes from Cell 2 with 18000 parts as compared to Cell 1 with 10000 parts. Likewise, an inspection of the table in the third year reveals that the higher load on machine 12 comes from Cell 1 with 18000 parts as compared to Cell 2 with 10000 parts. Although the GT measures show that the fixed CMS solution performs better than the reconfigurable solution, the reconfigurable solution is actually the better choice in the third year.

7.10 Reconfiguration without a FCT

In the literature review it was found that the term RCMS and DCMS were used interchangeably by researchers to describe a cellular system that underwent configuration changes in cells by the re-layout of machines over multiple periods. The models and algorithms used to propose configuration changes did not consider an initial system design for future reconfiguration.

The concept of a Factory Configuration Template was proposed in this research; the FCT is a design for the factory floor that facilitates future reconfiguration. In order to compare the reconfiguration plans of a RCMS based on the FCT concept to other DCMS methods, assume that all machines have the same footprint, allowing any machine to occupy any site. If this assumption is correct then the FCT concept is at a disadvantage since the FCT allocates dedicated sites to specific machines. In the study developed in this chapter, the FCT contained 10 sites per cell. If the FCT concept is abandoned and any machine can occupy any site; a maximum of six sites are necessary.

Table 7.32: Cell Arrangements without FCT Restrictions

	Sequence of Machines					
Location	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1 – Year 1	5	1	3	4	6	9
Cell 1 – Year 2	5	1	3	4	6	9
Cell 1 – Year 3	1	2	3	4	12	5
Cell 1 – Year 4	1	2	3	4	12	5
Cell 1 – Year 5	2	3	4	6	8	5
Cell 1 – Year 6	2	3	4	6	8	5
Cell 2 – Year 1	2	7	8	11	12	10
Cell 2 – Year 2	2	7	8	11	12	10
Cell 2 – Year 3	6	7	8	9	11	10
Cell 2 – Year 4	10	6	7	8	9	11
Cell 2 – Year 5	10	1	7	9	11	12
Cell 2 – Year 6	10	1	7	9	11	12

Table 7.32 presents the cell arrangements with unused sites removed. This table was developed from Table 7.27 under the assumption that any machine could occupy any site, thereby allowing the removal of unused sites. The order in which machines should appear to support maximum unidirectional flow has been maintained. The cell configurations as presented in Table 7.32 will therefore have identical inter-cell part flow volumes and intra-cell part backtracking volumes as the RCMS configurations developed in previous sections. Although the machine types and sequence in which machines appear are identical, the cost of achieving these machine layouts will not be the same.

According to Table 7.32, the following configuration changes are implemented in Cell 1:

Year 1: six machines are gained during the initial layout of the manufacturing system.

Year 2: no changes are proposed for Cell 1.

Year 3: machines 6 and 9 are removed, machines 2 and 12 are gained and machines 1 and 5 have been relocated internally.

Year 4: no changes are proposed to Cell 1.

Year 5: machines 1 and 12 are removed, machines 6 and 8 are gained and machines 2, 3 and 4 have been relocated internally.

Year 6: no changes are proposed for Cell 1.

According to Table 7.32, the following configuration changes are implemented in cell 2:

Year 1: six machines are gained during the initial layout of the manufacturing system.

Year 2: no changes are proposed for cell 2.

Year 3: machines 2 and 12 are removed, machines 6 and 9 are gained and machine 11 is relocated internally.

Year 4: machines 6, 7, 8, 9, 10 and 11 are relocated internally.

Year 5: machines 6 and 8 are removed, machines 1 and 12 are gained and machines 9 and 11 are relocated internally.

Year 6: no changes are proposed for cell 2.

Table 7.33 summarises the total number of machines gained and removed and from cells and the cost of machine relocation at an inter-cell level. Table 7.34 summarises the total number of machine relocations inside cells and the associated cost. The total cost of initial machine placement and inter-cell machine relocation is R121600.00. Total cost of intra-cell machine relocation is R47800.00. Total cost of re-layout if the FCT is abandoned and any machine is allowed to occupy any site is R149400.00. In the RCMS design based on the FCT concept, the cost of machine re-layout was R 123900.00. The cost of re-layout without the FCT is 20.6 % higher, this is due to 12 additional intra-cell machine relocations.

Table 7.33: Cost of Inter-Cell Machine Relocation

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Machines Gained/Removed	12	0	8	0	8	0
Cost per Machine Gained/Removed	R 4000.00	R 4200.00	R 4400.00	R 4600.00	R 4800.00	R 5000.00
Cost Per Year	R 48000.00	R 0.00	R 35200.00	R 0.00	R 38400.00	R 0.00

Table 7.34: Cost of Intra-Cell Machine Relocations

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Machines Gained/Removed	0	0	3	6	5	0
Cost per Internal Relocation	R 3100.00	R 3200.00	R 3300.00	R 3400.00	R 3500.00	R 3600.00
Cost Per Year	R 0.00	R 0.00	R 9900.00	R 20400.00	R 17500.00	R 0.00

In general a RCMS that allows any machine to occupy any site will require additional machine relocations in order to obtain the same flow characteristics as a RCMS with a FCT design that includes dedicated sites for specific machines. Nonetheless dedicates sites for machines, according to a FCT design, inflates the floor space requirements of a RCMS.

At this point, the trade-off may appear to be additional machine relocations versus additional sites for machines. Returning to the assumption that all machine have the same floor space requirements; in reality this will not be true for all machines. Moreover, some machines will have specific requirements for supplies of water, compressed air, etc. that will prevent any machine from being allocated to any site.

7.11 Chapter Summary

This chapter demonstrated the use of individual algorithms and methods in an integrated manner for the design of a Factory Configuration Template. A FCT was developed for a RCMS with twelve machines and three products that are to be manufactured by contract with specified volumes over a six year period.

The changing product demand caused the mix of parts and the part volumes to vary over the six year period. The objective of the FCT design was to create an arrangement of sites for machines in cells that facilitates the advantageous re-layout of machines across multiple periods.

Inter-cell and intra-cell machine assignment models were used to assign machines to sites in cells over the six year period thereby creating a plan for reconfiguration. A comparison between the RCMS and a fixed CMS was conducted; the reconfigurable layout produced significantly less inter-cell part movements and intra-cell part movements. The RCMS was also shown to be more economical to operate than the fixed CMS.

The generation of reconfiguration plans within the context of a FCT is what distinguished this work from other DCMS reconfiguration planning methods. The chapter concluded with an analysis of the benefits of performing reconfiguration within the context of an FCT. The disadvantage of the FCT concept was that only 60 % of the sites in cells were used at any instant in time. On the contrary the additional sites require that less machines be relocated internally in order to facilitate maximum unidirectional flow in cells.

8 Discussion

8.1 Chapter Introduction

In the competitive global market companies frequently introduce new products in order to gain a larger market share. These new products usually include updated technology and have different manufacturing requirements. The frequent introduction of new products also implies that the production lifecycles are short. Frequent product changes and changes in market demand require manufacturing systems to be more responsive in accommodating change.

Cellular Manufacturing Systems are a central concept in achieving lean manufacturing and just-in-time manufacturing [4]. The benefits of CMS include reduced part set-up times on machines, lower work-in-process levels, reduced material handling costs, less chaotic work flow, etc. [3, 8]. The disadvantage of adopting a cellular layout is that the configuration of cells will only remain optimal for a specific part mix and demand [7]. The concept of Dynamic CMS has been around since the 1990's, which advocates the change of cell configurations to accommodate product changes [13]. Case studies have been reported in CMS literature where the re-layout of the system and changes to the configurations of cell clusters were beneficial in coping with change [72]. These changes were not planned in advance at the point of installing the CMS.

Since the early works on DCMS by Rehault et al [13] other researches such as Safaei et al [58-60] and Bayram and Sahin[62] have developed multi-period reconfiguration plans that specify how the configurations of cells should evolve over time. The literature also showed that researchers have used the terms RCMS and DCMS interchangeably to refer to systems where cell configurations change. A gap found in existing DCMS/RCMS literature was the lack of an initial system design for future configuration changes. The lack of an initial system design for future changes to cells may lead to the solutions as proposed by optimization models being practically infeasible to implement.

Researchers such as Koren et al [1], and Malhotra et al [94] have emphasised the concept of design for reconfigurability at the outset in order to accommodate product and market changes. The research and development of a method of imparting reconfigurability to a CMS at the design stage was an objective of this research. The concept of a Factory Configuration Template was developed in this research. The FCT is a design of the factory floor space which includes provisions for future configuration changes. Models for inter-cell and intra-cell reconfiguration planning were developed that generate configuration changes within the allowances made by the FCT.

The merits of the FCT concept and the methods used for its design will be discussed in this chapter. The models that have been developed for inter and intra-cell reconfiguration will also be considered with respect to their ability to propose suitable reconfiguration plans.

8.2 Elements of FCT Design

A Factory Configuration Template is a design of the factory floor space in a cellular system. FCT concept was proposed to enable the future re-layout of a CMS. A FCT design may be designed with different levels of information. In this research methods were proposed to create templates that specified:

4. the number of cell in the system;
5. the layout shapes of cells;
6. sites for machine types in cells;
7. machine types requiring duplicate sites in parallel;
8. machine types requiring duplicate sites in series;
9. the arrangement of machine sites within cells;
10. the flow direction of parts within cells;
11. machines that will move in the future to create new configurations;
12. machines that will remain fixed in position irrespective of configuration changes.

The generation of these types of information required the application of nine steps that used various heuristics, calculations and optimization algorithms. This formed a significant portion of the analytical contribution of this research.

The rudimentary aspects of FCT design that do not require the application of optimization models and algorithms include the specification of the following information:

1. space requirements of individual machines and cells;
2. safe distances between machine sites;
3. safe working spaces for humans inside sites;
4. lanes with sufficient width for machine relocation;
5. supplies of water, electricity, compressed air, etc. for machines at sites.

Due to the rudimentary nature of generating this type of information, these aspects of FCT design were not given significant attention in this research. A further aspect of FCT design that is essential for the full specification of a design for the factory floor space includes the specification of the relative orientation of cells with respect to each. This could be explored as an optimization problem with the objective of minimizing the distance travelled by parts within cells and the distance travelled by machines during re-layout. This remains a topic for future research.

8.3 Forecast Information Required for FCT Design

The generation of a FCT that is relevant for an extended period of time requires information on the machines that are to be made available to the system and the relationship between parts and machines. The development of methods for the selection of machines was not conducted in this research. Researchers such as Taha and Rostam [74] have addressed the matter of appropriate machine selection to impart the necessary flexibility to the system to cope with current and future manufacturing requirements.

Although methods for selecting appropriate machines are available, the challenge remains of determining the types of parts that are to be used in future products and the demand for these products. In a business, Product Development or Research and Development departments can provide information on the parts required by future product generations, however, the less deterministic aspect is estimating the performance of future products in markets.

The Norton-Bass model is a model that can provide forecast information on how current and future products may perform in markets [95]. The original Bass Model was created by Bass [96] to forecast the performance of a new product in markets. The model is recognised as being one of the most frequently referenced, thoroughly researched and widely used marketing models in the world. The multi-generational forecasting model was created by Norton and Bass [97]. The Norton-Bass forecast for four generations of a single product type is given by equations 105-108:

$$S_1(t) = F(t)m_1[1 - F(t - \tau_2)] \tag{105}$$

$$S_2(t) = F(t - \tau_2)[m_2 + F(t)m_1][1 - F(t - \tau_3)], \tag{106}$$

$$S_3(t) = F(t - \tau_3)\big[m_3 + F(t - \tau_2)[m_2 + F(t)m_1]\big][1 - F(t - \tau_4)] \tag{107}$$

$$S_4(t) = F(t - \tau_4)[m_4 + F(t - \tau_3)\big[m_3 + F(t - \tau_2)[m_2 + F(t)m_1]\big]] \tag{108}$$

Here $S_i(t)$ is the total sales at time t, m_i is the incremental market potential and τ_i is the time to market of the i^{th} generation. The cumulative probability density of product adoption is given by:

$$F(\cdot) = \frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}} \tag{109}$$

Here p is the innovation coefficient and q is the imitation coefficient. A detailed presentation of the Norton-Bass model is beyond the scope of this thesis, however equations 105-108 provide the information required to forecast sales at given time. This can be used to determine the part mix and volume to be produced at a given time. Figure 8.1 shows an example Norton-Bass model forecast for two products with two generations.

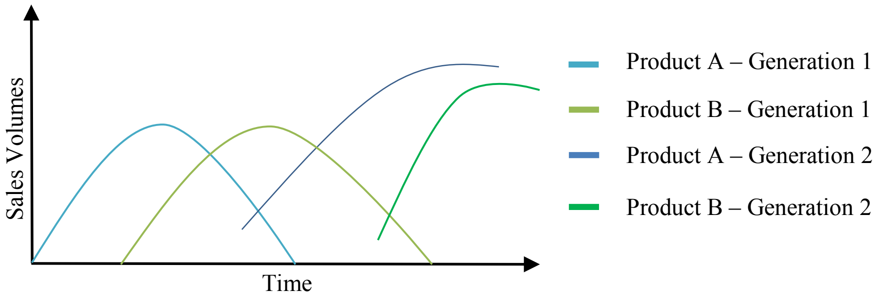

Figure 8.1: Norton-Bass Forecast for two Products in Two Generations (adapted from [95])

8.4 Group Analysis and FCT Design

An objective of this research was to research and develop an original cell formation method that supports the initial design of the system for reconfigurability. Conventional group analysis methods identify cells as clusters of machines. In this research cells were identified by creating clusters of sites for machines. These sites are occupied by machines when the need arises and are empty at other times. The identification and creation of these sites within cells is what imparts reconfigurability to the system.

In this research the Simultaneous Fuzzy Clustering heuristic was presented as means of simultaneously identifying sites for machines in cells and generating the corresponding part families. The advantage of performing these activities simultaneously is that the cell configurations and part family configurations can be simultaneously adjusted in order to better match each other. The SFC heuristic was able to successfully create clusters of sites for machines in cells, thereby fulfilling the objective of creating cells that support reconfigurability at the design stage.

In this research the FCMdd algorithm was used to generate an initial set of part family prototypes and a membership matrix as a seed to the SFC heuristic. The combination of the FCMdd algorithm and the SFC heuristic provided a fast and robust solution to clustering machine sites and part families. The FCMdd algorithm and the SFC heuristic required a maximum of six iterations each in order to generate a result for the problems in this research. Nonetheless there is room for improvement and shortcomings and areas requiring improvement will be discussed in the sections that follow.

8.4.1 Difficulty in Selecting Parameter *L*

The SFC heuristic is based on principles of fuzzy clustering where entities belong to clusters based on varying degrees of membership. In this heuristic, if the membership of a machine in a cell was $\mu_{ik} > L$ then this was considered sufficient for a site for machine i to be created in cell k. Parameter L represents the relative part load that a cell will impose on a machine type based on the manufacturing requirements of the corresponding part family. Lower values of L permit more sites to be added to a cell and if $L = 0$, sites for every machine type will be added to the cell.

A difficulty of using the heuristic is the selection of an appropriate value of L. In this research solutions for various values of L were generated and compared. If the SFC heuristic is researched further a goal would be to eliminate the need for multiple experiments with the parameter L in order to arrive at an acceptable solution. The parameter can be eliminated by research and development of alternative cluster formation rules.

8.4.2 Risk and Opportunity in Investing in Multiple Sites

In this research, solutions were generated for various values of parameter L and compared. Multiple solutions were compared with respect to the number of exceptional elements and voids that occur in the block diagonalised machine-component incidence matrix. This matrix was generated under the assumption that every part is active and every site is occupied by a machine.

In a RCMS every part will not be active and every site will not be filled with the corresponding machine at all times. A lower number of exceptional elements only indicates that a greater opportunity exists to reduce inter-cell flow by re-layout of machines. An increasing number of voids indicates that a greater risk exists that some cells may not be well utilised. In the RCMS study presented in Chapter 7, a solution for a very low value of L was accepted ($L=0.2$); consequently in the cost minimization assignment model, machine 7 was never assigned to the site that was available for it in Cell 1. This site was eventually eliminated by a review of the FCT.

In general a lower value of L reduces exceptional elements and increases the number of voids. These relationships are weak and the RCMS design study presented in Chapter 7 provided a case where a lower value of L produced a lower number of voids in a solution. The weak relationship between L and exceptional elements and voids presents further difficulty in selecting an appropriate value of L without multiple experiments.

Furthermore exceptional elements and voids only provide a loose indicator of the risks and opportunities that exist when investing in additional sites. The shortcoming is due to the fact that these indicators are based on the data available in a zero-one incidence matrix that does not include any additional production data such as product demand, etc. Improved indicators may be developed that include additional factors that influence site utilization.

8.4.3 Alternative Dissimilarity Measurers

In this research experiments were conducted with alternative types of dissimilarity measures. The Jaccard and Lance & Williams dissimilarity measures were shown to provide good performance when used with the Fuzzy C-Means membership model. In most instances the two measures provide identical solutions. Both the Jaccard and Lance & Williams dissimilarity measures are types of binary dissimilarity measures. An area in which further research may be conducted is the development of alternative dissimilarity measures for use with fuzzy membership models. These dissimilarity measures may include additional types of production data such as sequence of machine visitation, etc. Under special circumstances alternative dissimilarity measures may prove to be of some advantage.

8.5 Line Analysis in FCT Design

An objective of this research was to develop an original line analysis method that supports the initial design of a RCMS for reconfigurability. In this research the unidirectional flow optimization model was developed and solved with a Tabu Search algorithm. The model identifies how sites should be arranged within cells in the FCT to facilitate unidirectional flow within cells over multiple periods. This objective has been fulfilled.

The model was extended to identify duplicate sites for machines in cells that could be used to eliminate instances of part backtracking by the intra-cell re-layout of machines. The study of Chapter 7 showed that in a CMS with a fixed layout, the instances of part backtracking in cells was 66500. In the RCMS, with the same number of machines, the instances of part backtracking was 9000. This reduction in instances of part backtracking was facilitated by the re-layout of machines within cells. The intra-cell re-layout of machines within cells was possible due to the identification of duplicate sites for machines in cells during the FCT design.

With respect to the Tabu Search algorithm, this algorithm was selected for ease of implementation and its ability to provide good results quickly. The unidirectional flow optimization model could have been solved by a variety of different techniques including Genetic Algorithms and Simulated Annealing. The experimentation of the unidirectional flow optimization model with alternative algorithms can be conducted as future research to determine if any advantage can be gained.

In this research the case where bidirectional part flow is permitted in cells was not considered. Under bidirectional flow the arrangement of sites in cells should be generated with the objectives of minimizing the travel distance of parts in the cell. The investigation of site arrangement under bidirectional flow is a topic for future research.

8.6 Merits of the FCT concept

An objective of this research was to develop an original method of imparting reconfigurability to a CMS at the design stage. The Factory Configuration Template is a design of the factory floor space to support the future reconfiguration of cells. The advantages of having an initial system design for future reconfiguration include:

- the creation of machine movement lanes so that machines can be moved without interfering with other machines;
- the early identification and installation of equipment that can assist with reconfigurations, e.g. gantry cranes;

- safe space between machines can be maintained in a manner that is orderly and planned when reconfigurations occur;
- a safe working space for humans around machines can be maintained in a manner that is orderly and planned when reconfigurations occur;
- advanced installation of supplies of compressed air, water and electricity at sites to shorten system downtime;
- avoidance of irreversibility's such as the installation of immovable machines and other types of infrastructure that may block pathways for machine movement.

In general, any RCMS/DCMS method that generates reconfiguration plans without an initial system design for future reconfiguration can anticipate the following difficulties:

- the generation of reconfiguration plans that are impractical to implement;
- increased cost and difficulty in implementation;
- increased system downtime and loss of profitability.

8.7 Dedicated Sites on a FCT

8.7.1 Motivation for Dedicated Sites

A characteristic of the FCT is that sites are dedicated to individual machine types. The motivation for creating dedicated sites for machines is multifaceted, however, the various motivations are all based on reconfiguration plans being practically feasible. The first motivation for creating dedicated sites on a FCT is that individual machines have different footprints. A larger machine cannot be placed in a site that is tailored to a smaller machine.

The second motivation is that each machine may have specific requirements of compressed air, water, etc. Some machines may have specific requirements with respect to electrical connections, e.g. single-phase versus three-phase power systems. These types of connections are usually integrated into the physical structure of a factory and can be costly or inconvenient to modify. Moreover, modification of these connections can slow the reconfiguration process, increasing system downtime.

The third motivation is that dedicated sites for machines can prevent additional unnecessary machine relocations in order to create the best part flow characteristics in the system. In the study of section 7.10, without dedicated sites for machines, there was a total of 12 additional intra-cell machine relocations and the cost of re-layout increased by 20.6 %.

8.7.2 Increasing Space Requirements

The disadvantage of implementing dedicated sites for specific machine types on a FCT is that this increases the total number of sites in the system. An increasing number of sites increases the floor space required to implement a RCMS. In the RCMS study of Chapter 7, two cells were created with ten sites each. On average only 60 % of the sites in a cell were utilised at any time.

The advantage of the additional sites was that less machines would need to be relocated in order to obtain maximum unidirectional flow in cells (see section 7.10). In general RCMS/DCMS that allows any machine to occupy any site will require additional machine relocations in order to obtain the same flow characteristics as a RCMS that uses dedicated sites for specific machines.

A further advantage of a larger floor space with dedicated sites in a RCMS is that it imparts expansion flexibility to a system. Expansion flexibility is defined by Sethi and Sethi [98] as the ease in which the capacity or ability of a system can be increased. The additional sites available for machines will allow the capacity of the system to be scaled through the purchase of additional machines. The availability of pre-existing sites for machines will allow system capacity to be scaled quickly, as needed. Rapid scalability was identified by Koren et al [1, 9] as a key requirement of reconfigurable systems that are capable of responding quickly to an increase in market demand.

8.7.3 Sharing of Sites

The possibility of multiple machine types sharing sites is feasible provided that the following requirements are fulfilled:

1. the machines have the same space requirements or the sites are created to be large enough to accommodate a variety of machine sizes;
2. the machines are not required in the same site at the same time;
3. the machines have the same supply requirements (electricity, water, compressed air, etc.) or the manufacturer will willing to install new supplies accordingly.

Consider the example which will now be presented.

Example

Table 8.1: Reconfiguration Plan for Cell 1 Based on Shared Sites.

	Sequence of Machines							
Location	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Year 1	5	1	[ ]	3	4	6	[ ]	9
Year 2	5	1	[ ]	3	4	6	[ ]	9
Year 3	[ ]	1	2	3	4	12	[ ]	5
Year 4	[ ]	1	2	3	4	12	[ ]	5
Year 5	[ ]	[ ]	2	3	4	6	8	5
Year 6	[ ]	[ ]	2	3	4	6	8	5

With respect to the example of Chapter 7, the number of sites in Cell 1 may be reduced to eight as shown in Table 8.1. This is based on machine type 5 and 9 sharing the site at location 8 in the sequence of sites. Machine 6 and 12 also share the site at location 6 in the sequence of machines. See Table 7.27 for a comparison.

If the FCT specifies that two machine types may share a site then reconfiguration planning models require additional constraints to ensure that both machines are not allocated to the same site at the same time. The following constraint can be added to the capacity based assignment model and cost minimization model to ensure that machine 6 and 12 are not assigned to the cell at the same time:

$$x_{6,1,k} + x_{12,1,k} = 1, \qquad \forall k \in K \tag{110}$$

The addition of constraints of this type have the possibility to increase inter-cell part flow volumes. Consider the scenario where the part family corresponding to Cell 1 requires both machine 6 and 12 in the cell at the same time. Since machine 6 and 12 are not allowed in the cell at the same time, some parts will have to perform an inter-cell part movement.

In the case of machine 5 and 9, machine 5 has two sites that it may occupy. Although machine 5 and 9 are compatible for sharing sites, machine 9 is restricted to the site in the eighth location. A motivation for disallowing it at the first location is that this is guaranteed to introduce part backtracking. In this scenario machine 9 is given preferential access to location 8 and the placement of machine 5 at location 8 can only take place when machine 9 is absent. In order to enforce this variables must be fixed in the inter-cell machine assignment model as follows: $y_k = 0$, $p_k = 0$, $q_k = 0$ for $k \epsilon F$. Here F is the set of indices corresponding to years for which the machine is not allowed at the site, $F \subseteq K$. The scenario may arise where preventing machine 5 from occupying the site at location 8 may result in some intra-cell part backtracking flow.

In summary, allowing machines to share sites may reduce opportunities for eliminating inter-cell part flow and intra-cell part backtracking flow. This is due to the additional constraints and fixing of variables in reconfiguration planning models to ensure that two machines are not assigned to the same site at the same time.

Opportunities for machines to share sites is best identified after FCT creation and after reconfiguration planning has been performed. This is the manner in which shared sites were identified here and without the need for additional constraints in planning models. If FCT creation and reconfiguration planning are performed at the same time, the results of the reconfiguration planning models may then be used to amend the FCT before its implementation on the factory floor. Nonetheless, the argument remains that a reduction in the number of sites in a RCMS reduces opportunities to improve part flow and the ability of the system to respond to change.

8.8 Reconfiguration Planning Models

An objective of this research was to develop an original multi-period reconfiguration planning method for the re-layout of a RCMS within the provision for reconfigurability imparted to the system at the design stage. Accordingly, two inter-cell machine assignment models and one intra-cell machine assignment model was developed based on the FCT concept.

8.8.1 Parameter Estimation

Inter and intra cell reconfiguration planning models both require information on the product volumes that the RCMS will produce over multiple periods in order for the evolution of configurations to be optimised over many years. The scenario where a system configuration is optimal in one year yet excessively costly to reconfigure in the next year is not desirable. In companies operating on a make-to-order strategy, product volumes may be specified in contracts. In companies operating on the make-to-stock strategy, the Norton-Bass model discussed in section 8.3 may be able to provide the forecast information required.

With respect to the parts that will be used in future products, product development departments or research and development departments will have to provide the information required. The relationships between future parts and machines can be established by Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) software. CAM software such as VERICUT® [99] and the CAMWorks® Virtual Machine Module [100]can be used to establish process cycle times on machines by simulation.

Inter-cell part flow costs and intra-cell part backtracking costs are parameters that also require estimation. At the point of designing the system the costs associated with the purchase of material handling equipment may be integrated into the flow cost per part. Other factors that may be considered include the cost of hiring labour to perform the task, distances travelled, electricity consumption and in some cases, the cost of additional buffering. The cost of inter- and intra-cell machine relocation also require estimation. The cost of machine moving equipment, the hiring of labour, the use of specialist technicians and machine recalibration may be factored into the cost. In instances some preparation to the factory floor may be required in terms of providing a suitable foundation for precision machines, this cost may also be factored into inter- and intra-cell machine relocation costs.

8.8.2 Inter-Cell Machine Assignment Models

Two models for inter-cell machine assignment were developed in this research. The models assign machines to cells over multiple periods, thereby creating a reconfiguration plan. The two models were the capacity based assignment model and the cost minimization assignment model. The Matlab® *intlinprog* solver was used to solve the models and was able to obtain optimal solutions in under an hour.

A capacity based assignment model assigns machines to where the greatest need for machine capacity exists. The reconfiguration budget is a constraint in this model. If the reconfiguration budget is generous this model has the blind spot of possibly relocating a machine to a cell to gain a miniscule capacity advantage.

The cost based assignment model assigns machines to cells based on the optimal trade-off between reconfiguration cost and the cost of an hour of deficit capacity. This model assigns machines in a manner that determines the best economic trade-off between machine relocation and part relocations. This model has the blind spot of preventing a reconfiguration for a miniscule cost saving advantage.

In the study of Chapter 7 the best reconfiguration plan was determined by comparing the results of both models since each covers the other's deficiency. Criteria that may be used to determine which solution to accept was presented in section 7.5.4.

In the cost based assignment model, the cost of an hour deficit capacity was calculated based on an approximation of the number of parts that will have to perform an inter-cell moves. In the study of Chapter 7 it was shown that a 4.2 % error existed in the cost estimation by the objective function, in comparison to the calculated cost based on the specified part volumes. This error is acceptably small but may be reduced further by performing a weighted average estimation of processing times on machine i. Part volumes to be produced per period may be used to determine the weighted average.

In both reconfiguration planning models the factors that were considered were the cost of relocating machines and the cost of relocating parts. Another factor that may be of interest is the cost of purchasing additional machines. Additional machines can eliminate the need to relocate parts or machines between cells. The reconfiguration planning models may be extended further to include this possibility. This is a topic for future research.

8.8.3 Intra-Cell Machine Assignment Model

The intra-cell machine assignment model developed in this research was used to determine the location of a single machine in a cell when multiple sites were available for occupation by this machine. The Implicit Enumeration algorithm was used to solve the models and was able to obtain optimal solutions in under an hour.

The model was formulated to be consistent with the inter-cell machine assignment models. A cost was only charged to the objective function when a machine was relocated inside a cell and not when a machine was allocated to the cell by the inter-cell assignment models. The need for consistency with inter-cell assignment models lead to the model being a quadratic zero-one model.

In the RCMS problems for which the intra-cell machine assignment model was applied, a maximum of one duplicate site in series was allowed per cell. In problems where more than one duplicate site in series exists the complex problem arises as to how to accurately determine the α_k and β_k flow coefficients. In the scenario where multiple machines have sites in series the α_k and β_k flow coefficients become functions of the positions of other machines. The development of improved models that are able to solve this complex problem is a topic for future research.

8.9 The Advantages and Disadvantages of RCMS

An objective of this research was to investigate the advantages of RCMS over CMS with fixed layout. During the literature review it was determined that if any advantage is to be gained by reconfigurability then the initial system design must support future configuration changes. The lack of an initial system design for reconfigurability when planning reconfigurations is a shortcoming of other RCMS/DCMS methods.

In the study of Chapter 7 it was shown that a RCMS requires a larger floor space than a CMS with a fixed layout. This is due to multiple sites being available for machines in cells. In section 7.10 it was shown that the cost of reconfiguration increased with the use of an exact number of sites for machines. The allowance of additional sites for machines and a larger factory floor space is a significant factor in enabling hassle free reconfigurations. The need for a larger factory floor space and the lower utilization of the floor space is a disadvantage of RCMS.

The comparative study conducted in section 7.9 revealed that the instances of inter-cell part flow was 1.75 times higher in the CMS with a fixed layout as compared to the RCMS. The instances of part backtracking was also 7.38 times higher in fixed CMS cells. The fixed CMS will also have significantly more work-in-process and longer manufacturing lead times than the RCMS. This makes the fixed CMS a system that is more difficult to manage.

The study revealed that the drastic reduction in the instances of inter-cell part flow in RCMS caused the RCMS to be more cost effective to operate. Considering the cost of inter-cell part flow, intra-cell part backtracking and the cost of machine layout/re-layout, the RCMS solution cost was 33 % less than the cost of the fixed CMS. The cost of unidirectional flow, which is the preferred type of flow in cells, was not added to the solution cost.

Well known group technology performance measures such as group efficiency, group efficacy, weighted average efficiency, grouping measure, void count and exceptional element count were applied to the block diagonalised machine-component incidence matrices in each of the six periods. The RCMS generally outperformed the fixed CMS in all of these measures. In one instance the fixed CMS outperformed the RCMS, however, a closer examination of part volumes revealed that the RCMS solution was more appropriate.

In summary, a RCMS has advantages over a fixed CMS, however, these advantages are situation dependant. If the cost or complexity of relocating machines does not have an economical trade-off against eliminating inter-cell part flow or intra-cell part backtracking then the fixed CMS may be the better option.

8.10 Chapter Summary

This chapter reviewed the methods that were used for FCT creation and reconfiguration planning for RCMS. The tasks of FCT creation and reconfiguration planning are complex and the success of the system design and reconfiguration plans depend on a large quantity of information being known in advance. Reference was made to methods that could be used to generate the information required and the factors that should be taken under consideration when estimating part flow and reconfiguration costs. The generation of accurate information on which to base FCT designs and reconfiguration plans is possibly the most crucial matter in the success of RCMS.

The methods that have been developed for FCT design and reconfiguration planning have been demonstrated to be capable of providing acceptable solutions, nonetheless opportunities exists for further research, development and innovation. These opportunities were identified in the discussions of this chapter.

The chapter presented discussions on the merits of RCMS design and reconfiguration planning based on the FCT concept in comparison to RCMS/DCMS reconfiguration planning methods that do not consider an initial system design for reconfiguration. The chapter concluded with a discussion on the advantages and disadvantages of RCMS over traditional CMS with fixed layouts.

This page has been intentionally left blank.

9 Conclusion

The aim of this research was to review existing CMS and DCMS design and reconfiguration planning techniques and formulate a methodology for the design and subsequent evolution of the layout of a Reconfigurable Cellular Manufacturing System. This was to establish if an appropriately designed RCMS can respond to the introduction of new products and changing product demand in a manner that is economical and advantageous from a production management perspective. The ability of a manufacturing system to respond to change rapidly and cost effectively is essential for business survival in competitive markets. The study of Chapter 7 demonstrated that RCMS can respond to changing product demand, the introduction of new products and the phasing out of older products in a manner that is more economical than CMS. It was also shown that the RCMS has significantly less inter-cell work flow and intra-cell part backtracking; from a production management perspective this implies that the RCMS has the benefit of less complex part flow and reduced material handling requirements. The reduced inter-cell part movement also implies less work-in-process and shorter manufacturing lead times since parts will not have to wait to be processed by more than one cell.

An objective of this research was to establish the deficiencies in existing CMS and DCMS design and reconfiguration planning methods. In the literature review it was found that reconfiguration planning methods were developed by researchers, without an initial manufacturing system design for future re-layout and configuration changes to clusters of machines in cells. Without an initial system design for future configuration changes the plans developed by application of theoretical models may not be practically implementable.

An objective of this research was to develop an original method of imparting reconfigurability to a CMS at the design stage. In this research the concept of a Factory Configuration Template was proposed. The FCT is a design of the factory floor space for the future reconfiguration of the system. The FCT specifies the location of sites for specific machine types in cells and machines are moved in and out of sites in order to create the necessary capacity and manufacturing functions in cells when needed. When machines are no longer need by a cell or a greater need arises in another cell, the machines are redistributed accordingly.

An objective of this research was to develop an original cell formation method that supports the initial design of the system for reconfigurability. The Simultaneous Fuzzy Clustering Heuristic is an original innovation of this research; it is able to identify manufacturing cells and part families simultaneously. The SFC identified manufacturing cells as clusters of machine sites, these sites are implemented on the FCT to create reconfigurability. Conventional GT methods identify cells as clusters of machines.

An objective of this research was to develop an original line analysis method that supports the initial design of a RCMS for reconfigurability. A model was developed for creating unidirectional flow in manufacturing cells. This model was solved by a custom implementation of a Tabu Search algorithm. The model was able to successfully minimise instances of part backtracking by proposing an optimal arrangement of machine sites on the FCT. The model was later extended for identifying duplicate sites for machine in series; these duplicate sites create additional opportunities for reconfiguration according to the FCT design. It was demonstrated that a single machine, moved between two locations in a cell, could eliminate some volume of intra-cell part backtracking. The case where bidirectional flow in cells is permitted or encouraged was not considered in this research. Bidirectional flow is usually more complex and more costly and was therefore not given significant attention in this research. The consideration of bidirectional flow is a topic for future research.

An objective of this research was to develop an original multi-period reconfiguration planning method for the re-layout of a RCMS, within the provision for reconfiguration created in the system at the design stage. Two inter-cell machine assignment models and one intra-cell machine assignment model was developed. These models allow the creation of multi-period reconfiguration plans within the provisions for reconfiguration made by the FCT.

The two inter-cell machine assignment models were developed as integer linear programs that were solved by the Matlab® *intlinprog* solver. The first model was the capacity based assignment model that assigned machines based on minimising the surplus and deficit capacity in cells. The second model was the cost minimization model that assigned machines to cells based on an optimal cost trade-off between machine relocation and the relocation of parts between cells. Both models used in conjunction, provide insight into how machines should be assigned over multiple periods; since neither cost nor capacity alone can serve as the basis for reconfiguration planning. An opportunity for further research on the models is the integration of the possibility of additional machine purchases. Additional machine purchases can eliminate the need for the relocation of both parts and machines.

The two inter-cell machine assignment models that were developed, included the assumption that changes to the layout of machines does not necessitate changes to the layout of material handling equipment such as conveyors and robots. This assumption was justified based on an appropriately implemented FCT, where potential sites for machines were identified in advance and material handling equipment could be appropriately positioned during the installation of the RCMS. The case may exist where a manufacturer may wish to rearrange the layout of material handling equipment when the machine layout changes. The development of an inter-cell assignment model that is capable of considering the cost of re-layout of material handling equipment remains a topic for future research.

The intra-cell machine assignment model was developed as a zero-one quadratic program. The model was solved by the Implicit Enumeration algorithm; the algorithm was able to find optimal solutions quickly for problems solved in this research. The model determines the location of a single machine in a cell when two sites are available to the machine. The model assigns the machine to the available sites over multiple periods in order to negate instances of intra-cell part backtracking. A limitation of the model is that it is capable of assigning a single machine type at a time. The development of an intra-cell assignment model that is capable of simultaneously assigning multiple machine types to multiple sites in series remains a topic for future research.An objective of this research was to investigate the advantages of RCMS over CMS with fixed layouts, through the application of the design and reconfiguration planning methods that have been developed. The methods that were developed for FCT design and reconfiguration planning were demonstrated through a study in Chapter 7. The methods were used successfully to generate a FCT and generate reconfiguration plans for a six year period. The resulting RCMS was compared to a fixed CMS with the same product demand and the same number of machines. The RCMS was shown to provide a 33 % cost saving if implemented over the fixed CMS. The RCMS also had significantly less inter-cell flow and inter-cell part backtracking, therefore making it a less chaotic system that is easier to control. The RCMS is also expected to have less work-in-process and shorter manufacturing lead times due to the improved flow of work in the system.

In this research it was observed that the RCMS concept becomes feasible if an economic trade-off can be found between the cost of relocating parts and the cost of relocating machines. If the cost of machine relocation is not offset by the cost of relocating parts then a RCMS is not justified. It was also observed that a trade-off exists with respect to the size of the factory floor and the cost at which reconfiguration may be performed. A larger number of sites on the factory floor, dedicated to specific machines, allows the part flow in the manufacturing system to be improved while reducing machine movements. A further observation is that if machines are allowed to share sites, thereby reducing the number of sites, this results in reduced opportunities for eliminating inter-cell part flow and intra-cell part backtracking. In general, a RCMS can be expected to have a larger floor space than a CMS if the FCT concept is adopted.

The methods for RCMS design and reconfiguration planning developed in this research can be used by manufacturers wishing to implement a reconfigurable system. Nonetheless, these methods are only to serve as additional tools in determining the best type of manufacturing system to implement. The choice to implement a RCMS is dependent on the specific requirements of the manufacturing system, which includes the future outlook with respect to the introduction of new products and the anticipated product demand. The design of a RCMS and reconfiguration planning depend on a large amount of information pertaining to future products; this is difficult to determine with high accuracy at the outset. If it is difficult to generate such information or product variety is very high, a manufacturer may choose to emphasise flexibility rather than reconfigurability in the system design. Machines with high flexibility in cells can eliminate the need for inter-cell part flows and reconfiguration. Obtaining the best balance between flexibility and reconfigurability in a system is key to profitability, system responsiveness to change and simpler system control.

10 References

[1] Y. Koren and M. Shpitalni, "Design of Reconfigurable Manufacturing Systems," *Journal of Manufacturing Systems,* vol. 29, pp. 130-141, 2010.

[2] R. Tavakkoli-Moghaddam, M. B. Aryanezhad, N. Safaei, and A. Azaron, "Solving a dynamic cell formation problem using metaheuristics," *Applied Mathematics and Computation,* vol. 170, pp. 761-780, 2005.

[3] N. Hyer and U. Wemmerlov, *Reorganizing the Factory: Competing Through Cellular Manufacturing*, 1st ed. United States of America: Productivity Press, 2002.

[4] M. Aghajani, A. Keramati, and B. Javadi, "Determination of number of kanban in a cellular manufacturing system with considering rework process," *The International Journal of Advanced Manufacturing Technology,* vol. 63, pp. 1177-1189, 2012.

[5] M. Holweg, "The genealogy of lean production," *Journal of Operations Management,* vol. 25, pp. 420-437, 2007.

[6] S. A. Irani, S. Subramanian, and Y. S. Allam, "Introduction to Cellular Manufacturing Systems," in *Handbook of Cellular Manufacturing Systems*, 1st ed: John Wiley & Sons, Inc., 2007, pp. 1-23.

[7] C. R. Shiyas and V. Madhusudanan Pillai, "A mathematical programming model for manufacturing cell formation to develop multiple configurations," *Journal of Manufacturing Systems,* vol. 33, pp. 149-158, 2014.

[8] P. McLaughlin and I. Durazo-Cardenas, "Cellular Manufacturing Applications in MRO Operations," *Procedia CIRP,* vol. 11, pp. 254-259, 2013.

[9] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy*,* H.Van Brussel, "Reconfigurable Manufacturing Systems," *Annals of the CIRP,* vol. 48, pp. 527-540, 1999.

[10] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable Manufacturing Systems: Key to Future Manufacturing," *Journal of Intelligent Manufacturing,* vol. 11, pp. 403 - 419, 2000.

[11] A. Azab and B. Naderi, "Modelling the Problem of Production Scheduling for Reconfigurable Manufacturing Systems," *Procedia CIRP,* vol. 33, pp. 76-80, 2015.

[12] H. A. ElMaraghy, "Flexible and Reconfigurable Manufacturing Systems Paradigms," *International Journal of Flexible Manufacturing Systems,* vol. 17, pp. 261-276, 2006.

[13] M. Rheault, J. R. Drolet, and G. Abdulnour, "Dynamic cellular manufacturing system (DCMS)," *Computers & Industrial Engineering,* vol. 31, pp. 143-146, 1996.

[14] J. L. Burbidge, "Production Flow Analysis," *Production Engineer,* vol. 42, pp. 742 - 752, 1963.

[15] J. L. Burbidge, "Production flow analysis," *Production Engineer* vol. 60, pp. 139-152, 1971.

[16] J. L. Burbidge, "A manual of Production Flow Analysis," *Production Engineer* vol. 56, pp. 34-38, 1977.

[17] A. Mungwattana, "Design of Cellular Manufacturing Systems for Dynamic and Uncertain Production Requirements with Presence of Routing Flexibility." PhD Thesis, Virginia Polytechnic Institute and State University, 2000

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

[19] M. I. M. Wahab and M. F. S. Osman, "A fundamental flexibility measure: Machine flexibility," in *Service Systems and Service Management (ICSSSM), 2013 10th International Conference on*, 2013, pp. 70-74.

[20] B. R. Sarker, "Measures of grouping efficiency in cellular manufacturing systems," *European Journal of Operational Research,* vol. 130, pp. 588-611, 2001.

[21] J. Arkat, L. Hosseini, and M. H. Farahani, "Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm," *Expert Systems with Applications,* vol. 38, pp. 9597-9602, 2011.

[22] M. P. Chandrasekharan and R. Rajagopalan, "MODROC: an extension of rank order clustering for group technology," *International Journal of Production Research,* vol. 24, pp. 1221-1233, 1986.

[23] F. F. Boctor, "A linear formulation of the machine-part cell formation problem," *International Journal of Production Research,* vol. 29, pp. 343-356, 1991.

[24] M. Hassannezhad, M. Cantamessa, F. Montagna, and F. Mehmood, "Sensitivity Analysis of Dynamic Cell Formation Problem through Meta-heuristic," *Procedia Technology,* vol. 12, pp. 186-195, 2014.

[25] B. R. Sarker and Z. Li, "Measuring Matrix-Based Cell Formation with Alternative Routings," *The Journal of the Operational Research Society,* vol. 49, pp. 953-965, 1998.

[26] A.-P. Hameri, "Production flow analysis—Cases from manufacturing and service industry," *International Journal of Production Economics,* vol. 129, pp. 233-241, 2011.

[27] H. Opitz, W. Eversheim, and H. P. Wiendahl, "Workpiece classification and its industrial application," *International Journal of Machine Tool Design and Research,* vol. 9, pp. 39-50, 1969.

[28] J. McAuley, "Machine Grouping for Efficient Production," *The Production Engineer,* vol. 51, pp. 53 - 57 1972.

[29] K. Yasuda and Y. Yin, "A dissimilarity measure for solving the cell formation problem in cellular manufacturing," *Computers and Industrial Engineering,* vol. 39, pp. 1-17, 2001.

[30] T. Ghosh and P. K. Dan, "Effective Clustering Method for Group Technology Problems: A Short Communication," *e-Jornal of Science and technology,* vol. 6, pp. 23-28, 2011 2011.

[31] H. Seifoddini and M. Djassemi, "The Production Data-Based Similarity Coefficient Versus Jaccard's Similarity Coefficient," *Computers & Industrial Engineering,* vol. 21, pp. 263–266, 1991.

[32] T. Gupta and H. Seifoddini, "Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system," *International Journal of Production Research* vol. 28, pp. 1247-1269, 1990.

[33] J. C. Wei and G. M. Kern, "Commonality analysis: A linear cell clustering algorithm for group technology," *International Journal of Production Research,* vol. 27, pp. 2053-2062, 1989.

[34] W. S. Chow and O. Hawaleshka, "An efficient algorithm for solving the machine chaining problem in cellular manufacturing," *Comput. Ind. Eng.,* vol. 22, pp. 95-100, 1992.

[35] H. K. Seifoddini, "Single linkage versus average linkage clustering in machine cells formation applications," *Computers & Industrial Engineering,* vol. 16, pp. 419–426, 1989.

[36] J. R. King, "Machine-component group formation in group technology," *Omega,* vol. 8, pp. 193-199, 1980.

[37] J. R. King and V. Nakornchai, "Machine-component group formation in group technology: review and extension," *International Journal of Production Research,* vol. 20, pp. 117-133, 1982.

[38] W. T. McCormick, P. J. Schweitzer, and T. W. White, "Problem Decomposition and Data Reorganization by a Clustering Technique," *Operations Research,* vol. 20, pp. 933-1009, 1972.

[39] M. V. Bhat and A. Haupt, "An Efficient Clustering Algorithm," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. SMC-6, pp. 61-64, 1976.

[40] M. P. Chandrasekharan and R. Rajagopalan, "An ideal seed non-hierarchical clustering algorithm for cellular manufacturing," *International Journal of Production Research,* vol. 24, pp. 451-463, 1986.

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

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

[43] R. Rajagopalan and J. L. Batra, "Design of cellular production systems: A graph-theoretic Approach," *international Journal of Production Research,* vol. 13, pp. 567-579, 1975.

[44] H. Wang, B. Alidaee, F. Glover, and G. Kochenberger, "Solving group technology problems via clique partitioning," *International Journal of Flexible Manufacturing Systems,* vol. 18, pp. 77-97, 2006.

[45] G. Srinivasan, T. T. Narendran, and B. Mahadevan, "An Assignment model for the part families problem in group technology," *International Journal of Production Research,* vol. 28, , pp. 145-152, 1990.

[46] A. Kusiak, "The generalized group technology concept," *International Journal of Production Research,* vol. 25, pp. 561-569, 1987.

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

[48] M. Saidi-Mehrabad and V. R. Ghezavati, "Designing Cellular Manufacturing Systems under Uncertainty," *Journal of Uncertain Systems,* vol. 3, pp. 315-320, 1990 2009.

[49] G. Egilmez, G. A. Süer, and O. Ozgüner, "Capacitated Cellular Manufacturing System Design: A Genetic Algorithm Approach," presented at the 23rd Annual Production and Operations Management Conference, Chicago, Illinois, USA, 2012.

[50] A. T. Gomez, F. R. Ferrary, J. V. C. dos Santos, and L. Chiwiacowsky, "Cell formation problem: An multithreading Tabu search for setup time optimization for limited machine magazines: A New Solution For A Classical Problem," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, 2014, pp. 332-339.

[51] H. Nouri and T. S. Hong, "Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations," *Journal of Manufacturing Systems,* vol. 32, pp. 20-31, 2013.

[52] Y. Won and K. R. Currie, "Fuzzy ART/RRR-RSS: a two-phase neural network algorithm for part-machine grouping in cellular manufacturing," *International Journal of Production Research,* vol. 45, pp. 2073-2104, 2007.

[53] T. Ghosh, S. Sengupta, M. Chattopadhyay, and P. Dan, "Meta-heuristics in cellular manufacturing: A state-of-the-art review," *International Journal of Industrial Engineering Computations,* vol. 2, pp. 87-122, 2011.

[54] M. Chattopadhyay, S. Sengupta, T. Ghosh, P. K. Dan, and S. Mazumdar, "Neuro-genetic impact on cell formation methods of Cellular Manufacturing System design: A quantitative review and analysis," *Computers & Industrial Engineering,* vol. 64, pp. 256-272, 2013.

[55] S. C. Kumar and M. P. Chandrasekharan, "Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology," *International Journal of Production Research,* vol. 28, pp. 233-243, 1990.

[56] S. M. Ng, "Worst-case analysis of an algorithm for cellular manufacturing," *European Journal of Operational Research,* vol. 69, pp. 384-398, 1993.

[57] J. Miltenburg and W. Zhang, "A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology," *Journal of Operations Management,* vol. 10, pp. 44-72, 1991.

[58] N. Safaei, M. Saidi-Mehrabad, R. Tavakkoli-Moghaddam, and F. Sassani, "A fuzzy programming approach for a cell formation problem with dynamic and uncertain conditions," *Fuzzy Sets and Systems,* vol. 159, pp. 215-236, 2008.

[59] N. Safaei and R. Tavakkoli-Moghaddam, "Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems," *International Journal of Production Economics,* vol. 120, pp. 301-314, 2009.

[60] N. Safaei, M. Saidi-Mehrabad, and M. S. Jabal-Ameli, "A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system," *European Journal of Operational Research,* vol. 185, pp. 563-592, 2008.

[61] I. Mahdavi, A. Aalaei, M. M. Paydar, and M. Solimanpur, "Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment," *Computers & Mathematics with Applications,* vol. 60, pp. 1014-1025, 2010.

[62] H. Bayram and R. Şahin, "A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear Programming embedded hybrid solution techniques," *Computers & Industrial Engineering,* vol. 91, pp. 10-29, 2016.

[63] S. Sharifi, S. S. Chauhan, and N. Bhuiyan, "A dynamic programming approach to GA-based heuristic for multi-period CF problems," *Journal of Manufacturing Systems,* vol. 33, pp. 366-375, 2014.

[64] J. Miltenburg, "U-shaped production lines: A review of theory and practice," *International Journal of Production Economics,* vol. 70, pp. 201-214, 2001.

[65] K. K. Goyal, P. K. Jain, and M. Jain, "A comprehensive approach to operation sequence similarity based part family formation in the reconfigurable manufacturing system," *International Journal of Production Research,* vol. 51, pp. 1762-1776, 2013.

[66] M. P. Groover *Automation, Production Systems, and Computer-Integrated Manufacturing: International Version*, 3rd International ed. London, UK: Pearson Education, 2008.

[67] R. H. Hollier, "The Layout of Multi-Product Lines," *International Journal of Production Research,* vol. 2, pp. 47-57, 1963.

[68] Y.-C. Ho and T.-W. Liao, "A concurrent solution for intra-cell flow path layouts and I/O point locations of cells in a cellular manufacturing system," *Computers & Industrial Engineering,* vol. 60, pp. 614-634, 2011.

[69] H. Samarghandi and K. Eshghi, "An efficient tabu algorithm for the single row facility layout problem," *European Journal of Operational Research,* vol. 205, pp. 98-105, 2010.

[70] Y.-C. Ho and C. L. Moodie, "Machine layout with a linear single-row flow path in an automated manufacturing system," *Journal of Manufacturing Systems,* vol. 17, pp. 1-22, 1998.

[71] M. Solimanpur, P. Vrat, and R. Shankar, "An ant algorithm for the single row layout problem in flexible manufacturing systems," *Computers & Operations Research,* vol. 32, pp. 583-598, 2005.

[72] Y. S. Allam and S. A. Irani, "Systematic Redesign of a Manufacturing Cell," in *Handbook of Cellular Manufacturing Systems*, ed: John Wiley & Sons, Inc., 2007, pp. 661-679.

[73] R. Tavakkoli-Moghaddam, N. Safaei, and F. Sassani, "A new solution for a dynamic cell formation problem with alternative routing and machine costs using simulated annealing," *Journal of the Operational Research Society,* vol. 59, pp. 443-454, 2007.

[74] Z. Taha and S. Rostam, "A hybrid fuzzy AHP-PROMETHEE decision support system for machine tool selection in flexible manufacturing cell," *Journal of Intelligent Manufacturing,* vol. 23, pp. 2137-2149, 2012.

[75] D. Gyulaia, Z. Véna, A. Pfeiffera, J. Vánczaa, and L. Monostoria, "Matching Demand and System Structure in Reconfigurable Assembly Systems," in *45th CIRP Conference on Manufacturing Systems (CIRP CMS 2012)*, Athens, Greece, 2012, pp. 579 – 584.

[76] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics,* vol. 3, pp. 32-57, 1973.

[77] J. Bezdek, "Models for Pattern Recognition," in *Pattern Recognition with Fuzzy Objective Function Algorithms*, ed: Springer US, 1981, pp. 1-13.

[78] J. Macqueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[79] R. Krishnapuram, A. Joshi, and L. Yi, "A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering," in *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, 1999, pp. 1281-1286 vol.3.

[80] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[81] S. S. Choi, S. H. Cha, and C. Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal on Systemics, Cybernetics and Informatics,* vol. 8, pp. 43-48, // 2010.

[82] L. Yan and S. A. Irani, "Classroom Tutorial on the Design of a Cellular Manufacturing System," in *Handbook of Cellular Manufacturing Systems*, ed: John Wiley & Sons, Inc., 2007, pp. 613-659.

[83] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research,* vol. 13, pp. 533-549, 1986.

[84] F. Glover, "Tabu Search—Part I," *ORSA Journal on Computing,* vol. 1, pp. 190-206, 1989.

[85] F. Glover, "Tabu Search—Part II," *ORSA Journal on Computing,* vol. 2, pp. 4-32, 1990.

[86] A. H. Land and A. G. Doig, "An Automatic Method of Solving Discrete Programming Problems," *Econometrica,* vol. 28, pp. 497-520, 1960.

[87] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, "Gomory cuts revisited," *Operations Research Letters,* vol. 19, pp. 1-9, 1996.

[88] R. J. Vanderbei, *Linear Programming Foundations and Extensions*, 4 ed. New York: Springer US, 2013.

[89] R. E. Gomory, "An Algorithm for the Mixed Integer Problem," The RAND Corporation, Santa Monica, California1960.

[90] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh, "Lifted flow cover inequalities for mixed 0-1 integer programs," *Mathematical Programming,* vol. 85, pp. 439-467, 1999.

[91] A. N. Letchford and A. Lodi, "Strengthening Chvátal–Gomory cuts and Gomory fractional cuts," *Operations Research Letters,* vol. 30, pp. 74-82, 2002.

[92] A. Caprara and M. Fischetti, "Odd cut-sets, odd cycles, and 0-1/2 Chvátal–Gomory cuts," *Ricerca Operativa* vol. 26, pp. 51–80, 1996.

[93] A. Atamtürk , G. L. Nemhauser , and M. W. P. Savelsbergh, "Conflict graphs in solving integer programming problems," *European Journal of Operational Research,* vol. 121, pp. 40-55, 2000.

[94] V. Malhotra, T. Raj, and A. Arora, "Excellent Techniques of Manufacturing Systems: RMS and FMS," *International Journal of Engineering Science and Technology,* vol. 2, 2010.

[95] J. Padayachee and G. Bright, "Synthesis of Evolving Cells for Reconfigurable Manufacturing Systems," presented at the 27th International Conference on CADCAM, Robotics and the Factories of the Future, London England, 2014.

[96] F. Bass, "A New Product Growth for Model Consumer Durables," *Management Science,* vol. 15, pp. 215-227, 1969.

[97] J. A. Norton and F. M. Bass, "A Diffusion Theory Model of Adoption and Substitution for Successive Generations of High-Technology Products," *Management Science,,* vol. 33, pp. 1069-1086, 1987.

[98] A. K. Sethi and S. P. Sethi, "Flexibility in Manufacturing: A Survey," *The International Journal of Flexible Manufacturing Systems,* vol. 2, pp. 289 – 328, 1990.

[99] www.cgtech.com. (2015, 31 December). *VERICUT Module: CNC Machine Simulation*. Available: http://www.cgtech.com/products/about-vericut/machine-simulation/

[100] http://www.camworks.com/. (2015, 31 December). *Virtual Machine (VM)| CAMWorks Virtual Machine Module*. Available: http://www.camworks.com/modules/camworks-virtual-machine/

Appendix A

Note: machines are represented in rows and parts are represented in columns in the matrices provided in this appendix.

A.1 Random Generated Machine-Component Incidence Matrices

Table A - 1: Generated Machine-Component Incidence Matrix - Matrix 1

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
1	1	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0
3	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0
8	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0
10	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0
11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	0
13	0	0	0	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0
16	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
17	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

Table A - 2: Machine-Component Matrix With Block Diagonals

	3	12	13	14	16	17	18	19	21	26	1	8	20	22	23	24	25	27	28	29	30	2	4	5	6	7	9	10	11	15
2	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

There are 7 exceptional elements (1's) that are outside of the block diagonal structures. There are 163 voids (0's) within the block diagonal structures.

A.2 Machine-Component Incidence Matrices

Table A - 3: McAuley - 4x6[28]

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	0	0	1	0	0	1

Table A - 4: Chandrasekharan and Rajagopalan - 8x20 [22]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	1	1	0	0	0	0	1	1	0	0	0	1	1	1	0	1	1	0
2	0	1	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1
3	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	1	0	1	1
4	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	0	1	1	1	1
5	1	1	1	0	0	1	0	0	0	1	0	1	1	0	1	1	1	0	1	1
6	1	1	0	0	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1	0
7	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	1	0	1	1
8	1	1	1	1	1	0	0	1	1	1	0	0	1	1	0	1	1	0	0	0

Table A - 5: Boctor -Test Problem 1 - 16x30 [23]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0
2	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
5	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1
7	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1
10	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1
12	1	0	1	1	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	0	1	0
14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
15	1	0	1	1	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1

Table A - 6: King - 16x43 [36]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1
6	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1	1
7	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1
9	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
10	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
16	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0

Table A - 7: King and Nakornchai 16x43 [37]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1
6	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1	1
7	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1
9	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
10	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
16	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0

This page was intentionally left blank.

Appendix B

B.1 Sample Calculation - Binary Dissimilarity

Consider Table B - 1, which is an extract from Table 4.1 in section 4.4.1.

Table B - 1: Machine-Component Incidence Matrix for Two Parts

		PARTS	
		1	**2**
MACHINES	**1**	1	0
	2	1	1
	3	0	1
	4	0	0
	5	0	0
	6	0	0
	7	0	0

In this example part 1 and part 2 both make use of machine 2, therefore $a=2$. Part 1 makes exclusive use of machine 1 therefore $b = 1$. Part 2 makes exclusive use of machine 3, therefore $c = 1$. Both parts do not make use of machines 4, 5, 6, and 7 therefore $d= 4$. The Lance & Williams dissimilarity for parts 1 and 2 is calculated as follows:

$$D_{LANCE\&WILLIAMS} = \frac{b+c}{(2a+b+c)} = \frac{1+1}{(2\times1+1+1)} = 0.5$$

B.2 Example – Rule for Allocation of Parts to Families

The machine-component incidence matrix presented in Table 4.1 and the part-part dissimilarity matrix presented in Table 4.2 will be used in this illustration. The cluster prototypes (medoids) and membership matrix presented here were generated by the FCMdd algorithm, using the part-part dissimilarity matrix with three clusters (i.e. $C=3$):

$$V = \{z_2, z_3, z_5\}$$

$$U_{Parts} = \begin{bmatrix} 0.43 & 1.00 & 0.00 & 0.21 & 0.00 & 1.00 & 0.14 & 0.20 & 0.60 & 0.00 & 0.14 \\ 0.36 & 0.00 & 1.00 & 0.36 & 0.00 & 0.00 & 0.71 & 0.20 & 0.20 & 0.00 & 0.71 \\ 0.21 & 0.00 & 0.00 & 0.43 & 1.00 & 0.00 & 0.14 & 0.60 & 0.20 & 1.00 & 0.14 \end{bmatrix}$$

The resultant medoids for the problem were part 2 (z_2) for cluster 1, part 3 (z_3) for cluster 2 and part 5 (z_5) for cluster 3. Each column of the membership matrix represents a part; in this instance parts 1-11. If parts are allocated to families according to maximum membership, the part families would be: *PF1 = {1, 2, 6, 9}*, *PF2 = {3, 7, 11}*, *PF 3 = {4, 5, 8, 10}*.

B.3 Example - Affinity Analysis for Machine Membership Matrix

An example is presented here to demonstrate the calculation performed in the *Generate_U_Machines()* function. Consider the group analysis problem of Table 4.1 in the part families generated in Appendix B.2. If the part families are: *PF1 = {1, 2, 6, 9}*, *PF2 = {3, 7, 11}*, *PF3 = {4, 5, 8, 10}*. The affinity rating a_{ik} for each machine k with respect to each part family i is:

$$A = \begin{bmatrix} 1 & 3 & 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 3 & 0 & 1 & 3 \end{bmatrix}$$

The values of matrix A were populated by counting the number of parts in each family that utilise each machine. For example a_{12} pertains to the number of parts in *PF1* that utilise machine 2. According to the machine-component incidence matrix (Table 4.1) parts 1, 2 and 6 use machine 1 therefore $a_{12} = 3$.

The *U_Machines* membership matrix calculated by equation 39 is:

$$U\_Machines = \begin{bmatrix} 0.25 & 1.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.75 & 0.00 & 0.00 & 0.00 & 1.00 & 0.67 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.33 & 1.00 \end{bmatrix}$$

A sample calculation for element μ_{11} is as follows:

$$\mu_{11} = \frac{a_{ik}}{\sum_{j=1}^{C} a_{jk}} = \frac{a_{11}}{a_{11}+a_{21}+a_{31}} = \frac{1}{1+3+0} = 0.25$$

B.4 Example – Rule for Allocation of Machines to Sites

The *ClusterSites()* function uses the *U_Machines* matrix to allocate sites to machines in cells based on the rule $\tilde{\mu}_{ik} = 1$ if $\mu_{ik} \geq L$ and $\tilde{\mu}_{ik} = 0$ otherwise. Consider the matrix:

$$U\_Machines = \begin{bmatrix} 0.25 & 1.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.75 & 0.00 & 0.00 & 0.00 & 1.00 & 0.67 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.33 & 1.00 \end{bmatrix}$$

Based on this matrix and *L= 0.25* (effectively 25 % part load), the sites for machines in cells would be as follows: *CS1 = {S1, S2, S3}*, *CS2 = {S1, S5, S6}*, *CS3 = {S4, S6, S7}*. Here *CS 1* denotes the cluster of sites allocated to Cell 1, etc. The prefix '*S*' denotes a site allocated to a cell for a corresponding machine number. For example, machine 1 has been allocated sites in Cell 1 and Cell 2.

B.5 Example – Generation of New Prototypes and Membership Matrix

An example of the working of the *PrototypeGeneration()* function will be demonstrated here and a sample calculation corresponding to the *Generate_U_Parts()* function will be presented. The group analysis problem of Table 4.1 (section 4.4.1) is used in this example.

Consider the set of machine sites: *CS1 = {S1, S2, S3}*, *CS2 = {S1, S5, S6}*, *CS3 = {S4, S6, S7}*, generated in Appendix B.4. The representation of the sites in cells as zero-one vectors gives rise to a new set of part family prototypes.

Table B - 2: Updated Primary prototypes

		PART FAMILY MEDOIDS		
		v_1	v_2	v_3
MACHINES SITES	**1**	1	1	0
	2	1	0	0
	3	1	0	0
	4	0	0	1
	5	0	1	0
	6	0	1	1
	7	0	0	1

The updated prototypes presented here are different to the medoids generated by the FCMdd algorithm. The original primary prototypes were the data vectors corresponding to parts 2, 3, and 5 in Table 4.1 .The updated membership matrix is now:

$$U\_Parts = \begin{bmatrix} 0.65 & 0.71 & 0.00 & 0.13 & 0.14 & 0.71 & 0.22 & 0.25 & 0.50 & 0.14 & 0.20 \\ 0.22 & 0.14 & 1.00 & 0.22 & 0.14 & 0.14 & 0.65 & 0.25 & 0.25 & 0.14 & 0.60 \\ 0.13 & 0.14 & 0.00 & 0.65 & 0.71 & 0.14 & 0.13 & 0.50 & 0.25 & 0.71 & 0.20 \end{bmatrix}$$

For illustrative purposes a sample calculation for μ_{11} will be provided. Table B - 3 is an extract from Table 4.1, showing the machines required by part 1.

Table B - 3: Machine Incidence for Part 1

		Part 1
MACHINES	**1**	1
	2	1
	3	0
	4	0
	5	0
	6	0
	7	0

The dissimilarity between part 1 and the updated prototypes will now be determined using the Lance & Williams binary dissimilarity measure.

Dissimilarity between v_1 and part 1:

$$D_{11} = \frac{b+c}{(2a+b+c)} = \frac{1+0}{(2\times2+1+0)} = 0.2$$

Dissimilarity between v_2 and part 1:

$$D_{21} = \frac{b+c}{(2a+b+c)} = \frac{2+1}{(2\times1+2+1)} = 0.6 :$$

Dissimilarity between v_3 and part 1

$$D_{21} = \frac{b+c}{(2a+b+c)} = \frac{3+2}{(2\times0+3+2)} = 1.0 :$$

The membership of part 1 with respect to cluster 1 is therefore:

$$\mu_{ik} = \frac{\left(\frac{1}{D_{ik}}\right)^{1/(m-1)}}{\sum_{j=1}^{C}\left(\frac{1}{D_{ik}}\right)^{1/(m-1)}} = \frac{\left(\frac{1}{0.2}\right)^{1/(1-1)}}{\left(\frac{1}{0.2}\right)^{1/(1-1)} + \left(\frac{1}{0.6}\right)^{1/(1-1)} + \left(\frac{1}{1.0}\right)^{1/(1-1)}} = 0.65$$

Appendix C

C.1 Example – Calculation of Processing Hours Required on a Machine

Consider the data in Table C - 1, which contains the part demand for year 1 and cycle times for parts on machine 2. Table C - 1 has been constructed from Table 5.1 and Table 5.2.

Table C - 1: Part Demand and Process Cycle Times on Machine 2 in Year 1 (P=Parts)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Demand (thousands of units)	6	6	4	3	0	0	0	2	9	9
Time (min)	30	25	26	-	-	27	28	28	-	-

The total number of operating hours required on machine type 2 is calculated by:

$$H_{ik} = \frac{\sum_{p=1}^{P} volume_{pk} \times minutes_{ip}}{60}$$

$$H_{21} = \frac{(6\times30+6\times25+4\times26+0\times27+0\times28+2\times28)\times10^3}{60} = 8167\ hours$$

C.2 Example – Calculation of Machines Required by System

If $H_{12} = 8167\ hours$, $W_1 = 5600\ hours$ and machine availability is 70 %; the number of machines of type 2 required in year 1 is calculated as follows:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{8167}{5600\times0.7} = 2.1$$

Machine numbers must be rounded up to ensure sufficient capacity, therefore a total of three machines of type 2 are required in year 1.

C.3 Example – Calculation of Processing Hours Per Machine Per Cell

Consider the data Table C - 2, which contains the part demand for Cell 1 in year 1 and cycle times for parts on machine 2. Table C - 2 has been constructed from Table 5.1 and Table 5.2.

Table C - 2: Part Demand and Process Cycle Times on Machine 2/Cell 1/Year 1

	P1	P2	P3	P4	P5
Demand (thousands of units)	6	6	4	3	0
Time (min)	30	25	26	-	-

The total number of operating hours required on machine type 2 in Cell 1 in year 1 is calculated by the following equation where *j* is the cell/part family number:

$$H_{ijk} = \frac{\sum_{p\in Part\ Family\ j} volume_{pk} \times minutes_{ip}}{60}$$

$$H_{211} = \frac{(6\times30+6\times25+4\times26)\times10^3}{60} = 7233\ hours$$

C.4 Example – Calculation of Machines Required by Cell

If H_{211} = 7233 *hours*, W_1 = 5600 *hours* and machine availability is 70 %; the number of machines of type 2 required by Cell 1 in year 1 is calculated as follows:

$$N_{ijk} = \frac{H_{ijk}}{W_k\times Availability} = \frac{7233}{5600\times0.7} = 1.8$$

Two machines of type 2 are required by Cell 1 in year 1.

Appendix D

D.1 Unidirectional Flow Optimization Model With Duplicate Machines

The updated unidirectional flow optimization model with machine sites in series is as follows:

$$min \sum_{i=1}^{M} \sum_{j \neq i}^{M} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{\alpha-1} x_{i\alpha} x_{j\beta} U_{i\alpha j\beta}$$

Subject to:

$$x_{i\alpha} = x_{j\beta}\,, \qquad \forall (i, \alpha) = (j, \beta)$$

$$\sum_{i=1}^{M} x_{i\alpha} = 1, \qquad \forall \alpha = 1, \dots, N$$

$$\sum_{\alpha=1}^{N} x_{i\alpha} \leq \zeta, \qquad \forall i = 1, \dots, M$$

$$x_{i\alpha} \in \{0,1\}\,;\; x_{j\beta} \in \{0,1\}$$

D.2 Sample PML Assignment Problem - Possible Solutions

Table D - 1: Space of All Solutions for Process-Machine-Location Mappings

#	Processes 1 – 7							#	Processes 1 – 7							#	Processes 1 – 7						
	1	2	3	4	5	6	7		1	2	3	4	5	6	7		1	2	3	4	5	6	7
1	2	4	3	5	3	4	1	33	7	4	3	5	3	4	1	65	10	4	3	5	3	4	1
2	2	4	3	5	3	4	9	34	7	4	3	5	3	4	9	66	10	4	3	5	3	4	9
3	2	4	3	5	3	8	1	35	7	4	3	5	3	8	1	67	10	4	3	5	3	8	1
4	2	4	3	5	3	8	9	36	7	4	3	5	3	8	9	68	10	4	3	5	3	8	9
5	2	4	3	5	6	4	1	37	7	4	3	5	6	4	1	69	10	4	3	5	6	4	1
6	2	4	3	5	6	4	9	38	7	4	3	5	6	4	9	70	10	4	3	5	6	4	9
7	2	4	3	5	6	8	1	39	7	4	3	5	6	8	1	71	10	4	3	5	6	8	1
8	2	4	3	5	6	8	9	40	7	4	3	5	6	8	9	72	10	4	3	5	6	8	9
9	2	4	6	5	3	4	1	41	7	4	6	5	3	4	1	73	10	4	6	5	3	4	1
10	2	4	6	5	3	4	9	42	7	4	6	5	3	4	9	74	10	4	6	5	3	4	9
11	2	4	6	5	3	8	1	43	7	4	6	5	3	8	1	75	10	4	6	5	3	8	1
12	2	4	6	5	3	8	9	44	7	4	6	5	3	8	9	76	10	4	6	5	3	8	9
13	2	4	6	5	6	4	1	45	7	4	6	5	6	4	1	77	10	4	6	5	6	4	1
14	2	4	6	5	6	4	9	46	7	4	6	5	6	4	9	78	10	4	6	5	6	4	9
15	2	4	6	5	6	8	1	47	7	4	6	5	6	8	1	79	10	4	6	5	6	8	1
16	2	4	6	5	6	8	9	48	7	4	6	5	6	8	9	80	10	4	6	5	6	8	9
17	2	8	3	5	3	4	1	49	7	8	3	5	3	4	1	81	10	8	3	5	3	4	1
18	2	8	3	5	3	4	9	50	7	8	3	5	3	4	9	82	10	8	3	5	3	4	9
19	2	8	3	5	3	8	1	51	7	8	3	5	3	8	1	83	10	8	3	5	3	8	1
20	2	8	3	5	3	8	9	52	7	8	3	5	3	8	9	84	10	8	3	5	3	8	9
21	2	8	3	5	6	4	1	53	7	8	3	5	6	4	1	85	10	8	3	5	6	4	1
22	2	8	3	5	6	4	9	54	7	8	3	5	6	4	9	86	10	8	3	5	6	4	9
23	2	8	3	5	6	8	1	55	7	8	3	5	6	8	1	87	10	8	3	5	6	8	1
24	2	8	3	5	6	8	9	56	7	8	3	5	6	8	9	88	10	8	3	5	6	8	9
25	2	8	6	5	3	4	1	57	7	8	6	5	3	4	1	89	10	8	6	5	3	4	1
26	2	8	6	5	3	4	9	58	7	8	6	5	3	4	9	90	10	8	6	5	3	4	9
27	2	8	6	5	3	8	1	59	7	8	6	5	3	8	1	91	10	8	6	5	3	8	1
28	2	8	6	5	3	8	9	60	7	8	6	5	3	8	9	92	10	8	6	5	3	8	9
29	2	8	6	5	6	4	1	61	7	8	6	5	6	4	1	93	10	8	6	5	6	4	1
30	2	8	6	5	6	4	9	62	7	8	6	5	6	4	9	94	10	8	6	5	6	4	9
31	2	8	6	5	6	8	1	63	7	8	6	5	6	8	1	95	10	8	6	5	6	8	1
32	2	8	6	5	6	8	9	64	7	8	6	5	6	8	9	96	10	8	6	5	6	8	9

Table D - 1 lists the locations corresponding to machine types, on which processes 1-7 may be performed for the problem of section 5.4.3.

Table D - 2: Performance of Solutions*

#	UM	BM	#	UM	BM	#	UM	BM
1	5	3	**33**	5	4	**65**	5	4
2	5	2	**34**	5	3	**66**	5	3
3	6	3	**35**	6	4	**67**	6	4
4	6	2	**36**	6	3	**68**	6	3
5	6	3	**37**	6	4	**69**	6	4
6	6	2	**38**	6	3	**70**	6	3
7	7	2	**39**	7	3	**71**	7	3
8	7	1	**40**	7	2	**72**	7	2
9	6	3	**41**	6	4	**73**	6	4
10	6	2	**42**	6	3	**74**	6	3
11	7	3	**43**	7	4	**75**	7	4
12	7	2	**44**	7	3	**76**	7	3
13	5	3	**45**	5	4	**77**	5	4
14	5	2	**46**	5	3	**78**	5	3
15	6	2	**47**	6	3	**79**	6	3
16	6	1	**48**	6	2	**80**	6	2
17	6	3	**49**	6	3	**81**	6	4
18	6	2	**50**	6	2	**82**	6	3
19	5	3	**51**	5	3	**83**	5	4
20	5	2	**52**	5	2	**84**	5	3
21	7	3	**53**	7	3	**85**	7	4
22	7	2	**54**	7	2	**86**	7	3
23	6	2	**55**	6	2	**87**	6	3
24	6	1	**56**	6	1	**88**	6	2
25	7	4	**57**	7	4	**89**	7	5
26	7	3	**58**	7	3	**90**	7	4
27	6	4	**59**	6	4	**91**	6	5
28	6	3	**60**	6	3	**92**	6	4
29	6	4	**61**	6	4	**93**	6	5
30	6	3	**62**	6	3	**94**	6	4
31	5	3	**63**	5	3	**95**	5	4
32	5	2	**64**	5	2	**96**	5	3

*UM = number of utilized machines, BM = number of backtracking movements, DT = distance travelled.

The number of utilized machines is determined by counting the number of unique elements in a solution. The set *A* corresponding to the first solution in Table D - 1 is $A = \{2, 4, 3, 5, 3, 4, 1\}$. Although there are seven elements in the set, only five elements are unique. The number of machines used is therefore five. Under this solution the part revisits the machines at locations $\alpha = 3$ and $\alpha = 4$ according to the ordered set *S'*. The number of backtracking movements is determined by counting the pairs of element in *A* that do not represent an ascending order.

For example, the pairs $\{4,3\}$,$\{5,3\}$ and $\{4,1\}$ are not ascending order pairs and the number of backtracking for this solution is three.

The solution with the highest machines used and lowest instances of part backtracking is solution 8, with seven machines used and one instance of part backtracking. The solution is $A = \{2,4,3,5,6,8,9\}$. Part p required processes 1-7 on machines as follows: $P_p = \{2,4,3,5,3,4,1\}$. The output of the process to machine mapping function g is therefore $Q_p = \{(2,2),(4,4),(3,3),(5,5),(3,6),(4,8),(1,9)\}$. The first element is $(i,\alpha)=(2,2)$ implying that the first process is performed on machine 2 which is in location 2, etc. Notice that the third process is performed on machine type $i=3$ in location $\alpha = 3$ and the fifth process is performed on machine type $i= 3$ in location $\alpha=6$. The fifth process could have been performed on machine type $i=3$ in location $\alpha = 3$, however this would unnecessarily inflate the instances of part backtracking for the sequence of machine locations as defined by S'.

Appendix E

E.1 Calculation – Excess Capacity (Hours)

For the example problem of section 6.3.4, the hours of excess capacity made available by machine purchases is listed in Table E - 1. The total hours of excess capacity for mobile machines 1, 3, 8, 12 and 15 is 41007 hours. The breakdown of total excess hours per period is shown in the table.

Table E - 1: Hours of Excess Capacity Available

	E_{ik} (hours)		
Machine (*i*)	***k*= 1**	***k*= 2**	***k*= 3**
1	665	5210	6585
2	816	1089	1284
3	3806	1741	3835
4	425	420	183
5	382	382	382
6	392	1270	969
7	194	489	1103
8	1638	345	3189
9	779	195	584
10	747	747	747
11	878	324	1064
12	4061	4624	1638
13	1061	705	769
14	1028	818	36
15	1166	1406	1098
Total Mobile Machines	11336	13326	16345
Total Fixed Machines	6702	6439	7121
Total	18038	19765	23466

As a sample calculation consider the excess capacity with respect to machine type 1 in period 1. According to Table 6.1, Cell 1 requires 0 hours, Cell 2 requires 6735 hours and Cell 3 requires 5560 hours. According to Table 6.3, three copies of machine type 1 are to be made available in period *k=1*. Period 1 is 180 days and each working day is 24 hours, the total working hours per period is therefore $W_K = 4320\ hours$. The total hours of excess capacity is calculated as follows:

$$E_{ik} = W_k N_{ik} - \sum_{j\in J} H_{ijk} = 4320 \times 3 - (0 + 6735 + 5560) = 665\ hours$$

The total hours of excess capacity for mobile machines 1, 3, 8, 12 and 15 is calculated as follows:

$$11336 + 13326 + 16345 = 41007\ hours$$

This page was intentionally left blank.

Appendix F

F.1 Machine-Component Incidence Matrix

Table F- 1 presents the machine-component incidence matrix for the RCMS study of Chapter 7. Note: machines are represented in rows and parts are represented in columns in the matrix provided.

Table F- 1: Machine-Component Incidence Matrix

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	1	0	1	1	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0
2	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	1	0	1
3	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	0
4	0	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	0
5	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	1	1
7	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0
8	1	0	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1
9	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
10	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	0	0	0	0	1
11	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1
12	1	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	1	0	0

F.2 Part Quantities to be Produced per Annum

Table F- 2 presents the quantities in which parts 1-20 must be produced in years 1-6 in order to match the demand for Product A, B and C.

Table F- 2: Part Quantities* to be Produced per Year (Y=Year, P=Part)

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**
Y1	4	4	0	8	0	4	4	0	0	0	4	0	0	8	0	4	0	0	0	0
Y2	3.5	3.5	6	7	0	3.5	3.5	0	6	3	3.5	0	3	7	0	3.5	0	3	3	3
Y3	2	2	9	4	0	2	2	0	9	4.5	2	0	4.5	4	0	2	0	4.5	4.5	4.5
Y4	0	0	6.4	0	1.5	0	0	3	6.4	4.7	0	1.5	3.2	0	1.5	0	3	4.7	3.2	3.2
Y5	0	0	3	0	3	0	0	6	3	4.5	0	3	1.5	0	3	0	6	4.5	1.5	1.5
Y6	0	0	0	0	5	0	0	10	0	5	0	5	0	0	5	0	10	5	0	0

*Part quantities are represented in thousands.

As a sample calculation consider part 18. According to Table 7.2, one unit of part 18 is required for each unit of Product B and one unit of part 18 is required for each unit of Product C. The individual demands of Product B and Product C are shown in Table 7.1. The annual demand for part 18 based on the demand for these products is calculates as follows:

Year 1:

$$1 \times 0 + 1 \times 0 = 0\ parts$$

Year 2:

$$1 \times 3000 + 1 \times 0 = 3000\ parts$$

Year 3:

$$1 \times 4500 + 1 \times 0 = 4500\ \ parts$$

Year 4:

$$1 \times 3200 + 1 \times 1500 = 4700\ parts$$

Year 5:

$$1 \times 1500 + 1 \times 3000 = 4500\ parts$$

Year 6:

$$1 \times 0 + 1 \times 5000 = 5000\ parts$$

F.3 Hours of Processing Time Required on Machines

Table F- 3 shows the hours of processing time requires on machines 1-12 for years 1-6.

Table F- 3: Hours of Processing Time Required on Machines (M= Machine, Y = Year)

	Y1	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**
M1	2600	3625	3325	2215	2225	2584
M2	2600	4075	4000	3045	3150	0
M3	667	1934	2359	2115	2025	0
M4	1934	3442	3592	2342	1825	0
M5	1467	2484	2534	1930	1900	0
M6	2467	3609	3409	2072	1775	0
M7	1467	2334	2309	1870	2025	2500
M8	2867	4359	4209	2874	2725	0
M9	2134	3567	3617	2239	1700	1417
M10	2534	2667	1942	955	1175	1584
M11	1200	2000	2025	1464	1375	1500
M12	2934	4167	3867	2832	3050	3000

As a sample calculation consider machine 1 in the first year. This machine is required by parts 2, 4, 5, 6, 9, 13, 15, 16 and 17. According to

Table 7.3 parts 2,4, and 5 have a process cycle time of seven minutes, parts 15 and 17 have a process cycle time of eight minutes and parts 6, 9, 13 and 16 have a process cycle time of nine minutes. Individual part volumes for the calculation were read from Table F- 2. The total hours of processing time required in machine type 1 is calculated by equation 40 as follows:

$$H_{ik} = \frac{\sum_{p=1}^{P} volume_{pk} \times minutes_{ip}}{60}$$

$$H_{ik} = \frac{(4\times7+8\times7+0\times7+4\times9+0\times9+0\times9+0\times8+4\times9+0\times8)\times10^3}{60}$$

$$H_{ik} = 2600\ hours$$

F.4 Sample Calculation - Mandatory Machines Required Per Annum

Consider the following sample calculations for the number of machine type 1 required in years 1-6. Table F- 4 summarises the information required to determine the mandatory number of machine type 1 that is required by the RCMS in order to ensure sufficient production capacity. The working hours and the machine availability were defined in the problem of section 7.2.

Table F- 4: Information Required for the Machine Capacity Calculation

	Total Hours Required	Working Hours Available	Machine Availability
Y1	2600	7200	80 %
Y2	3626	7200	80 %
Y3	3325	7200	80 %
Y4	2215	7200	80 %
Y5	2225	7200	80 %
Y6	2584	7200	80 %

The mandatory annual numbers of machine type 1 are calculated by equation 42 as follows:

Year 1:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{2600}{0.8\times7200} = 0.45\text{ , say 1 machine}$$

Year 2:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{3626}{0.8\times7200} = 0.63\text{ , say 1 machine}$$

Year 3:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{3325}{0.8\times7200} = 0.58\text{ , say 1 machine}$$

Year 4:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{2215}{0.8\times7200} = 0.38\text{ , say 1 machine}$$

Year 5:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{2225}{0.8\times7200} = 0.39\text{ , say 1 machine}$$

Year 6:

$$N_{ik} = \frac{H_{ik}}{W_k \times Availability} = \frac{2584}{0.8\times7200} = 0.45\text{ , say 1 machine}$$

F.5 Hours of Processing Time Required on Machines per Cell

Table F- 5 presents the hours of processing time required on machine types 1-13 in years 1-6 in each cell.

Table F- 5: Hours of Processing Time Required on Machines per Cell (C= Cell, M= Machine, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	2600	0	3625	0	3325	0	1440	775	675	1550	0	2584
M2	0	2600	1350	2725	2025	1975	2565	480	2925	225	3750	0
M3	667	0	1934	0	2359	0	2115	0	2025	0	2250	0
M4	1934	0	3442	0	3592	0	2342	0	1825	0	1584	0
M5	1467	0	2484	0	2534	0	1930	0	1900	0	2167	0
M6	2467	0	2159	1450	1234	2175	525	1547	1050	725	1750	0
M7	1000	467	875	1459	500	1809	0	1870	0	2025	0	2500
M8	0	2867	350	4009	525	3684	1274	1600	1975	750	3000	0
M9	2134	0	1867	1700	1067	2550	0	2239	0	1700	0	1417
M10	0	2534	0	2667	0	1942	0	955	0	1175	0	1584
M11	0	1200	0	2000	0	2025	0	1464	0	1375	0	1500
M12	0	2934	1600	2567	2400	1467	1932	900	1250	1800	750	3000

As a sample calculation, consider the hours required on machine type 1 in cells in the first year. Machine type 1 is required by parts 2, 4, 6, 9, 13 and 16 corresponding to part family *PF1* and parts 5, 15, 17 corresponding to *PF2*. The hours are calculated by an adaptation of equation 40 as follows:

Year 1 – Cell 1:

$$H_{ijk} = \frac{\sum_{p \in Part\ Family\ j} volume_{pk} \times minutes_{ip}}{60}$$

$$H_{111} = \frac{(7\times4+7\times8+9\times4+9\times0+9\times0+9\times4)\times10^3}{60} = 2600\ hours$$

Year 1 – Cell 2:

$$H_{121} = \frac{(7\times0+8\times0+8\times0)\times10^3}{60} = 0\ hours$$

F.6 Sample Calculation - Machines Required per Cell per Year

Consider machine on machine type 1 in year 1. The hours required on this machine by each cell were determined in Appendix F.5, this enables the numbers of machine type 1 required by each cell to be calculated by an adaptation of equation 42 as follows:

Year 1 – Cell 1:

$$N_{111} = \frac{H_{ijk}}{W_k \times Availability} = \frac{2600}{7200 \times 0.8} = 0.45 \text{ , say 1 machine}$$

Year 1 – Cell 2:

$$N_{121} = \frac{H_{ijk}}{W_k \times Availability} = \frac{0}{7200 \times 0.8} = 0$$

Note that the working hours W_k and machine availability were provided in the problem description of section 7.2.

F.7 Part Volumes Requiring Operations on Machines per Cell per Annum

Table F- 6 shows the volumes of parts that require operations on each machine for years 1-6. If a machine is not available in a cell, the corresponding volume of parts will contribute to the total volume of inter-cell part flow.

Table F- 6: Parts* Requiring Operations on Machines per Cell per Annum (C=Cell, M=Machine, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	20	0	26.5	0	23.5	0	9.6	6	4.5	12	0	20
M2	0	16	9	17	13.5	12.5	17.1	3.2	19.5	1.5	25	0
M3	4	0	12.5	0	15.5	0	14.1	0	13.5	0	15	0
M4	12	0	22.5	0	24	0	15.8	0	12	0	10	0
M5	12	0	19.5	0	19.5	0	14.1	0	13.5	0	15	0
M6	20	0	17.5	12	10	18	4.5	12.8	9	6	15	0
M7	8	4	7	12.5	4	15.5	0	15.6	0	16.5	0	20
M8	0	20	3	29.5	4.5	28	9.2	12.8	13.5	6	20	0
M9	16	0	14	12	8	18	0	15.8	0	12	0	10
M10	0	16	0	17	0	12.5	0	6.2	0	7.5	0	10
M11	0	8	0	13	0	13	0	9.4	0	9	0	10
M12	0	20	12	17.5	18	10	14.3	6	9	12	5	20

* Part volumes are represented as thousands of parts.

As a sample calculation, consider machine 1. Machine 1 is required by parts 2, 4, 6, 9, 13 and 16 corresponding to family *PF1* and parts 5, 15 and 17 corresponding to *PF2*.

Table F- 7 shows the volume of parts requiring the services of machine type 1 in Cell 1 for years 1-6. The total per year is calculated by summing the individual part volumes across rows. Individual part volumes were obtained from Table F- 2.

Table F- 7: Volume* of Parts Requiring Machine 1 in Cell 1 (P=Part, Y=Year)

	P2	P4	P6	P9	P13	P16	Total
Y1	4	8	4	0	0	4	20
Y2	3.5	7	3.5	6	3	3.5	26.5
Y3	2	4	2	9	4.5	2	23.5
Y4	0	0	0	6.4	3.2	0	9.6
Y5	0	0	0	3	1.5	0	4.5
Y6	0	0	0	0	0	0	0

* Part volumes are represented as thousands of parts.

Table F- 8 shows the volume of parts requiring the services of machine type 1 in Cell 1 for years 1-6. The total per year is calculated by summing the individual part volumes across rows.

Table F- 8: Volume* of Parts Requiring Machine 1 in Cell 2 (P=Part, Y=Year)

	P5	P15	P17	Total
Y1	0	0	0	0
Y2	0	0	0	0
Y3	0	0	0	0
Y4	1.5	1.5	3	6
Y5	3	3	6	12
Y6	5	5	10	20

* Part volumes are represented as thousands of parts.

F.8 Cost of Inter-Cell Part Flow – Capacity Based Assignment Model

The cost of inter-cell part movements, corresponding to the reconfiguration plan of the capacity based assignment model, is calculated in Table F- 9 . The cost of inter-cell part movement per cell per year is the multiplication of the cost per part and the number of parts. Values of c_k (inter-cell) are presented in Table 7.4. The total cost of inter-cell material handling is R 290620.00, determined by summing individual costs.

Table F- 9: Cost of Inter-Cell Part Flow (C = Cell, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
Flow Volume**	0	4	31	24	26.5	22.5	13.7	15.2	13.5	13.5	5	0
c_k (inter-cell)	R 1.50	R 1.50	R 1.60	R 1.60	R 1.70	R 1.70	R1.80	R 1.80	R 1.90	R 1.90	R 2.00	R 2.00
Cost*	R 0.00	R 6.00	R 49.6	R 38.40	R 45.05	R 38.25	R 24.66	R 27.36	R 25.65	R 25.65	R 10.00	R 0.00

* Amounts are in thousands of Parts, ** Amounts are in thousands of Rand.

F.9 Calculation of Cost Coefficients – Cost Based Assignment Model

The cost of an hour of deficit capacity C_{ijk} with respect to machine *i* in cell *j* in period *k* is based on the cost of inter-cell part movements that will result as a consequence of the deficit. The cost coefficient is determined by estimating the number of parts that are affected by the hour of deficit capacity, multiplied by the cost of an inter-cell movement per part. Table F- 10 presents the coefficients corresponding to the problem of section 7.2.

Table F- 10: Cost Coefficients C_{ijk} (C=Cell, M = Machine, Y=Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	R 12.00	R 12.00	R 12.80	R 12.80	R 13.60	R 13.60	R 14.40	R 14.40	R 15.20	R 15.20	R 16.00	R 16.00
M2	R 10.50	R 10.50	R 11.20	R 11.20	R 11.90	R 11.90	R 12.60	R 12.60	R 13.30	R 13.30	R 14.00	R 14.00
M6	R 13.50	R 13.50	R 14.40	R 14.40	R 15.30	R 15.30	R 16.20	R 16.20	R 17.10	R 17.10	R 18.00	R 18.00
M7	R 12.00	R 13.50	R 12.80	R 14.40	R 13.60	R 15.30	R 14.40	R 16.20	R 15.20	R 17.10	R 16.00	R 18.00
M8	R 10.50	R 12.00	R 11.20	R 12.80	R 11.90	R 13.60	R 12.60	R 14.40	R 13.30	R 15.20	R 14.00	R 16.00
M9	R 12.00	R 12.00	R 12.80	R 12.80	R 13.60	R 13.60	R 14.40	R 14.40	R 15.20	R 15.20	R 16.00	R 16.00
M12	R 12.00	R 10.50	R 12.80	R 11.20	R 13.60	R 11.90	R 14.40	R 12.60	R 15.20	R 13.30	R 16.00	R 14.00

As a sample calculation consider machine 1. This machine is required by parts 2, 4, 6, 9, 13 and 16 corresponding to family *PF1* and parts 5, 15, 17 corresponding to *PF 2*. Process cycle times for parts are presented in Table 7.3.

The average processing time for parts corresponding to *PF1* on machine 1 is:

$$t_{11} = \frac{7+7+9+9+9+9}{6} = 8.33\ min$$

The average processing time for parts corresponding to *PF2* is:

$$t_{12} = \frac{7+8+8}{3} = 7.67\ min$$

The estimated cost per hour capacity deficit with respect to machine type 1 in Cell 1 in year 1 is calculated by equation 89 as follows:

$$C_{111} = \left[\frac{60}{\bar{t}_{11}}\right] c_1 = \left[\frac{60\ min}{8.33\ min}\right] \times 1.5 = 8\ \times 1.5 = R12$$

Likewise the estimated cost per hour capacity deficit with respect to machine type 1 in Cell 2 in year 1 is:

$$C_{121} = \left[\frac{60}{\bar{t}_{12}}\right] c_1 = \left[\frac{60\ min}{7.67\ min}\right] \times 1.5 = 8\ \times 1.5 = R12$$

Since the number of parts performing an inter-cell movement per hour capacity deficit is based on an average time. There will therefore be some discrepancy between the cost of inter-cell movements as determined by the cost minimization assignment model and the actual inter-cell material cost associated with its solution.

If a conservative approach is preferred, the use of the minimum cycle time among all parts in the family can be used with equation 89. This would result in higher cost coefficients and the model would effectively yield an upper bound on material handling cost if the associated solution is implemented.

F.10 Cost of Inter-Cell Part Flow – Cost Minimization Assignment Model

The cost of inter-cell part movements, corresponding to the reconfiguration plan of the capacity based assignment model, is calculated in Table F- 11. The cost of inter-cell part movement per cell per year is the multiplication of the cost per part and the number of parts. Values of c_k (inter-cell) are presented in Table 7.4. The total cost of inter-cell material handling is R 296620.00, determined by summing individual costs.

Table F- 11: Cost of Inter-Cell Part Flow (C = Cell, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
Flow Volume**	8	0	31	24	26.5	22.5	13.7	15.2	13.5	13.5	5	0
c_k (inter-cell)	R 1.50	R1.50	R1.60	R1.60	R1.70	R1.70	R1.80	R 1.80	R1.90	R1.90	R2.00	R2.00
Cost*	R 12.00	R 0.00	R 49.6	R 38.4	R 45.05	R 38.25	R 24.66	R 27.36	R 25.65	R 25.65	R 10.00	R 0.00

* Amounts are in thousands of Parts, ** Amounts are in thousands of Rand.

F.11 Total Part Load per Cell per Year

Table F-12 presents the total part load per cell per year. The total part load is the total number of parts to be manufactured by a cell, corresponding to its own part family.

Table F-12: Total Part Load per Cell per Year (C=Cell, Y=Year)

	Y1	Y2	Y3	Y4	Y5	Y6
C1	20000	32500	32500	23500	22500	25000
C2	20000	29500	28000	18800	18000	20000

As a sample calculation consider Cell 1 in year 1. In year 1 parts 2, 4, 6 and 8 are to be manufactured in the cell. The sum of their individual volumes, as read from Table F- 2, is 4000+8000+4000+4000 = 20000.

F.12 Revised Factory Configuration Template

Figure F- 1 shows the updated FCT, the site corresponding to machine 7 in Cell 1 was eliminated.

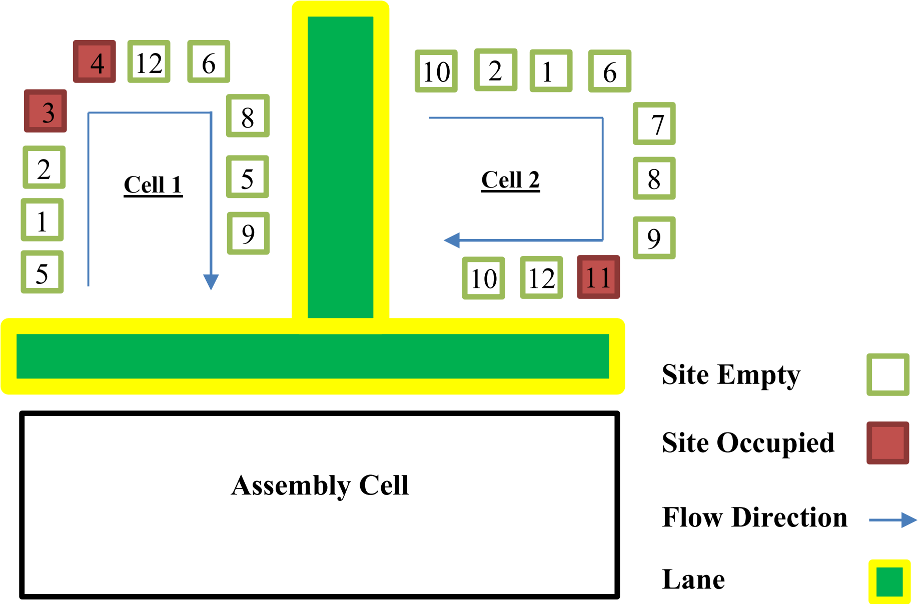

Figure F- 1: Updated Factory Configuration Template

F.13 Intra-Cell Machine Assignment Model – Results

Table F- 13 and Table F- 14 present the variable values to the best solution of the intra-cell machine assignment model for machine 5 and machine 10 respectively.

Table F- 13: Best Solution - Intra-Cell Assignment for Machine 5 in Cell 1

	x_k	a_k	b_k	y_k	c_k	d_k
$k=1$	1	1	0	0	0	0
$k=2$	1	0	0	0	0	0
$k=3$	0	0	1	1	1	0
$k=4$	0	0	0	1	0	0
$k=5$	0	0	0	1	0	0
$k=6$	0	0	0	1	0	0

Table F- 14: Best Solution Intra-Cell Assignment for Machine 10 in Cell 2

	x_k	a_k	b_k	y_k	c_k	d_k
$k=1$	0	0	0	1	1	0
$k=2$	0	0	0	1	0	0
$k=3$	0	0	0	1	0	0
$k=4$	1	1	0	0	0	1
$k=5$	1	0	0	0	0	0
$k=6$	1	0	0	0	0	0

F.14 Sample Calculation of GT Performance Measures

Table F- 15: Machine Component Incidence Matrix with Block Structures – fixed CMS – Year 3

	2	4	6	9	10	13	16	18	1	3	7	11	14	19	20
1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	0	1	0	0	1	1	1	0	1
3	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
4	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0
5	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0
12	0	0	0	1	0	1	0	1	1	0	1	1	1	0	0
6	1	1	1	0	0	0	1	0	0	1	0	0	0	1	1
7	1	0	0	0	0	0	1	0	1	1	0	0	0	1	0
8	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1
9	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1
10	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
11	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1

In this sample calculation, the cell and part family configurations will be examined in the third year of operation. Note that parts not being produced in this year are omitted from the machine-component incidence matrix. Table F- 15 presents the machine-component incidence matrix with block structures; note that machine numbers are represented as row indices and part numbers are represented as column indices.

The characteristics of the matrix are as follows: $e_1 = 50, e_v = 40, e_o = 18, e = 68, o = 112$. Assume $q=0.5$, giving equal significance to machine utilization and inter-cell part movement.

Group Efficiency

$$\eta_1 = \frac{e_1}{e_1 + e_v} = \frac{50}{50 + 40} = 0.56$$

$$\eta_2 = \frac{o - e_v}{(o - e_v) + (e - e_1)} = \frac{112 - 40}{(112 - 40) + (68 - 50)} = 0.8$$

$$\eta = q\eta_1 + (1 - q)\eta_2 = 0.5(0.56) + (1 - 0.5)0.8 = 0.68$$

Group Efficacy

$$\tau = \frac{e - e_o}{e - e_v} = \frac{68 - 18}{68 - 40} = 1.79$$

Weighted Average Efficacy

$$\gamma = \frac{q(e - e_o)}{q(e + e_v - e_o) + (1 - q)e_o} = \frac{0.5(68 - 18)}{0.5(68 + 40 - 18) + (1 - 0.5)18} = 0.46$$

Group Measure

$$\xi_g = \xi_u - \xi_m = \frac{e_1}{e_1 + e_v} - \frac{e_o}{e} = \frac{50}{50 + 40} - \frac{18}{68} = 0.29$$

Utilization Measure

$$U = \frac{e_1}{e_1 + e_v} = \frac{50}{50 + 40} = 56$$

Containment Measure

$$C = \frac{e - e_o}{e} = \frac{68 - 18}{68} = 74$$

Exceptional Elements

$e_o = 18$

Voids

$e_v = 40$

This page was intentionally left blank.

Appendix G

In this appendix a CMS with a fixed layout is generated for the problem data presented in section 7.2. The fixed CMS was developed using many of the same algorithms with the exception that each machine, once placed in a cell, could not be relocated.

G.1 Group Technology Solution

The cell and part family combinations for the CMS with a fixed layout were generated by a combination of the FCMdd algorithm and a modified version of the SFC heuristic. The structure of the modified SFC heuristic is exactly the same as described in section 4.5.2 with the exception of the cluster formation rule. For both machines and parts the following crisp cluster formation rule was applied: $\tilde{\mu}_{ik} = 1$ if $i = \underset{1 \le i \le C}{\arg\max}\{\mu_{ik}\}$ and $\tilde{\mu}_{ik} = 0$ otherwise.

In instances where a machine or part had a maximum membership that was tied across multiple clusters, the element was allocated to the cell/family with the least members. Under this cluster formation rule the parameter L is irrelevant. Table G- 1 lists the parameters that were used with the FCMdd algorithm and SFC heuristic.

Table G- 1: Parameters – FCMdd algorithm and SFC heuristic

FCMdd Algorithm		**SFC Heuristic**	
Binary Dissimilarity Measure	Jaccard	***m***	2
Maximum Iterations	10	**Number of Clusters**	2
m	2	**Maximum Iterations**	50
Number of Clusters	2		

The resulting machine cells and part families were:

MC1 = {1, 2, 3, 4, 5, 6};
MC2 = {7, 8, 9, 10, 11, 12};
PF1 = {2, 4, 6, 8, 9, 10, 12, 13, 16, 18};
PF2 = {1, 3, 5, 7, 11, 14, 15, 17, 19, 20}.

The part families in this solution are the same as those established for the RCMS solution in section 7.3. The number of parts requiring the services of each machine in each in each cell will be the same and Table F- 6 applicable. The calculation of the mandatory number of machines to be purchased is also the same, as shown in Table 7.7 . One mandatory machine of each type will be made available.

Table G- 2 presents the machine cells and part families in the format of a machine-component incidence matrix with block diagonals. Note that machine numbers are listed as row indices and part numbers are listed as column indices.

Table G- 2: Machine-Component Matrix with Block Diagonals

	2	**4**	**6**	**8**	**9**	**10**	**12**	**13**	**16**	**18**	**1**	**3**	**5**	**7**	**11**	**14**	**15**	**17**	**19**	**20**
1	1	1	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	1	0	0
2	0	0	0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	0	0	1
3	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
4	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
5	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1
7	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0
9	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1
8	0	0	0	1	0	1	1	0	0	0	1	1	0	1	1	1	0	0	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1
12	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1	1	0	0

An examination of Table G- 2 reveals that there are 16 exceptional elements outside of the block diagonal structures and 38 voids within the structures. The calculation of inter-cell part flow volumes per year based on the fixed cell configurations is presented in Appendix G.2.

G.2 Calculation of Inter-Cell Part Movements and Cost

Note that one inter-cell part movement is counted for each machine required by a part, which is not within its designed cell. Upon closer inspection, it may be possible for the part to visit multiple machines that are not present in the relevant cell by a single inter-cell move (i.e. combine inter-cell visits). Combined visits are only possible if the sequence in which processes must occur is favourable. Combined visits are not considered here, as in the calculations with respect to the RCMS design. All calculations on volumes of inter-cell moves therefore represent the worst case scenario of one inter-cell move per missing machine.

Table G- 3 shows the volumes of parts that will perform inter-cell movements based one the configurations of Cell 1 and Cell 2. Table G- 4 shows the cost of inter-cell part flow; the cost is calculated by the multiplication of the cost per part and the number of parts. There is a total of 302500 inter-cell part movements at a total cost of R 526200.00.

Table G- 3: Inter-Cell Part Flow Volumes* (C=Cell, M=Machine,Y=Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
M1	0	0	0	0	0	0	0	6	0	12	0	20
M2	0	16	0	17	0	12.5	0	3.2	0	1.5	0	0
M3	0	0	0	0	0	0	0	0	0	0	0	0
M4	0	0	0	0	0	0	0	0	0	0	0	0
M5	0	0	0	0	0	0	0	0	0	0	0	0
M6	0	0	0	12	0	18	0	12.8	0	6	0	0
M7	8	0	7	0	4	0	0	0	0	0	0	0
M8	0	0	3	0	4.5	0	9.2	0	13.5	0	20	0
M9	16	0	14	0	8	0	0	0	0	0	0	0
M10	0	0	0	0	0	0	0	0	0	0	0	0
M11	0	0	0	0	0	0	0	0	0	0	0	0
M12	0	0	12	0	18	0	14.3	0	9	0	5	0
Total	24	16	36	29	34.5	30.5	23.5	22	22.5	19.5	25	20

* Amounts are in thousands of Parts

Table G- 4: Cost of Inter-Cell Part Flow (C = Cell, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
Flow Volume**	24	16	36	29	34.5	30.5	23.5	22	22.5	19.5	25	20
Cost pp	R 1.50	R 1.50	R 1.60	R 1.60	R 1.70	R 1.70	R 1.80	R 1.80	R 1.90	R 1.90	R 2.00	R 2.00
Cost*	R 36.00	R 24.00	R 57.60	R 46.40	R 58.65	R 51.85	R 42.30	R 39.60	R 42.75	R 37.05	R 50.00	R 40.00

* Amounts are in thousands of Parts, ** Amounts are in thousands of Rand

The arrangement of machines in cells for unidirectional flow will be presented in Appendix G.3.

G.3 Line Analysis and Intra-Cell Back Flow Volumes

One machine of each type is available to the fixed CMS. Since one machine of each type is available and the re-layout of machines is not permitted here, the arrangement of machines in cells will be optimized for unidirectional flow using the Tabu Search algorithm of section 5.3.2. The application of the TS algorithm to cell yielded the machine arrangements as shown in Table G- 5.

Table G- 5: TS Results – Best Arrangement of Machines

	Sequence of Machines					
Location	**1**	**2**	**3**	**4**	**5**	**6**
Cell 1	1	2	3	4	6	5
Cell 2	10	7	8	9	11	12

Figure G- 1 reveals that the best solution for Cell 1 imposed three part backtracking movements. Figure G- 2 reveals that the best solution for Cell 2 also imposed three part backtracking movements. In both cases no improvement was found although the algorithm was allowed to search for 1000 iterations.

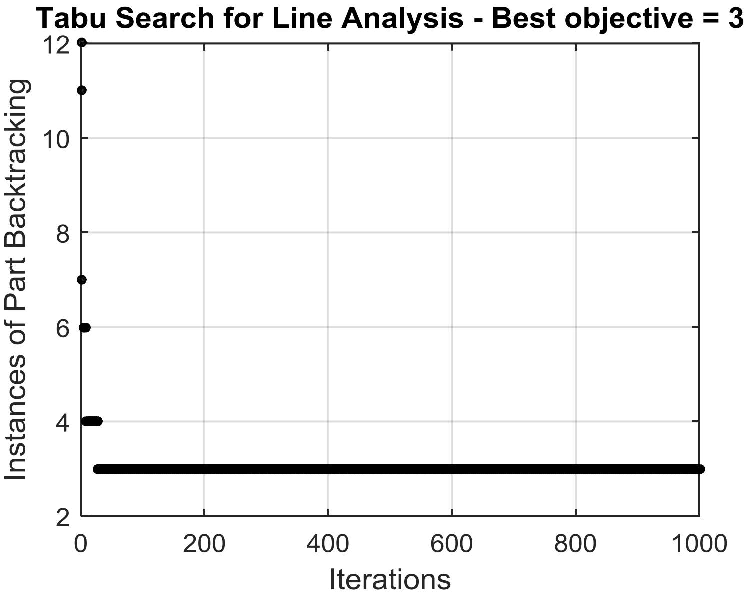

Figure G- 1: TS Algorithm – Best *f per Iteration – Cell 1**

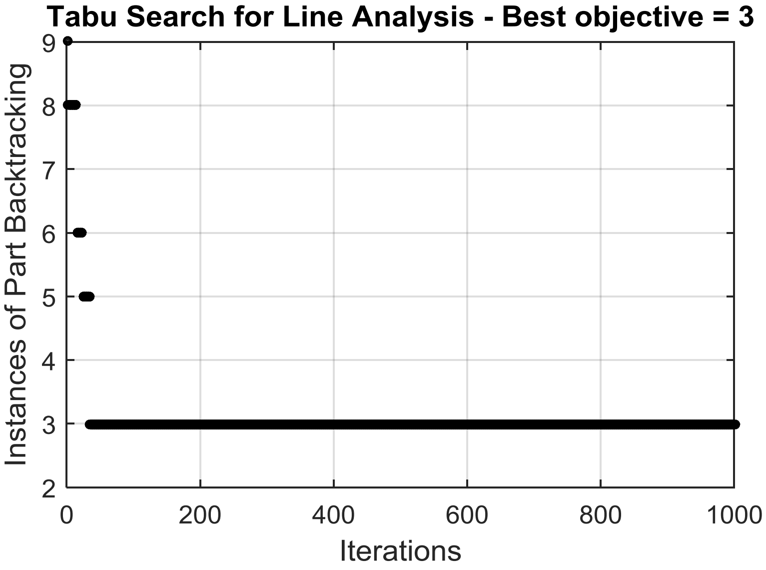

Figure G- 2: TS Algorithm – Best *f per Iteration – Cell 2**

By examining the sequence in which machines will be arranged in Cell 1, parts 2, 6 and 16 will perform inter-cell backtracking movements. Parts 7, 11 and 14 will perform inter-cell backtracking movements in Cell 2. Table G- 6 and Table G- 7 present the volumes of parts that will perform intra-cell backtracking movements in Cell 1 and Cell 2 respectively.

Table G- 6: Volumes of Parts Performing Backtracking Movements in Cell 1* (P=Part, Y= Year)

	P1	P6	P16	Total Back Flow
Y1	4	4	4	12
Y2	3.5	3.5	3.5	10.5
Y3	2	2	2	6
Y4	0	0	0	0
Y5	0	0	0	0
Y6	0	0	0	0

* Amounts are in thousands of Parts

Table G- 7: Volumes of Parts Performing Backtracking Movements in Cell 2* (P=Part, Y= Year)

	P7	P11	P14	Total Back flow
Y1	4	4	8	16
Y2	3.5	3.5	7	14
Y3	2	2	4	8
Y4	0	0	0	0
Y5	0	0	0	0
Y6	0	0	0	0

* Amounts are in thousands of Parts

Table G- 8: Cost of Intra-Cell Part Back Flow (C = Cell, Y = Year)

	Y1		Y2		Y3		Y4		Y5		Y6	
	C1	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**	**C1**	**C2**
Flow Volume**	12	16	10.5	14	6	8	0	0	0	0	0	0
Cost pp	R 1.00	R 1.00	R 1.10	R 1.10	R 1.20	R 1.20	R 1.30	R 1.30	R 1.40	R 1.40	R 1.50	R 1.50
Cost*	R 12.00	R 16.00	R 11.55	R 15.40	R 7.20	R 9.60	R 0.00	R 0.00	R 0.00	R 0.00	R 0.00	R 0.00

* Amounts are in thousands of Parts, ** Amounts are in thousands of Rand

The cost of intra-cell part backtracking is calculated in Table G- 8 using the cost per part provided in the problem definition of section 7.2. There is a total back flow volume of 66500 parts in cells at cost of R 71750.00.

G.4 Sample Calculation of GT Performance Measures

In this sample calculation, the cell and part family configurations will be examined in the third year of operation. Note that parts not being produced in this year are omitted from the machine-component incidence matrix. Table G- 9 presents the machine-component incidence matrix with block structures; note that machine numbers are represented as row indices and part numbers are represented as column indices.

Table G- 9: Machine Component Incidence Matrix with Block Structures – fixed CMS – Year 3

	2	4	6	9	10	13	16	18	1	3	7	11	14	19	20
1	1	1	1	1	0	1	1	0							
2	0	0	0	0	1	1	0	1			1	1	1		1
3	0	0	0	0	1	1	1	1							
4	0	1	1	1	0	1	0	1							
5	1	0	1	1	1	0	1	0							
6	1	1	1	0	0	0	1	0		1				1	1
7	1						1		1	1	0	0	0	1	0
8					1				1	1	1	1	1	1	1
9	1	1	1						0	1	0	0	0	1	1
10									0	0	1	1	1	0	1
11									1	0	0	1	0	1	1
12				1		1		1	1	0	1	1	1	0	0

The characteristics of this matrix are as follows: $e_1 = 52$, $e_v = 38$, $e_o = 16$, $e = 68$, $o = 112$. Assume $q=0.5$, giving equal significance to machine utilization and inter-cell part movement.

Group Efficiency

$$\eta_1 = \frac{e_1}{e_1 + e_v} = \frac{52}{52 + 38} = 0.58$$

$$\eta_2 = \frac{o - e_v}{(o - e_v) + (e - e_1)} = \frac{112 - 38}{(112 - 38) + (68 - 52)} = 0.82$$

$$\eta = q\eta_1 + (1 - q)\eta_2 = 0.5(0.58) + (1 - 0.5)0.82 = 0.7$$

Group Efficacy

$$\tau = \frac{e - e_o}{e - e_v} = \frac{68 - 16}{68 - 38} = 1.73$$

Weighted Average Efficacy

$$\gamma = \frac{q(e - e_o)}{q(e + e_v - e_o) + (1 - q)e_o} = \frac{0.5(68 - 16)}{0.5(68 + 38 - 16) + (1 - 0.5)16} = 0.49$$

Group Measure

$$\xi_g = \xi_u - \xi_m = \frac{e_1}{e_1 + e_v} - \frac{e_o}{e} = \frac{52}{52 + 38} - \frac{16}{68} = 0.34$$

Exceptional Elements

$e_o = 16$

Voids

$e_v = 38$