Computer-aided design of cellular manufacturing layout.

Wu, Yue

COMPUTER-AIDED DESIGN OF CELLULAR MANUFACTURING LAYOUT

A thesis submitted to the
University of Durham
for the degree of
Doctor of Philosophy

by

Yue Wu

School of Engineering
University of Durham

1999

To my wife *Xiaojing* and my parents

ABSTRACT

Manufacturing layout is one of the important features of a manufacturing system, and can have a dominating effect upon the management of the plant, adaptability, efficiency, and the cost for the internal material handling. Cellular manufacturing is a strategy, which groups machines required to manufacture similar components into machine cells, to reduce the material handling cost and setup time. This research aims at the optimisation of cellular manufacturing layout using three stages: cell formation stage, cell layout stage and plant layout stage. In the first stage, a heuristic based upon the material flow is developed, which allows the machines to be clustered more naturally. In the second stage, a genetic algorithm is used to optimise the machine layout within each cell, in which a previously defined material handling system is provided. In the third stage, a block-tree representation for a slicing floorplan problem is developed. This representation allows the design of the layout and aisle-structure simultaneously, and can be converted into a string representation adaptable by a genetic algorithm for optimisation. Genetic operators for the string representation for this purpose are proposed. The number of aisles in the optimised aisle-structure is also minimised. A computational experiment was carried out based upon solving three problems from the literature. The results showed that cell configuration and layout planning should be considered simultaneously. Finally, an outline of an integrated design method for whole system optimisation is provided.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor, Professor Ernest Appleton, for his trust, guidance and full support over the period of this research. During my studies at Durham, his ideas and inspiration in the discussion let me not only learn many aspects in depth in the area of manufacturing, but also broaden my horizons in the related fields. Without his continuous supervision and encouragement, I would not be in a position to submit this thesis.

I am deeply grateful to Mr. Trevor Nancarrow for his support on computing. I would also like to thank Mr. John Garside, Dr. Neil Strutchury, Mrs. Wendy Lister, Miss Amanda Temby, Miss Angela Jones, Mrs. Janice Smith, Mr. Emad Summad, Dr. Zhihui Yao, Dr. Hugh Bradley, Dr. Ian Carpenter, Mr. Michael Betteridge and Mr. Rob Baker for their kind help during the period of the study.

Finally, I would like to express my special thanks to School of Engineering, University of Durham for the studentship, which made it possible for me to study for the degree of PhD at Durham.

DECLARATION

I hereby declare that no part of the work presented in this thesis has been previously submitted for any degree and is being currently submitted in candidature for any other degree in this or any other university.

Signed ....Yue Wu....

The work reported in this thesis was carried out by the candidate. Any work not carried out by the candidate is acknowledged in the main text.

Signed ....
Supervisor

Signed ....Yue Wu....
Candidate

STATEMENT OF COPYRIGHT

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

This section presents a comprehensive list of the algebraic notation used within this thesis, along with a short description and the appropriate standard units, if any. For ease of reference, brief descriptions are also given at the first occurrence of any item of notation in each chapter.

Symbol	Meaning
$\boldsymbol{A}$	part-machine incidence matrix.
α	a weight on the mass flow and part types, $0 \leq \alpha \leq 1$.
a_{ij}	an element of the incidence matrix A, $i \in \{1,\cdots,m\}$, $j \in \{1,\cdots,n\}$.
a_i	area of department *i*,
a^F	area of the shop floor,
a_i^s	the *i*th element in the aisle-structure string, $i = 1, 2, \ldots, m\text{-}1$.
b_{xsub}	lower border of the vertical side of the plant area.
b_{xup}	upper border of the vertical side of the plant area.
b_{ysub}	lower border of the horizontal side of the plant area.
b_{yup}	upper border of the horizontal side of the plant area.
c_{jl}	cost of transporting unit material from location *j* to location *l*.
c_{ij}^r	numerical value of the closeness rating between machine *i* and machine *j*.
$\mathbf{C}$	set of machines in a machine cell.
C_{lji}	processing time to perform operation *l* of part *j* on machine type *i*.
d_{ij}	distance between machine *i* and machine *j*.
dc_k	minimum safe distance on the input side of machine *k*.
dp_l	minimum safe distance on the output side of machine *l*.
D_j	average demand for part *j* (in units) per period.
e_{ij}	number of the cut-lines, that are not included in the aisle-structure string a_t^s, for path p_{ij},

E intercell material handling cost for every unit of material volume moving a unit of distance.

f_{ij} flow density between machine *i* and machine *j*.

F_{cn} flow between the new cell and the rest of the machines.

F_{co}^{v} flow between the current cell and all the other machines.

F_{mc} the number of part types flowing between the candidate and the current cell.

F_{mr} the number of part types flowing between the candidate and the sets of clustered cells and the remaining machines.

G_i variable cost per unit time to operate a machine of type *i*.

h_i length of the horizontal side of the machine *i*.

H material handling cost per intercell move.

L_c lower bound of machine cells.

L_f length of the shop floor.

L_m lower bound of machines.

n number of part types.

m number of machines.

p number of part groups.

p^e penalty on the excluded aisles that is essential for a necessary path ,

p_{ij} a set of cut-lines for a path of $\mathbf{P}_{ij}$.

$\mathbf{P}_{ij}$ a set of the shortest paths from department *i* to department *j*.

ρ_i intercell transportation lot size of part *i*.

ρ_i' transportation lot size of part *i* within a cell.

ω number of operation types.

q number of cells.

r_m weighted flow ratio.

r_s floor shape limit, $r_s \leq 1$.

R_i cost to train a worker to operate a machine of type *i*.

s_{ij} value of similarity between part *i* and part *j*.

t number of tool types.

T_t	cost per period to stock tool type t.
U_a	is the area utilisation limit,
U_m	upper bound of machines.
U_c	upper bound of machine cells.
v_i	length of the vertical side of the machine i.
V_i	amortised cost per period to procure a machine of type i.
w	number of workers.
w_i	weight parameter.
W_f	width of the shop floor.

ACRONYMS

This section presents a list of acronyms used within this thesis, along with the full names of the acronyms. For ease of reference, the full names and their acronyms are also given at the first occurrence in each chapter.

AGV	automatic guided vehicle
AI	artificial intelligence
ANN	artificial neural networks
BNC	basic natural cell
CIMS	computer integrated manufacturing system
CMS	cellular manufacturing system
CSCR	cell safety combination rate
CX	cycle crossover
FMS	flexible manufacturing system
GA	genetic algorithm
GT	group technology
HMS	material handling system
HME	material handling equipment
IMF	inter-machine flow
JIT	just in time
NCE	natural cell exploring
OX	order crossover
PMX	partially matched crossover
RLP	row layout problem
RS	random search
SA	simulated annealing
SCR	safety combination rate
SFT	segment flow topology
QAP	quadratic assignment problem
QSP	quadratic set covering problem
VLSI	very large scale integration
WIP	work in progress

Chapter One

Introduction

This chapter describes the manufacturing system design problems involved in the research and the objectives of this research. The chapter also outlines the research and the structure of the thesis.

1.1 Background

In the last decade of this century, manufacturing companies all over the world face stiffer competition than ever before and consumers tend to pursue more diversified products and higher quality. As product life cycles shorten, mass production techniques have been found to be inflexible for meeting orders which are increasingly small in size, large in variety and tight in deadlines to meet current market requirements. As the ability to respond quickly to changing market requirements is becoming more important, flexible manufacturing systems (FMS) and cellular manufacturing systems (CMS) provide an opportunity to adjust to these changes. FMS and CMS are two applications of manufacturing systems and are important modern techniques in advanced manufacturing.

1.2 Manufacturing Systems

Introducing a product into the market generally needs the following stages. Firstly, there is the requirement for a demand for the products by consumers. Secondly, the product needs to be carefully designed to fulfil the requirements of the consumers. Next, naturally occurring or artificial raw materials are changed to the products according to the design. Then, the products are sent to consumers. Finally, services for the maintenance of product are sometimes necessary to ensure satisfactory use of the product. The objective of design and manufacturing is to provide low-price, quality products to meet customers' requirements. A beneficial cycle, in which consumers obtain the products they want, while merchants, designers and manufacturers make profits, is to everybody's advantage. In short, manufacturing is a production procedure that converts raw materials into useful products.

Accordingly, a manufacturing system can be defined as a production system whose function is to convert raw materials into desired products (Riggs 1987, Harrington 1984 and Gaither 1992). A manufacturing procedure converting raw materials into final products can be further divided into sub-operations which produce parts, sub-assemblies, etc. An operation can be defined as a production procedure covering a number of process stages. For example, a single operation can be regarded as producing a part in a manufacturing cell; cutting a feature on an individual machine, or even just a step in assembling or packaging, depending on the point of view at the manufacturing system. According to the layout of facilities, manufacturing systems can be classified generally into three categories, functional manufacturing systems, mass production systems and flexible manufacturing systems or cellular manufacturing systems.

1.2.1 Functional Manufacturing Systems

A functional manufacturing system, which may also be known as a job shop, usually pools the machines of the same type together and product flows according to the process sequences. In a job shop, the types of machine tools are general and flexible enough to meet a variety of product needs. Among the pool of the machines of the same type, loading balance can be achieved without much effort and can be used to

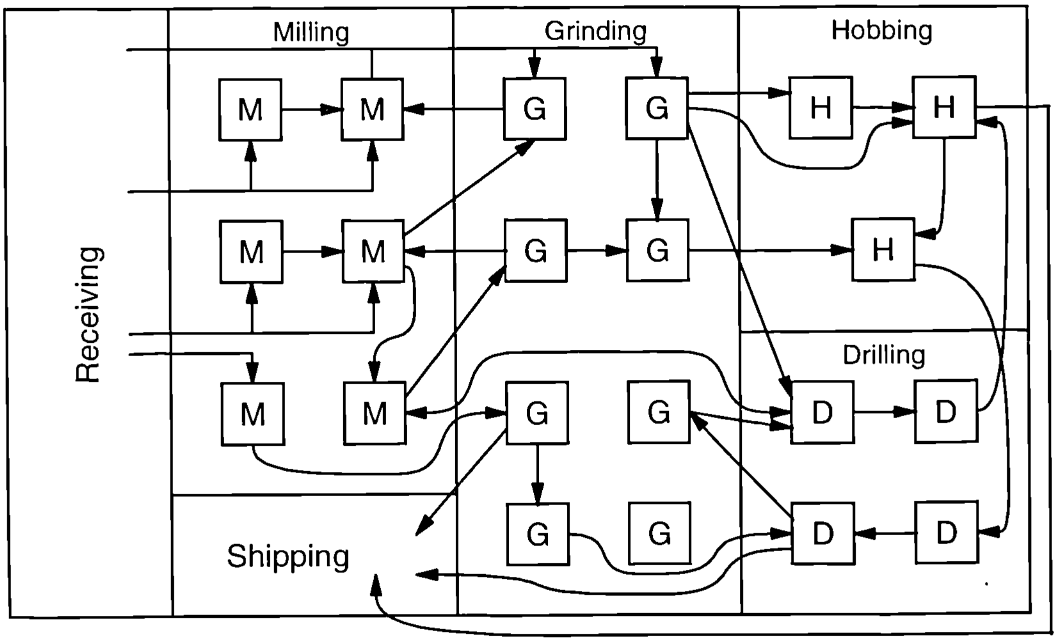


Figure 1-1 Job shop: jumbled flow (Vonderembse and White 1991)

eliminate the restriction or impact of a bottleneck machine. However, as a trade-off, the material flow in a job shop is frequently complex as shown in Figure 1-1 (Vonderembse and White 1991).

1.2.2 Mass Production Systems

In a mass production system, each product flows along a line in which most of machines are dedicated to a certain type of product, which dominates the flow to achieve the lowest material handling cost and highest productivity. Because of the simple material flow and the use of dedicated machines, this system can only accommodate little or no product variation and therefore, it has little flexibility to adapt to changing markets.

1.2.3 Flexible Manufacturing Systems

A flexible manufacturing system is defined as a production system which typically consists of a set of identical and/or complementary numerically controlled machines which are connected through an automated transportation system (Tempelmeier and Kuhn 1993). Flexible manufacturing systems are sometimes known as Computer Integrated Manufacturing Systems (CIMS). In this thesis, the term flexible manufacturing system can be regarded as a wider definition in which some of the machines or transport in the system may be operated manually.

FMS can offer a number of advantages for manufacturers:

- Flexibility within a family of parts.
- Random feeding of parts.
- Simultaneous production of different parts.
- Decreased set-up times.
- More efficient machine usage.
- Decreased direct and indirect labours costs.
- Low work in progress (WIP).

1.2.4 Cellular Manufacturing Systems

A Cellular manufacturing system is a manufacturing system based on groups of processes, people and machines to produce a specific family of products with similar manufacturing characteristics (Apple 1977). In a cellular manufacturing system, each cell may perform as an FMS. An illustration of a cellular manufacturing system is given in Figure 1-2 (Vonderembse and White 1991). A cellular manufacturing system could consist of independent cells, between which there is no material flow, or hybrid cells, in which shared machines are allowed.

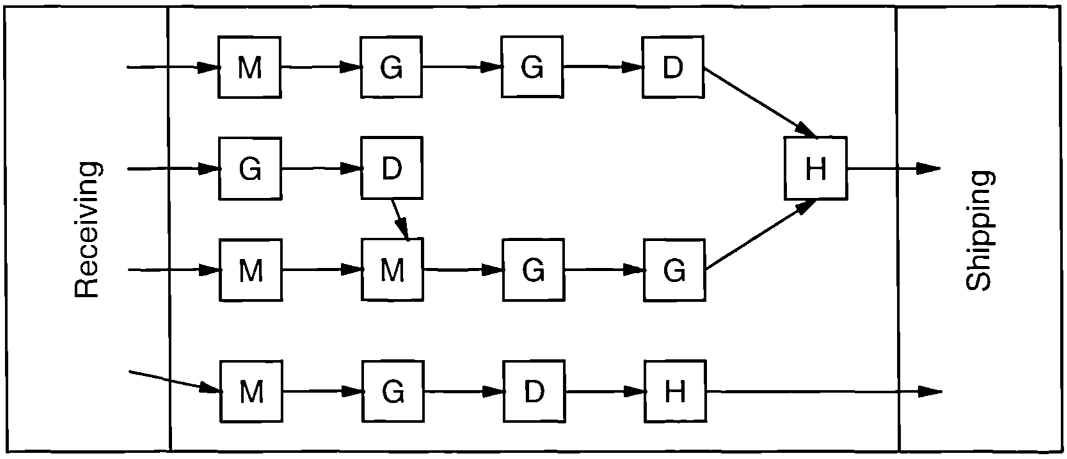


Figure 1-2 Families of parts and manufacturing cells: organised flow (Vonderembse and White 1991)

The benefits of cellular manufacturing include (Gallagher and Knight 1986):

- Reduction of material handling cost.
- Organised material flow.
- Reduction of work-in-progress inventory.

- Reduction of setup time.
- Better lead times and fast response.
- Improvement of resource utilisation.
- Improvement of the productivity of the systems.
- Improved quality.
- Improved customer service.

In general, a mass production system is most appropriate for parts with high demand and low variety. A functional production system is best suited for a product mix characterised by high variety and low demand, while a cellular manufacturing system is in the middle and has more flexibility to adapt to product variety and demand. A comparison of different kinds of manufacturing systems based on their production rate and production variety is illustrated as in Figure 1-3 (Black 1991). Agarwal and Sarkis (1998) gave a recent review and analysis of the performance comparison between functional and cellular manufacturing systems. The review reveals contradictory results from previous simulation and empirical studies and shows the complexity of the evaluation and the simulation of the systems.

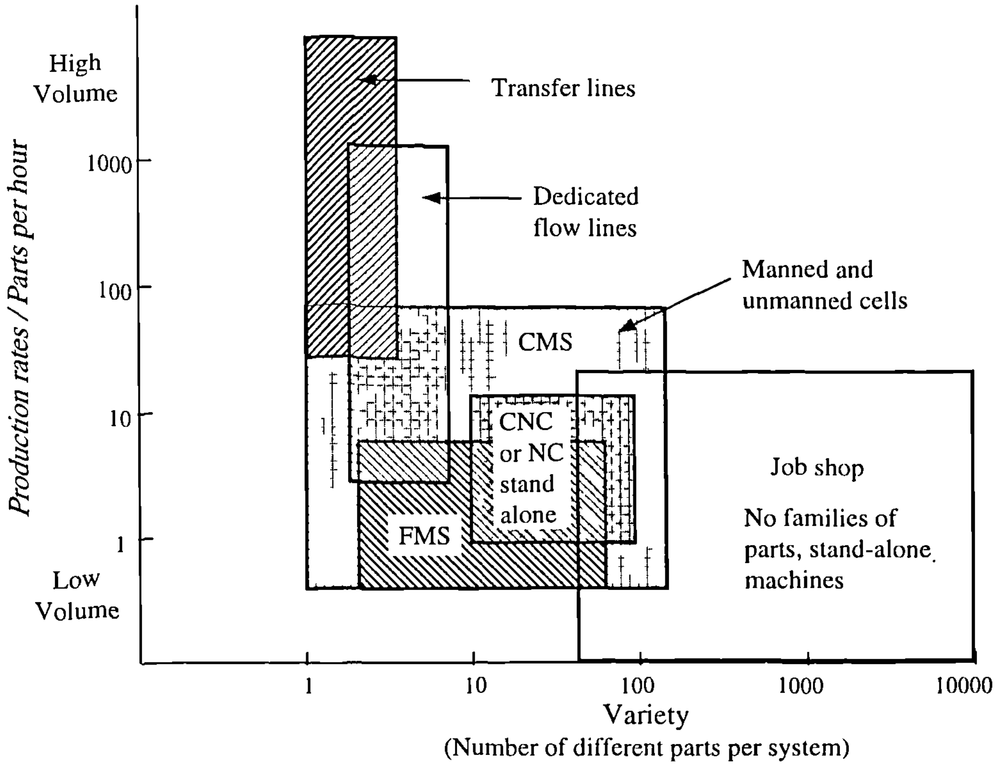


Figure 1-3 Comparison of different kinds of manufacturing systems (Black 1991)

1.3 Designing Cellular Manufacturing Systems

There are two kinds of manufacturing system design problems, the green field problem and the brown field problem. In a brown field problem, the factory already exists. The existing shop floor, machines, transport and layout should be taken into consideration in the design of the new system. Thus, the design in a brown field site can be considered to be just a 'renewal' of the existing system. On the other hand, there are no such constrains in a green field layout and designers are free to select the most suitable machines, transport and layout for the factory. More decisions have to be made in solving the green field problem. Decisions that should be made in solving green field design problems and may sometimes be required in a brown field design are discussed below.

1.3.1 Job Assignment Problem

To produce a product, each operation to be carried out on the product has to be assigned to a certain type of machine. These operations and the operation sequences are decided by process planning after the product design stage. The assignment may have a number of choices. For instance, a hole in a part could be performed on a drilling machine, a milling machine, a machining centre, a boring machine or sometimes a lathe. This problem is known as the job machine assignment problem or machine assignment problem (Cheng 1992, Kamel *et al* 1994, Shanker and Agrawal 1997). This decision should consider the investment and running cost of the machine type, the location of the machine tool, the scheduling of the machine and the loading balance among the other machines. In a green field design, the job assignment problem should be solved simultaneously with the cell formation problem and layout problem.

1.3.2 Cell Formation Problem

Cell formation assigns machines or machine types to cells in order to simplify the flow and minimise the material handling cost. The decisions include how many cells are in the plant, also which and how many machines of a machine type should be clustered into a machine cell. As in the case of the job assignment problem, the investment and running cost of each machine type, the location of the machine, the loading balances among the machines or cells, the scheduling of parts at each machine and the tooling within cells should be taken into account. The flexibility of cells may also need to be considered corresponding to the policy of the company.

1.3.3 Layout Planning Problem

Based upon cell formation, layout planning decides the dimensions of the plant floor and location of the cells, machines and auxiliaries. The objective of layout planning is mainly the reduction of material handling cost. The problem is solved in conjunction with transportation system design, product planning and scheduling, because it should leave spaces for inventory and transport on the plant floor. Not only are the above problems highly interactive with layout planning, the layout problem itself is also complicated and difficult. The following is a description of the layout problem given by Demes *et al* (1993). "Layout design tackles many complex issues that typically arise in the design of artefacts that must satisfy spatial and functional constraints. A potentially infinite number of location and orientation combinations are available for placing any single object. In each combination design objects interact through their shapes, sizes, and their spatial or topological relations. Layout design decisions must simultaneously satisfy global requirements (e.g., use of space) and local requirements (e.g., adjacencies between pairs of objects). An acceptable spatial arrangement often exhibits a complex pattern of trade-offs. For these reasons, an exploration of the structure of the layout task and a search for candidate solutions is required. Due to cognitive limitation, human designers have limited capability for making systematic explorations of alternative arrangements. This shortcoming in human performance has motivated numerous attempts to apply computational methods to layout. What is desired is a structured method for producing alternatives, each of which embodies trade-offs that can be understood and justified." This thesis is a response to the challenge described in the concluding sentence.

1.3.4 Transportation System Design Problem

In order to transfer products between machines or inventories, a proper transportation system or systems should be carefully designed to match the machine layout. The design includes the arrangement of the aisle structure for the movement of parts/products and workers and the selection of the types of the transports, their speed and loading ability. The production management policy, such as just in time (JIT), has a strong impact on the transportation system design.

1.3.5 *Planning and Scheduling Problem*

Although planning and scheduling are production management problems that occur after the manufacturing system has been built, some of the planning and scheduling aspects may also be involved in the manufacturing system design, for example, operation batch size, transportation batch size, capacity of inventories, and so forth. These aspects also impact strongly on the behaviour of the system during the system evaluation.

1.4 Research Objectives

Manufacturing system design is a complex creative decision activity which applies scientific knowledge, engineering theories and technologies to process a large quantity of information about products, machines, human resources and markets. Optimising the system is much more complicated, not only for human designers, but also for computers. This research concentrated on the layout design and its related problems. The following presumptions omit some of the aspects previously described to simplify the problem. Some detailed assumptions for each sub-problem are discussed in the relevant chapters.

1.4.1 Assumptions and Simplifications

This research concentrated on the manufacturing system layout design on a green field. The facilities of the manufacturing system are in a planar plant. The products and their parts are known. The predicted volumes of the products have been given as an average. The operations of each part have been assigned to machines or machine types by process planning. Human resources and their effects are ignored.

1.4.2 Objectives of the Research

The objectives of this research were as follows:

1. To investigate the development of a method for cell formation.
2. To develop a method for cell layout with a consideration of transportation systems.

3. To investigate and develop an algorithm for plant layout integrated with common aisle structures.

1.5 Outline of the Thesis

The remainder of this thesis contains the following chapters:

Chapter 2 contains a review of the methodologies of the design of manufacturing systems, including the presentation of the problem and methods for solving cell formation and layout problems.

Chapter 3 presents a heuristic for cell formation, which does not need to pre-specify the number of cells and the limitation of the number of machines within a cell and which clusters machine cells in a more natural way.

Chapter 4 describes a genetic algorithm method to design the within-cell layouts in some given layout patterns. These patterns include single row, double row and loop layouts with consideration of the direction of the material flow.

Chapter 5 proposes a method for the hard block layout problems in conjunction with aisle structure. The method designs a layout and an aisle structure simultaneously. In the second stage of the method, the aisle structure is optimised by removing the surplus aisles.

Chapter 6 draws conclusions from this research, and suggests future work in this area and related fields.

The simulation data in this research are provided in Appendix.

Chapter Two

Literature Review

Previous work on the design of cellular manufacturing systems is reviewed in this chapter. The review is focused on the problem representations and the methodologies used to solve the problems. The shortcomings of the existing representations of the problems and methods available for solving the problems are also analysed in this chapter. An introduction to the methodology of the current research is outlined in the conclusions of the chapter.

2.1 Introduction

Variations of the design problem for flexible manufacturing systems have been proposed, implemented and evaluated for over thirty years (Reisman *et al* 1997, Meller and Gau 1996a). Due to its complexity, the problem is generally divided into sub-problems as described in the previous chapter and the design procedure follows accordingly the sequences of the sub-problems. Cell formation is a pre-layout design stage to provide the configuration of a system. Layout planning implements the skeleton of the system and material handling system selection provides further details of the system. Arvindh and Irani (1994) analysed the issue of the interaction of the cell formation and layout problems in the design for a cellular manufacturing system and addressed the need for an integrated solution. Ioannou and Minis (1998) reported the recent research on the integration of the sub-problems in manufacturing shop design. The following sections will review the related work of previous researchers.

2.2 Cell Formation and Group Technology

Group technology was first proposed by R. E. Flanders (1925). The publication of Mitrofanov's book "The Scientific Principles of Group Technology" (Mitrofanov 1958) in Russia, which was translated into English in 1966, generated wide interest in the subject. Burbidge developed methods suitable for hand computation in 1971. Burbidge, Gallagher and Knight, and others propagated the technology (Burbidge 1963, 1971a, 1971b, 1974, and 1975, Gallagher and Knight 1973). To date the problem has been developed in several different mathematical forms, but the optimisation of the cell formation problem, in general, is NP-complete (Cheng, *et al*, 1996a), which means that the problem is hard to solve for a computer.

NP-complete (Garey and Johnson 1979, Poole *et al* 1998) is a terminology in algorithm analysis (Sedgewick and Flajolet 1996), which studies the efficiency of an algorithm. Problems in algorithm analysis can be divided into two classes: P problems and NP problems. The class of P problems contains those that can be solved in polynomial time. The problems in the NP (Non-deterministic Polynomial) class are those that could be solved in polynomial time by an algorithm that is not known so far, however the fact that they cannot be solved within polynomial time has not been proved. Polynomial time transfers can convert some of the problems in the NP class to another form in the NP class. Thus, they are equally complex. If one of them can be solved in polynomial time, they all can. These problems are known as NP-complete. A problem is called NP-hard if it is at least as hard as an NP-complete problem. The details refer to the book by Garey and Johnson (1979). The term "NP-complete" will be frequently used in this chapter, because most of the decision problems involved in the current research are either NP-complete or NP-hard.

The basic principle of group technology (GT) is that the parts to be produced and the machines required to manufacture the parts are divided into families and cells respectively. Group technology identifies the part families and the machine clusters according to the processing requirements of the parts. Typically, the processing requirements are represented by using a part-machine relationship chart. Table 2-1 is an example of such a chart that represents the parts and their incidental machines. Table 2-2 gives an example of grouped families and clustered cells for the problem in Table 2-1.

	Machines							
Parts	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
P_1	1			1			1	
P_2	1	1		1			1	1
P_3		1				1		1
P_4			1		1	1		
P_5	1			1				1
P_6		1						1
P_7			1		1		1	
P_8	1	1		1				
P_9					1	1		
P_{10}		1		1				1

Table 2-1 A part-machine incidence chart

Part	cell 1			cell 2		cell 3		
Families	M_1	M_4	M_7	M_2	M_8	M_6	M_3	M_5
1 P_1	1	1	1					
P_5	1	1	1					
P_8	1	1		1				
P_2	1	1	1	1	1			
2 P_6				1	1			
P_{10}		1		1	1			
P_3				1	1	1		
3 P_4						1	1	1
P_7			1				1	1
P_9						1		1

Table 2-2 Machine cells and part families

Ideally, a part family is processed through its entire operations in a single machine cell, called an independent cell. In practice, however, some parts have operations on machines outside their parent cell. In this case, independent cells can still be obtained by duplicating the machines that process the exceptions (Vakharia and Wemmerlov 1990), but it is sometimes found that too much investment is involved for such an arrangement to be accepted. Moreover, the problem with duplicated machines is that they may involve cycle time and set-up time for every operation of each part, machine assignment, and scheduling (Kern and Wei 1991). Therefore, if machine duplication for different cells is taken into consideration, the problem is not only an identification of part families and machine cells, but also trade-offs among complex problems, such as the costs of intercell flow, the investment for the duplicated machines, utilisation of facilities, WIP and lead time. This topic will be further developed in the following sub-sections.

With regard to the concept of group technology applied to manufacturing, the part grouping and cell formation problem can be classified into two categories. The first category is on the basis of the concept of grouping the similar parts together to form a part family. In this category a part-machine incidence matrix as shown in Table 2-1 is often used for representation of the part-machine relationship. The second category directly aims at the manufacturing cost by analysis of the material flow. The representation is based upon the operation sequences of each part, i.e. the material flow. The representations and methods in the two categories are detailed in the following sub-sections.

2.2.1 Incidence-based Representation

The cell design method using incidence-based representation focuses on the grouping of parts that have similar processing requirements to form part families and the clustering of machines that meet these requirements into machine cells (Salvendy 1982). In these methods, the relationship between parts and machines can typically be presented as a part-machine incidence matrix that uses 1 to indicate that the part visits the machine and leaves a blank or zero if the machine is not visited as shown in Table 2-1.

In this representation method, the parts and the machine are permuted in the matrix, so that a block diagonal is formed. A block diagonal is a matrix of the same dimension in which the diagonal sub-matrices contain the maximum number of '1s'. The boundaries of the diagonal sub-matrices are set by consideration of the exceptions. Exceptions are locations which contain a '1' that lies outside the diagonal sub-matrices. Fewer exceptions mean a better solution. Table 2-2 gives three cells with five exceptions. To increase accuracy, some weight values, which include volumes, weight and dimensions of the parts, etc., can be associated with the exceptions.

One of the first mathematical models for the cell formation problem was proposed by Kumar *et al* (1986) to maximise the incidences '1s' in the diagonal sub-matrices. That is,

$$\text{Maximise } \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{q} a_{ij}x_{ik}y_{jk} \qquad (2\text{-}1)$$

where m is the number of machines.

n is the number of parts.

q is the number of cells to be formed.

a_{ij} is an element of the incidence matrix A.

x_{ik}, y_{jk} are design variables for location of machines and parts into cells respectively.

$x_{ik}, y_{jk} \in \{0, 1\} \qquad i \in \{1,\cdots,m\},\ j \in \{1,\cdots,n\},\ k \in \{1,\cdots,q\}$

subject to

$$\sum_{k=1}^{q} x_{ik} = 1 \qquad i \in \{1,\cdots,m\} \tag{2-2}$$

$$\sum_{k=1}^{q} y_{jk} = 1 \qquad j \in \{1,\cdots,n\} \quad . \tag{2-3}$$

Constraint set (2-2) and set (2-3) express that each machine and each part must be assigned to exactly one cell. The above model can be further constrained by limitation of the number of parts and machines in each cell:

$$L_m \le \sum_{i=1}^{m} x_{ik} \le U_m \quad \text{and} \quad L_p \le \sum_{j=1}^{n} x_{jk} \le U_p \tag{2-4}$$

where L_m and L_p are the lower bounds for machines and cells respectively.

U_m and U_p are the upper bounds for machines and cells respectively.

The above model, which is a quadratic integer programming formulation, is not easily solved by mathematical programming techniques. Kumar *et al* (1986) transformed the model into a linear expression. Boctor (1991) proposed a linear formulation by minimising the number of exceptions. Kusiak and Heragu (1987a), Ventura *et al* (1990), Crama and Oosten (1996), Shanker and Agrawal (1997) also gave the alternative linear models. Although linearised, these models are still NP-complete (Kumar *et al* 1986, Shanker and Agrawal 1997). Therefore, other optimisation techniques, such as heuristics, have to be used rather than traditional programming approaches.

Another type of model for part-grouping problems is p-median formulation. As first suggested by Mulvey and Crowder in 1979 (Benarieh and Chang 1994), the p-median algorithm was adopted by Kusiak (1985, 1987a) and applied to group technology. A

p-median model identifies p groups out of n parts. It fixes p as nucleus parts around which the remaining parts are grouped by assigning them to the nucleus part with which it has maximum similarity. That is,

$$\text{Maximise } \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} \tag{2-5}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i \in \{1,\cdots,n\} \tag{2-6}$$

$$\sum_{j=1}^{n} x_{jj} = p \tag{2-7}$$

$$x_{ij} \le x_{jj} \qquad i \in \{1,\cdots,n\}, j \in \{1,\cdots,n\} \tag{2-8}$$

$$x_{ij} \in \{0,1\} \qquad i \in \{1,\cdots,n\}, j \in \{1,\cdots,n\} \tag{2-9}$$

where n is the number of parts.

p is the number of part groups.

s_{ij} is the value of similarity between part i and part j.

Constraint set (2-6) and (2-9) ensure that each part is assigned to exactly one part family. Constraint (2-7) specifies that p parts will be chosen as the nuclei for creating the families and constraint set (2-8) ensures that the following assignment of a nucleus is to only one of p part families. The p-median problem is also NP-complete (Parker and Rardin 1988). Deutsch *et al* (1998) presented an improved p-median model and the vertex substitution heuristic (Teitz and Bart 1968) was adapted to solve the problem.

The similarity between a pair of objects is defined by a similarity coefficient. A variety of similarity coefficients or dissimilarity coefficients have been proposed to measure closeness of similarity among machines and parts. McAuley (1972) attempted first to use cluster analysis for the formation of machine cells by adapting Jacard's similarity coefficient. McAuley defined the measure of closeness between machines as the following equation.

$$s_{ij} = \frac{a}{a+b+c} \tag{2-10}$$

where s_{ij} is the similarity between machine *i* and *j*.

a is the number of parts processed on both machines.

b is the number of parts processed on machine *i* only.

c is the number of parts processed on machine *j* only.

Later, McAuley's similarity coefficient was re-expressed by Seifoddini and Wolfe (1986, Seifoddini 1990) in a general formulation in which the production volume is considered. Seifoddini and Djassemi (1991) reported that the sum of intercell and intracell flow is the most effective measure and proposed a data-based coefficient. Many other forms of similarity coefficients or dissimilarity measures, which are not included in this thesis, can be found in the reviews by Shafer and Rogers (1993a, 1993b), Sarker (1996), Mosier *et al* (1997). Most of them are extensions of Jacard's coefficient. These similarity coefficients are not only used in manufacturing for the machine-part grouping but also utilised in other field of science, such as medicine, biology, ecology, psychology, geology and artificial intelligence. Though many coefficients are available, it has not been conclusively proved that a particular similarity coefficient is efficient in all situations (Sarker 1996). Some of the similarity coefficients have been evaluated and compared (Mosier 1989, Taboun *et al* 1991, Seifoddoni 1990, Shafer and Rogers 1993b, Wang and Roze 1995, Vakharia and Wemmerlov 1995). The comparisons provided guidelines for choosing the appropriate similarity coefficient in an application. There are various measures for the evaluation and comparison of the goodness of cell formation solutions. The most common measures can be found in the papers of Sarker (1996), Nair and Narendran (1998) and a recent review by Sarker and Mondal (1999).

The main shortcoming of incidence-based representation is that it neglects processing sequences, so that the representation is too crude to measure a realistic situation of material flow (Dahel 1995). Different process sequences could cause quite different results in material flow volumes. The previous example, in Table 2-2, is again taken into account. If the process sequence of part 2 (P2) is M1 → M4 → M7 → M8 → M2, the intercell flow volume of part 2 is simply the batch size from cell 1 to cell 2 (Figure 2-1). In contrast, if the process sequence is M1 → M2 → M4 → M8 → M7, the intercell flow volume of part 2 is four times the batch size (Figure 2-2), while the exceptions remain the same. Other sequences for part 2 could lead the intercell flows from one to four times the lot size. Therefore, the variance of operation sequences could cause a significant difference with regard to the material handling cost.

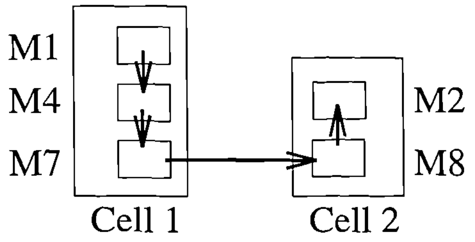


Figure 2-1 No-backtracking flow

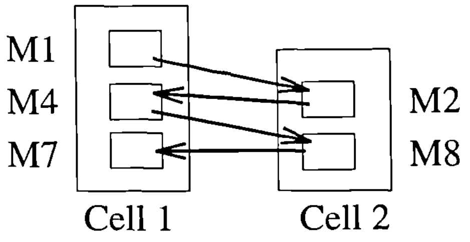


Figure 2-2 Multi-backtracking flow

Process sequences impact not only on material flow but also on work-in-progress (WIP), lead time, etc. By way of example, the transportation of intercell flow is commonly on pallets. As in the previous analysis, if one pallet is used for the intercell transportation of parts in Figure 2-1, four pallets are required for the intercell transportation in Figure 2-2. More pallets being used in the system imply more parts which are waiting for transportation and thus imply larger WIP. Thus, the WIP of part 2 in Figure 2-2 may be four times that in Figure 2-1.

In general, process sequences mainly influence the following aspects in a manufacturing system:

- Intercell traffic
- Work-in-progress
- Number of buffers and space for buffers
- Lead time
- Complexity of scheduling
- Complexity of management

Therefore, processing sequences have a substantial influence on design of manufacturing systems and should not be ignored. Selvam and Balasubramanian (1985), and Balasubramanian and Panneerselvam (1993) developed a heuristic to arrange cells, derived from an incidence matrix, by reconsidering excess handling effort arising from component routing. Tam (1990) proposed an operation based similarity coefficient for part family formation, though the coefficient only includes the sequence order rather than intercell flow volume. Chow and Hawaleshka (1993) emphasised that the inter-cellular movement of part types should be minimised rather than reducing the number of exceptions. Ho and Moodie (1996) developed a method based on an operation similarity coefficient that allows flexible processing and routing. Nair and Narendran (1998) presented a weighted machine sequence similarity coefficient by using ordinal data in the incidence matrix. In conclusion, incidence-based representation cannot provide enough information for including operation sequences and an alternative is flow-based representation.

2.2.2 Flow-based Representation

In the last decade, more researchers have realised the importance of part process sequence in the design of cellular manufacturing systems (Logendran 1991a and Dahel 1995). Logendran (1991a, 1991b, and 1992) built a model for cell formation and cell layout based upon the sequence of operations by evaluating the intercell and intracell moves. In this approach total moves are computed as a weighted sum of both intercell and intracell moves. Selim *et al* (1998) reviewed the previous models and presented a general model which includes investment for machines, material handling cost, utilisation of machines, tooling cost and operator training cost in the objective function presented as follows.

$$\text{Minimise } \sum_{i=1}^{m}\sum_{k=1}^{q} V_i \xi_{ik} + H\sum_{j=1}^{n}\sum_{l=1}^{\omega}\sum_{k=1}^{q} D_j y_{ljk} + \sum_{i-1}^{m} G_i \sum_{k=1}^{q} \xi_{ik} \sum_{j=1}^{n}\sum_{l\ 1}^{\omega} D_j C_{lji} x_{ljk}$$

$$+\sum_{r=1}^{t} T_r \sum_{k=1}^{q} z_{rk} + \sum_{i=1}^{m} R_i \sum_{k=1}^{q} \zeta_{ik} \qquad (2\text{-}11)$$

where n is the number of part types.

m is the number of machines.

q is the number of cells.

ω is the number of operation types.

t is the number of tool types.

w is the number of workers.

V_i is the amortised cost per period to procure a machine of type i.

H is the handling cost per intercell move.

D_j is the average demand for part j (in units) per period.

G_i is the variable cost per unit time to operate a machine of type i.

C_{lji} is the processing time to perform operation l of part j on machine type i.

T_t is the cost per period to stock tool type t.

R_i is the cost to train a worker to operate a machine of type i.

Decision variables:

$$x_{ljk} = \begin{cases} 1, & \text{if operation } l \text{ of part } j \text{ is assigned to cell } k; \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ljk} = \begin{cases} 1, & \text{if operation } l \text{ of part } j \text{ is performed in a different cell } k \text{ than the} \\ & \quad \text{preceding operation;} \\ 0, & \text{otherwise.} \end{cases}$$

$$z_{rk} = \begin{cases} 1, & \text{if tool } r \text{ is used in cell } k; \\ 0, & \text{otherwise.} \end{cases}$$

ζ_{ik} is the number of workers trained to operate machine type i in cell k.

ξ_{ik} is the number of machines of type i assigned to cell k.

ψ_{isk} is the amount of time of worker s assigned to machine type i in cell k.

Although it seems more accurate than previous models, it has so many decision variables involved that the model has not yet been used in any application. Moreover, some costs in the model are not correctly defined. Firstly, the trips used for computing the material handling cost should be based upon the transportation batch size of each part type instead of one trip per part as assumed in the model. Secondly, the total training cost of the work force should not be simply added to the cost function of a certain period of time, which is normally on an annual basis.

Nagi *et al* (1990) suggested that the sequence of machines visited by a part be recorded in the incidence matrix by using the ordinal number of an operation. However, if the same machine is visited for two or more non-consecutive operations, then the representation by the ordinal incidence matrix is not adequate (Heragu 1994). Heragu and Kakuturi (1997) consequently proposed to use dummy columns when the same machine is visited more than once. Some other studies use a machine-machine chart and concentrate on the flow density between machines. The total intercell flow can be expressed by the following formulation (Sofianopoulou 1997).

$$\sum_{i=1}^{m-1}\sum_{j=1+1}^{m} f_{ij}(1-x_{ij}) \qquad (2\text{-}12)$$

where m is the number of machines.

f_{ij} is the flow density between machine i and machine j.

$$x_{ij}=\begin{cases}1, & \text{if machine } i \text{ and } j \text{ are assigned to the same cell,}\\ 0, & \text{otherwise.}\end{cases}$$

The formulation above contains fewer loops than that of Selim *et al* (1998). Thus, it has the advantage of reduced computational time.

2.2.3 Specific Methods For Cell Formation Problems

The methods described in this sub-section are mainly the specific approaches for part grouping and cell formation problems. Some general methodologies for solving optimisation problems will be presented in Section 2.5.

2.2.3.1 Heuristics

"Heuristics stand for strategies using readily accessible though loosely applicable information to control problem-solving processes in human beings and machines." (Pearl 1984) Heuristics, in more popular understanding, are rules of thumb, educated guesses, intuitive judgements or simply common sense ideas to solve a particular problem. A heuristic improves the efficiency of a search process, possibly by sacrificing claims of completeness (Rich and Knight 1991). According to the above definition, some common used techniques, such as expert systems, simulated annealing (SA), tabu search (TS) and genetic algorithms (GAs), also belong to the catalogue of heuristics. However, the heuristics in the current section are restricted to those of rule-based algorithms, in which specific rules are used to solve cell formation problems, in order to distinguish the some general methods, which will be discussed later in Section 2.5.

There are various heuristics that have been developed for cell formation in the past three decades. Notably among these algorithms are those reported by Sarker and Xu (1998) and compared by Miltenburg and Zhang (1991). The following approaches are relevant to the research in this thesis.

Vannelli and Kumar (1986, Kumar and Vannelli 1987) developed a incidence-based heuristic which was evolved and extended by Wei (1987), Wei and Gaither (1990). The heuristic starts with a seed machine in each cell and subsequently adds up to one part to each cell along with all the machines required by that part, so that the increase in the exceptions is minimised. Frazier and Gaither (1991) studied the problem of seed selection. Ballakur and Steudel (1987) suggested a clustering algorithm based upon within-cell utilisation. Chandrasekharan and Rajagopalan (1986, 1987) developed an algorithm by selecting seed machines arbitrarily.

Harhalakis *et al.* (1990) presented a two-phase heuristic algorithm to minimise intercell movements. The heuristic starts with each machine as a cell, then aggregates the two cells with the largest flow between the cells and repeats the aggregation until the required number of cells is achieved. Finally the solution is improved by trying to reassign each machine to another cell. Lee and Chen (1997) extended the method of Harhalakis by balancing workload among duplicated machines. Nevertheless, Harhalakis' twofold heuristic sometimes might not be good enough to cluster the machines between which there are many small flows. Okogbaa *et al* (1992) developed an intercell flow reduction (INCFR) heuristic based on operation sequence and inter-machine flow (IMF). The cell formation procedure starts with a number of representative machines or seeds of the cells. These machines are selected to have the minimum inter-machine flow. IMFs between the pairs of each unassigned machine and the seed or the assigned machines are calculated. Okogbaa defined the terms "safety combination rate" (SCR) and "cell safety combination rate" (CSCR). Unassigned machines with an SCR higher than a specified threshold are merged into the relevant cells. For the remaining machines, the cell-machine flow is computed and the machines with a CSCR higher than the threshold are merged into the relevant cells. Finally, any remaining machines are assigned to the cells with the largest CSCR value. The solution by this method highly depends on the threshold given. Verma and Ding (1995) developed a heuristic method, considering the cost of intercell movement, as well as intracell skipping cost and backtracking cost. Akturk and Balkose (1996) proposed a cluster analysis heuristic that considers both design characters, manufacturing attributes and operation sequences simultaneously by including material flow problems such as backtracking and skipping (Akturk 1996). The method evaluates solutions by considering the investment, the loading balance and the skippings. However, the operation sequences were taken into account only within the cells. Consequently, the multi-step heuristic of Akturk and Balkose could not minimise the backtracking between cells.

2.2.3.2 Graph Theoretic Methods

The relationships among the machines or the components to be manufactured can be represented by a weighted graph where a node represents a machine or component and an edge represents association between the nodes (Amirahmadi and Choobineh 1992). In a p-median problem, each node in the graph represents a component and the weight of an edge will be the similarity between the components. The problem can be converted into that of finding the minimum cut-set, which partitions the graph into p sub-graphs. In a flow-based model, each node represents a machine and the weight of an edge will be the flow density between the machines. Rajagopalan and Batra (1975) developed a graph-partitioning algorithm that utilises cliques to group the machines. Lee and Garcia-Diaz (1993 and 1996) presented network flow procedures for minimising inter-cellular part moves. Ng (1993) used a minimum spanning tree to solve the problem. Wu and Salvendy (1993) used a cut-tree method to solve the cell formation problem modelled by using an undirected graph and a simplified algorithm was developed to reduce the computational effort. Later, Wu (1998) extended the method of Wu and Salvendy by considering duplicated machines.

2.3 Facility Layout

Developing a facility layout is an important step in the design of a manufacturing system because of its high impact on material handling cost and time, on throughput time of the products, and on productivity of the facilities. However, the facility layout problem is more complicated than the cell formation problem, because more parameters, such as distance between the facilities and dimensions of the facilities, should be taken into consideration. A poor layout may also negate some of the flexibility of the system. Moreover, modifications in the layout of modern facilities could be costly. Therefore, facility layout in a manufacturing system must be properly developed in order to reduce the manufacturing cost and should avoid frequent modifications. Hundreds of papers about facility layout problems have been published in the last two decades. A variety of representations and methods were utilised for layout problems. Kusiak and Heragu (1987b), Hassan (1995), Welgama and Gibson (1995a), and Balakrishnan and Chen (1998) presented surveys of the layout problem and its mathematical models, including quadratic assignment models, set covering models, integer and mixed integer programming problems, non-linear programming models, graph-theory-based models and hybrid models. Sirinaovakul and Thajchaypong (1996) presented an analysis of the techniques used for layout problems.

2.3.1 Representations of the Layout Problem

Representations of layout problems can be classified according to their objective or the geometrical appearances of the layout and its departments. Depending upon the objective, evaluation of layout problems is divided into two categories: quantitative criteria and qualitative criteria. Quantitative criteria can be expressed by measurable terms such as material handling cost and travel distance of operators. Qualitative criteria, which are much fuzzier, are usually expressed by the desirable closeness between two machines.

2.3.1.1 Qualitative Criteria Problems

Qualitative criteria use quality terms to specify the relationships between departments. In 1962, Muther and Wheeler described the closeness relationships as "absolutely necessary", "especially important", "important", "ordinary closeness", "unimportant" and "undesirable". These terms are still used commonly. For the ease of the evaluation of layout, the terms are quantified to numerical ratings. An example of numerical ratings by Harmonosky and Tothero (1992) is given in Figure 2-3. In the model of Houshyar and White (1997), material flow is directly used as the quantified qualitative criterion. The objective in a quantitative model is to maximise the total closeness as follows.

Numerical Rating	Symbol Rating	Relationship Definition
4	A	Absolutely necessary
3	E	Especially important
2	I	Important
1	U	Unimportant
-1	X	Undesirable

Figure 2-3 Quantified qualitative criteria

$$\text{Maximise} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{r} \qquad (2\text{-}13)$$

where n is the number of machines to be located.

i, j are the index to the machines.

c_{ij}^{r} is the numerical value of the closeness rating between machine i and machine j.

Quality criteria are good, especially when there is no meaningful numerical expression of the closeness relationships between the departments. For instance, accessories of a facility are necessarily near to the facility; the press tools on blanking for car bodies should be put close to the press machine and it is forbidden to put the welding area and the painting area together. In some applications, hybrid criteria were used to describe both quantitative and qualitative relationships by quantifying qualitative criteria.

2.3.1.2 Quadratic Assignment Problems

A quadratic assignment problem (QAP) was first modelled by Koopman and Beckman (Kusiak and Heragu 1987b). Unlike the quality criteria problem, transportation costs between facilities are taken into account in a QAP. In the quadratic assignment model, plant area is gridded into n equal blocks corresponding to n machines to be located. Each machine occupies exactly one block. The model is to minimise the total material handling cost and the design variables in the model are expressed in 0-1 integral forms. This is represented in mathematical terms as follows.

$$\text{Minimise} \sum_{j=1}^{n} x_{ij} = 1 \tag{2-14}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, 2, 3, ..., \text{n} \tag{2-15}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, 2, 3, ..., \text{n} \tag{2-16}$$

where n is the total number of machines or locations.

f_{ik} is the volume of the material flow from machine i to machine k.

c_{jl} is the cost of transporting unit material from location j to location l.

$$x_{ij} = \begin{cases} 1, & \text{if machine i is at location j.} \\ 0, & \text{otherwise.} \end{cases}$$

The term $f_{ik}c_{jl}x_{ij}x_{kl}$ in the above formulation calculates the cost of material flow from machine i to machine k, provided that the machine i is placed in block j and machine k in block l. The summations of the terms with respect to i, j, k and l add up the total cost of material flow between every machine in every location. The constraints ensure each machine occupies exactly one block.

If there are more locations than machines, the problem becomes a "quadratic set covering problem" (QSP) as given by Bazaraa (Bazaraa 1975, Kusiak and Heragu 1987b). In a QSP, constraints are needed to ensure that each facility is assigned to one location and that each block is occupied by, at most, one facility.

It has been proved that quadratic assignment problems or quadratic set covering problems are NP-complete (Sahni and Gozalez 1976, Garey and Johnson 1979). Only problem instances with up to 17 facilities can be solved within reasonable computation time, which varies linearly as the factorial of the number of facilities.

2.3.1.3 Distance-based Models

In most real word problems of manufacturing layout, the machines or departments occupy different sizes of areas, and the dimensions of machines play an important role in the evaluation of the layouts. In the above cases, a quadratic model is not sufficient for layout design. This leads to the study of layout problems with unequal-area departments and models which involve distance measure.

In distance-based models, the total material handling cost is the most commonly used as the objective to measure the efficiency of a layout (Kusiak and Heragu 1987b). The total material handling cost is calculated as the sum of the flow volume times the distance travelled, as in the following equation.

$$\text{Minimise} \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}c_{ij}d_{ij} \qquad (2\text{-}17)$$

where n is the number of machines to be located.

i, j are index to the machines

f_{ij} is the volume of the material flow from machine i to machine j.

c_{ij} is the cost of transporting unit material from machine i to machine j.

d_{ij} is the distance between machine i and machine j.

When the positions of the centres of the machines are located in a Cartèsian co-ordinate system and if the departments are supposed to be placed in rectangular shapes, the following constrains are used for preventing the departments being overlapped or out of the border in a rectangular shop.

$$|x_i - x_j| \geq \frac{1}{2}\cdot(h_i + h_j) \qquad i, j = 1, 2, ..., \text{n}; \quad i \neq j \qquad (2\text{-}18)$$

$$|y_i - y_j| \geq \frac{1}{2}\cdot(v_i + v_j) \qquad i, j = 1, 2, ..., \text{n}; \quad i \neq j \qquad (2\text{-}19)$$

$$b_{xsub} + \frac{h_i}{2} \leq x_i \leq b_{x\sup} - \frac{h_i}{2} \qquad (2\text{-}20)$$

$$b_{ysub} + \frac{v_i}{2} \leq y_i \leq b_{yup} - \frac{v_i}{2} \qquad (2\text{-}21)$$

where x_i, y_i are the co-ordinates of the centre of machine i.

h_i is the length of the horizontal side of the machine i.

v_i is the length of the vertical side of the machine i.

b_{xsub} is the lower border of the vertical side of the plant area.

b_{xup} is the upper border of the vertical side of the plant area.

b_{ysub} is the lower border of the horizontal side of the plant area.

b_{yup} is the upper border of the horizontal side of the plant area.

The distance expressed in the above equation has appeared in literature in various forms. The commonly used distance measures are Euclidean distance, squared Euclidean distance and rectilinear distance. They are formulated in the following equations.

Euclidean distance:

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}} \tag{2-22}$$

Squared Euclidean distance:

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{2-23}$$

Rectilinear distance:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{2-24}$$

Euclidean distance or squared Euclidean distance is used to ease the computation for analytical optimisation methods, such as the Lagrangian method (Tam and Li 1991). When rectilinear distance is utilised, the model becomes a mixed integer programming problem, as formulated by Kaufman (Kusiak and Heragu 1987b). Rectilinear distance is the most commonly used measure, because most material handling systems in planar layouts are in a rectilinear arrangement. Thus, rectilinear distance is the closest among the above three to the distance travelled (a layout other than that of a manufacturing system could have a different type of model, see Domschke and Krispin 1997).

Bozer and Meller (1997) re-examined the present distance measures and found that if the flow between two departments is distributed to sub-departments, the centroid to centroid (CTC) distance is not necessarily equal to the distributed centroid to centroid (DCTC) distance. A layout that is optimal under DCTC distance is hereby not necessarily optimal under CTC distance. Welgama and Gibson (1993) calculated the distance between a pick-up point and a drop-off point, although the measure was still rectilinear. Benson and Foote (1997) applied the aisle distance between a pick-up point and a drop-off point to the computation of the material handling cost.

Aisle distance differs from the rectilinear distance in two ways. First, it is sometimes impossible to build a rectilinear aisle between two department doors. For instance, if there is another department between the two doors, the materials have to make a detour around the department as shown in Figure 2-4. Second, it may not be necessary to build rectilinear aisles for all appropriate pairs of department doors, especially for those with less material flows, although the distance may be longer (Figure 2-5). Therefore, an aisle structure not only is an integration of a layout and material handling system, but also plays a substantial role in the system evaluation. It is appropriate that aisle structure is taken into consideration during the layout design stage and that the real distance travelled along aisles is used as the distance measure. Although no simple equation exists in general to represent aisle distance, the implementation of the aisle distance calculation by a computer-based numerical method is not difficult.

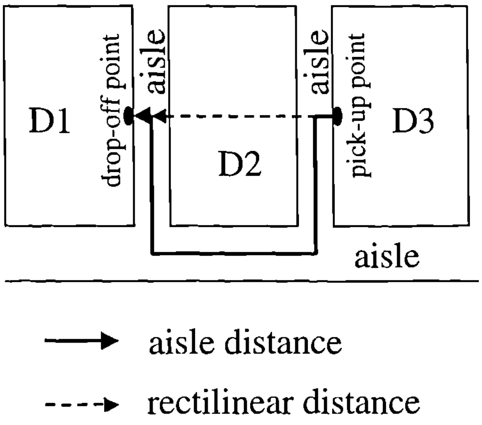


Figure 2-4 Aisle distance and rectilinear distance

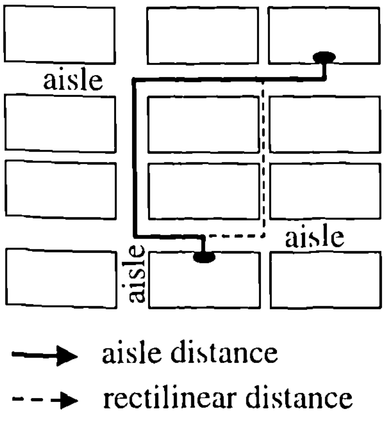


Figure 2-5 An aisle may not be necessary

2.3.1.4 Multiple Criteria Models

In most hybrid models, the quantity criteria and the quantified quality criteria are combined in a weighted objective (Rosenblatt 1979, Dutta ans Sahu 1982, Rosenbalatt and Sinuany-Satern 1986, Houshyar 1991, Savsar 1991, Harmonosky and Tothero

1992, Meller and Gau 1996b). Rosenbalatt and Sinuany-Satern, Meller and Gau studied the weight parameter and proposed methods for robust layouts. Cambron and Evans (1991) represented the problem in a contrary way. They converted the quantitative flows into qualitative criteria and then re-evaluated the hybrid closeness terms.

Some applications used more objectives, such as flexibility, work-in-progress, machine utilisation and other managerial measures (Afentakis *et al* 1990, Savsar 1991, Massay *et al* 1995, Fu and Kaku 1997, Farrington and Nazemetz 1998), for the evaluation of the layouts but these measures are only computable by simulation. Simulation of the running of a system is time-consuming in order to get reasonably accurate results, so simulation is mainly used in system comparison.

2.3.2 Specific Methodologies for Layout Problems

The methods described in this sub-section are mainly the specific approaches for layout problems, although some of them are also used for cell formation problems. Kusiak and Heragu (1987b), Welgama and Gibson (1995a) gave surveys of algorithms for facility layout. Some general methodologies for both layout problems and cell formation problems will be presented in Section 2.5.

2.3.2.1 Improvement Algorithms

Improvement algorithms, as its name implies, are based on an initial layout and the following improvement of the layout by rearranging the facilities. Among the present improvement algorithms, the best known and the earliest procedure is CRAFT. CRAFT (Computerised Relative Allocation of Facilities) was developed by Armour and Buffa (1963) for quadratic assignment problems and was subsequently improved by several other writers (Nugent *et al* 1968, Allenbach and Werner 1990, Malmborg 1994). The CRAFT software creates new layouts by exchanging machines in pairs and then evaluates them. When all pairs of exchanges have been completed, the exchange with the best evaluation is selected and becomes the new initial layout. This cycle repeats until no better exchange can be found. As to unequal-area problems, CRAFT exchanges facilities in three patterns; facilities with the same area, common borders and a fixed outline of the layout area. Bozer *et al* (1994) used the space-filling curve to allow unequal-area facility exchange but the shapes of the departments are limited to the previous given patterns of the space-filling curves.

2.3.2.2 Construction Algorithms and Hybrid Algorithms

Unlike improvement algorithms, construction algorithms start with a blank shop floor and then add machines into it. Seehof and Evans (1967) developed a construction programme called ALDEP (Automated Layout Design Program) to solve qualitative-criteria layout problems. The ALDEP package begins by randomly selecting a machine and placing it in the layout. From then, the machines are divided into two groups. One group is included in the layout, the other outside. The program scans all the machines in the second group and puts the machine with the highest closeness to the first group into the layout until all the machines are put in place. Many layout plans are generated by iteration of the entire process with many different random beginning selections. The best layout is then selected (Schroeder 1993). Most construction algorithms, such as CORELP (Parsaei *et al* 1987, Ziai and Sule 1988), SHAPE (Hassan *et al* 1986), etc., have a similar procedure. The differences are mainly in the four ways listed below:

- How to select the next machine to put into the layout.
- How to evaluate the relationship between the machines already located and the selected machine.
- How to score each machine in the layout.
- How to represent the layout.

Once a machine has been put in the layout, it cannot be moved to another location in construction algorithms. Thus, it is difficult to obtain even a local optimum solution. To overcome this shortcoming, a construction algorithm is used for creating an initial layout, and then an improvement algorithm is applied. This idea leads to hybrid algorithms.

A hybrid algorithm establishes an initial layout by a construction approach and ameliorates the layout by an improvement approach. Therefore, hybrid algorithms have the merits of both a construction approach and an improvement approach. Construction algorithms could give a good starting solution and improvement algorithms could find, at least, a local optimum (Welgama and Gibson 1995a). A hybrid algorithm has quite good performance for solving a quadratic assignment problem or a quadratic set covering problem, when applying some techniques, such as simulated annealing and/or tabu search (see Section 2.6.1 and 2.6.2). However, the swap-based method may have difficulties when the areas of the departments are unequal. Other techniques are needed to deal with an unequal-area problem.

2.3.2.3 Floorplan Representations and Methods

A floorplan problem is that of assigning departments into rectangular blocks in a rectangular-shaped plant. A floorplan is sliceable if it is constructed by recursively partitioning a rectangular block. Lauther (1980) and Otten (1982 and 1983) first introduced the floorplan problem for very large scale integration (VLSI) design and Stockmeyer (1983) developed a polynomial time algorithm for area optimisation. An example of a slicing floorplan with six departments is illustrated in Figure 2-6.

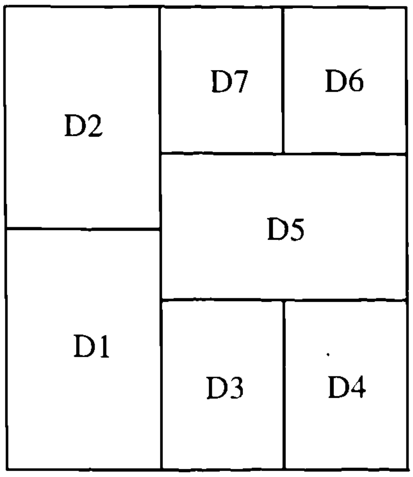


Figure 2-6 A floorplan with six departments

Slicing floorplan problems can be classified into soft block problems and hard block problems. In a soft block problem, the block areas are maintained constant, while their lengths and widths may vary. In a hard block problem, each basic block has a space large enough to accommodate the corresponding department, which is one of a series of the previously defined realisations of the same area. For example, the series of the dimensions of department D2 could be previously defined as one of 2x12, 3x8, 4x6, 6x4 and so on, but excludes the other side dimensions of the same area, which may appear in a soft block approach.

A slicing tree can be used to represent a slicing floorplan (Stockmeyer 1983). A slicing tree is a binary tree representing the recursive partitioning process that generates a slicing floorplan. Each internal node of the tree is labelled either *h* or *v*, specifying whether it is a horizontal or vertical slice. Each external node represents a basic rectangular block. A slicing tree as shown in Figure 2-7, therefore, can represent the slicing floorplan in Figure 2-6. A slicing tree can be expressed by a string, in which each internal node performs as an operator and each basic block represented by a leaf node as an operand. There are three types of string sequences: inorder, preorder and postorder (Horowitz and Sahni 1976). The tree in Figure 2-7 would be represented by the three expressions as follows.

Inorder	(2 h 1) v ((7 v 6) h (5 h (3 v 4)))
Preorder	v h 2 1 h v 7 6 h 5 v 3 4
Postorder	2 1 h 7 6 v 5 3 4 v h h v

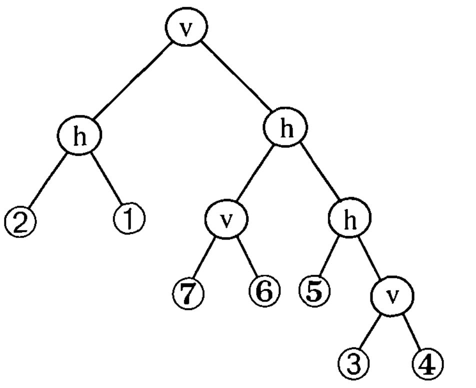


Figure 2-7 A slicing tree of the floorplan

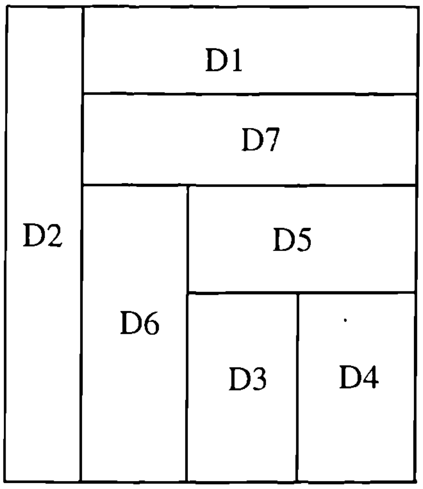


Figure 2-8 Another layout represented by the same string

The above three types of the strings are mixtures of operators and operands, which belong to totally different symbol sets. The mixture of different types of symbols makes these types of representations difficult to be handled by an optimisation algorithm. Nevertheless, it seems impossible to separate a string of any of the expressions into two sub-strings, operator string and operand string, without losing the information which would be required to rebuild the original tree. For instance, v h h v h v and 2 1 7 6 5 3 4 could be the tree of v 2 h 1 h 7 v 6 h 5 v 3 4 in preorder, which gives a different layout as in Figure 2-8 rather than the layout in Figure 2-6.

Tam (1992) tried to separate the operators and operands and keep the operands in a fixed sequence by using four symbols "U", "B", "L", "R" to denote both the direction of a partition cut and the relative position of the blocks, as shown in Figure 2-9. However, the fixed sequence cannot represent the entire set of the combination of the blocks. For example, it is impossible to represent the layout in Figure 2-10 with an operand sequence of 1 2 3 4. This indicates that some possible solutions, which cannot be represented in the given sequence, are excluded, and therefore, the quality of the slicing tree solution by Tam's representation would depend on the operand sequence.

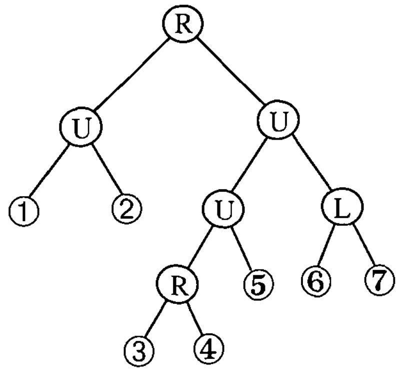


Figure 2-9 A tree representation of Tam

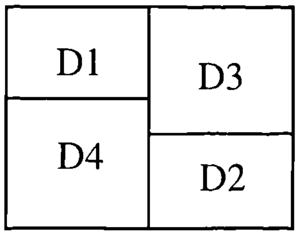


Figure 2-10 A layout which cannot represented by the string of 1 2 3 4

On the other hand, if the string for departments is not fixed and all possible operand sequences are considered, many different operator strings in Tam's method could present the same layout (see Figure 2-11). This implies that the search space is enlarged, which would cause low search efficiency, because much time will be spent on the evaluation of the solutions that present an identical layout. Tam, in his paper, did not explain how to rebuild the slicing tree or the layout from the operator string.

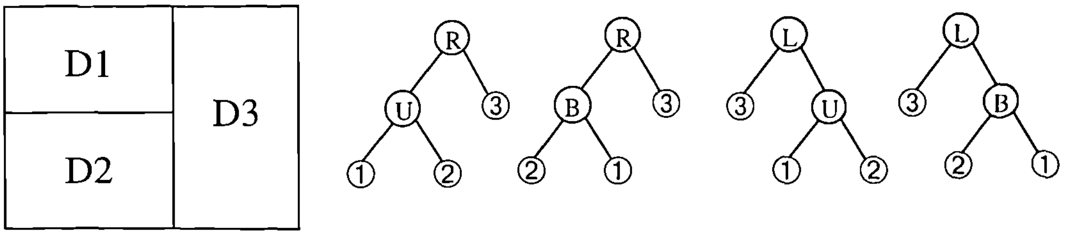


Figure 2-11 The same layout represented by different trees

Tretheway and Foote (1994) developed a heuristic, FAST, for facility layout problems including aisle location. They utilised Drezner's scatter diagram (Drezner 1987) to allocate departments in a slicing floorplan, while preserving department areas instead of equally-shaped departments in Drezner's technique. In their method, the aisle structure is pre-specified, including the number of main aisles and the number of sub-aisles. The sub-aisles are limited to one level (see Chapter 5). In their method, distance is used for calculating the objective of the material handling cost. Drezner began with the rectilinear distance, which was then approximated to a squared Euclidean distance, and finally simplified to an expression of one-dimension, for mathematical ease. In their one-dimensional model, the distances of sub-aisles were ignored. In spite of the above shortcomings, they had a very good concept of design aisle structure based upon slicing cut-lines, because the simple and straight cut-lines provide the aisle structure naturally. Benson and Foote (1997) extended FAST by the optimisation of door locations in each department. In their DoorFAST package, aisle distances between each pair of doors were calculated for the evaluation of the layout, which was derived by the scatter diagram method.

Bay structure, which was first proposed by Tong (1991), is a specific slicing floorplan in which the floor is cut into parallel bays and then each bay is normally cut into rectangular basic blocks with a pre-specified area for each department. Bay structure can be simply represented by a string of departments and a string of break points. Tate and Smith (1995) optimised the unequal-area problem in bay structure by utilising an "aspect ratio" to constrain the shape of blocks. Peters and Yang (1997) integrated bay layout and material handling system design. They proposed a methodology for spine or perimeter layout configuration with consideration of pickup and deposit points.

Hard block models, found in literature so far, were concentrated on the area optimisation of non-slicing floorplans in the design of VLSI circuits. Most of the hard block methods can be used for solving soft block problems, because hard block problems have stricter constraints. Chong and Sahni (1993), Pan *et al* (1996), Rebaudego and Reorda (1996) developed methods for area minimisation, in which wheel blocks (as shown in Figure 2-12), the most simple non-slicing form, are included. Okada *et al* (1998) also presented a shape formation method for area minimisation based upon a slicing floorplan. They used rectilinear soft blocks to minimise the waste area created by the composing of rectangular hard blocks. The method has potential to be used in manufacturing layout because in manufacturing most auxiliary area is flexible in shape. Other non-slicing approaches are described in the following sub-section.

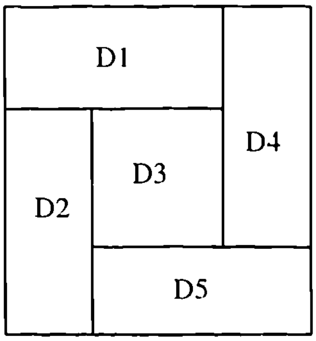


Figure 2-12 A wheel structure

2.3.2.4 Graph-theoretic Methodologies

Graph-theoretic models can be classified into three sub-classes: adjacency graph approaches, maximal spanning tree approaches and polar graph models. Adjacency graph approaches have to solve two sub-problems in order to develop a layout on the basis of facility adjacency requirements (Hassan and Hogg 1989, Meller and Gau 1996b, Irvine and Rinsma-Melchert 1997). Firstly, an adjacency graph is built, based upon a graph of the quantified facility adjacency requirements. The adjacency graph is a planar sub-graph, which determines the neighbour relationships (the dual graph) of the corresponding layout. Secondly, the adjacency graph is converted into a layout.

The first problem in an adjacency graph approach is frequently regarded as that of finding a maximum-weight, maximal planar sub-graph of the adjacency requirement graph (Wascher and Merker 1997). The latter problem is known to be NP-complete (Giffin 1984), so heuristics (Foulds and Robinson 1978, Eades *et al* 1982, Green and Al-Hakim 1985, Leung 1992, Boswell 1992, Merker and Wascher 1997, Cimikowski and Mooney 1997) are necessary to solve problems of practical sizes (Boswell 1992). Wascher and Merker (1997) gave a review and comparison of the heuristics. The material handling cost cannot be calculated precisely due to having not involved any distance measure in the problem.

Converting a maximal planar graph into a block layout is more difficult. Foulds and Robinson (1978) proposed a deltahedron heuristic for finding an orthogonal floorplan, but it is limited to maximal planar adjacency graphs of a restricted type. Hassan and

Hogg (1989) proposed a procedure which assigned each department to an integral number of squares in an grid shop floor. However, according to Al-Hakim (1992), it was found that the method of Hassan and Hogg does not always work. Rinsma *et al* (1990) presented an Orthogonal Division Algorithm, which divides the area required by a facility into two halves and then iteratively selects rectanguloids to obtain the correct area. Irvine and Rinsma-Melchert (1997) improved the algorithm. Nevertheless, the improved algorithm needs the user to select carefully the values of points α and β, which may significantly change the shape of a department in the final layout. Welgama *et al* (1994) used a knowledge-based system to convert the dual graph of a maximal planar graph into a block layout. The knowledge base consists of web-grammar rules which select facilities sequentially, determine facility dimensions and achieve automated identification and reduction of empty spaces. Watson and Giffin (1997) presented an alternative method by splitting vertices rather than dividing an area, but the shape irregularity in the layout makes the algorithm not applicable to real world problems. According to Meller and Gau (1996a), "like the QAP-approach, unequal-area problems of even small size cannot be solved to guaranteed optimality with graph-theoretic approaches".

Instead of a maximal planar graph, maximum spanning tree approaches build a maximum spanning tree based upon flow analysis (Carrie 1975, Heragu and Kusiak 1988, 1990, Heragu 1989). Irani *et al* (1993) proposed a "maximal spanning arborescence" approach by utilising directed trees. The method identifies a flowline layout and shared machines in the flowlines. Gomory and Hu (1961) are the pioneers of cut tree methods. A cut tree method improves the maximal spanning tree approaches by minimising the value of the total material flow multiplied by the travelled distance on a gridded shop floor. Montreuil and Ratliff (1989), Banerjee *et al* (1990), Chhajed *et al* (1992) performed research on the improvement of the method. Particularly, Montreuil and Ratliff included aisle structure in the layout and the distance measure calculated by the distance travelled. Kim *et al* (1995) first presented a method for generating a cut tree automatically. However, most of the above methods are limited to equal area layout problems.

Otten (1988) and Lengauer (1990) proposed a non-slicing representation by use of a pair of polar graphs. The polar graphs are defined as directed acyclic graphs with a source and a sink, called V-graph and H-graph (Figure 2-13). In the polar graphs each cutting line segment represents a node and each department is represented as a link between a pair of nodes. Based upon polar graphs, Pan and Liu (1995) developed two methods for floor area optimisation by further partitioning the non-slicing floorplan into a slicing problem and then merging the partitioned blocks back into the whole.

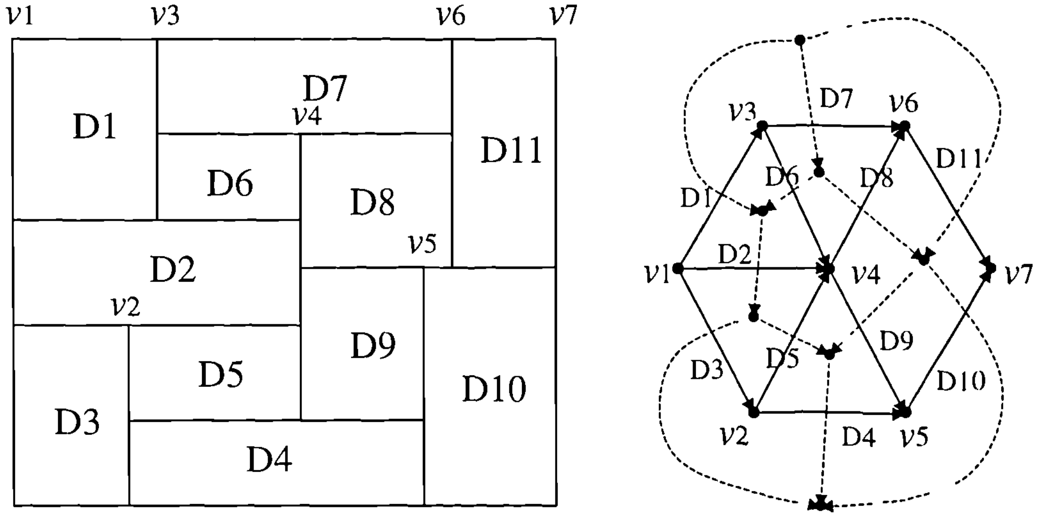


Figure 2-13 A non-slicing floorplan and its polar graphs
(the V-graph in solid lines and the H-graph in dashed lines)

2.4 Material Handling Systems

A material handling system (MHS) carries out the material flow in a production environment. The MHS is related to the layout in three ways. Firstly, it is the material handling system that determines the route of the flow between workstations, the equipment to perform the material transfer, and times at which jobs become available at the workstations. Secondly, the machines or workstation should not conflict with the material handling system. That is, the layout should leave space for the material handling system and, probably the most important, the structure of the layout and the structure of the MHS should match each other. Finally, the cost for material handling is, in fact, determined by the MHS. A manufacturing layout is not practical without considering a corresponding material handling system. The success of a QAP in layout design is dependent upon the ease of adapting its structure to a common MHS.

Therefore, integrated design is appropriate to obtain the optimal solution. Sinriech (1995), Noble *et al* (1998) provided recent reviews of the studies in the design of MHSs. Hassan (1994), Manda and Paleker (1997) also reviewed and analysed the interactions among MHS, layout and scheduling.

Conventional Structures of MHS includes single line structure, spine structure, loop (or U-shape) structure and the structure of a complex network (Heragu and Kusiak 1988, 1990, and Hassan 1994). In addition to the layout pattern, the type and the quantity of the material handling equipment (MHE), the number of machines to be served and the density of the material flow, are also important factors in the determination of a material handling structure. When multiple vehicles are used, the vehicle control and scheduling issue becomes a serious problem. Some modern structures, such as tandem systems for AGVs (automatically guided vehicles) and SFT (segmented flow topology) systems, have been developed in order to reduce the complexity of the vehicle control and scheduling (Sinriech 1995).

2.4.1 Single Line Structure

Single line structure is typically equipped with a conveyor or AGV. When a conveyor is used, the material flow is usually uni-directional, although a bi-directional conveyor can be applied to handle the backtracking. If applying an AGV for transport, only one vehicle is allowed on the track without conflicting to move in the full distance. Thus, the capacity is limited. An AGV works in two directions, handling backtrackings without difficulty. The layout of departments could be a single row (Figure 2-14) or double rows (Figure 2-15). The benefits of the single line structure include low material handling cost, less delays and better control of operations (Hassan 1994).

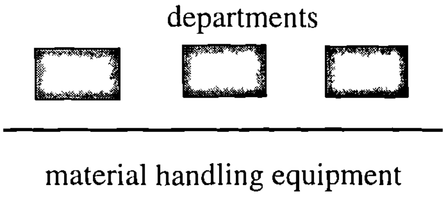


Figure 2-14 A single line structure with single-row departments

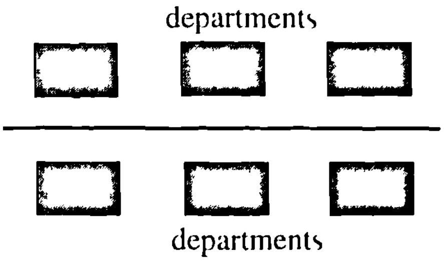


Figure 2-15 A single line structure with double row departments

More literature will be presented in Chapter 4, where further details are required.

If an industrial robot serves for a single row of machines, the line has to bend into an arc around the robot and a circular structure is formed (Figure 2-16). The circular structure is particularly used in robot cells, or as a part of a robot cell which contains a number of robots for material handling (Appleton and Williams 1987).

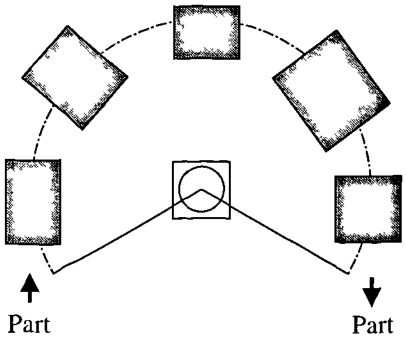


Figure 2-16 A circular structure (Appleton and Williams 1987)

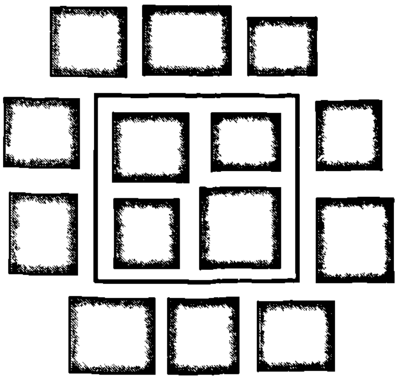


Figure 2-17 A loop structure

2.4.2 Loop and U-shape Structure

When the two ends of a line track are connected, the structure becomes a loop structure (Figure 2-17). A loop structure allows more than one carrier on the track. This allows an increase in the transportation capacity, provided that the vehicles run in the same direction (clockwise or anti-clockwise). A loop structure can also allow one vehicle to travel in the shortest distance between two departments a long the track. However, even the path-problem of a loop layout alone is still complex. De Guzman *et al* (1997) proved that either the shortest path problem or the shortest loop problem in a given machine layout is NP-complete.

2.4.3 Spine Structure

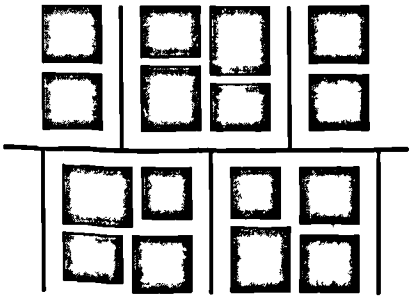


Figure 2-18 A spine structure

Tompkins and Spain (1983) suggested a spine structure that is an extension of the single line structure and created by connecting a number of sub-lines with the main line (Figure 2-18). Industrial trucks, AGVs, and conveyors can be used. One of the main merits of the spine structure is its flexibility in routing and capacity. However, congestion and conflict of the vehicles may occur when more than one carrier is used. Langivan *et al* (1994) presented a hybrid method of heuristic and integer programming for designing spine layouts with respect to the handling costs and the investment costs.

2.4.4 Complex Network

A complex network (Figure 2-19) for material handling is normally used in a large plant equipped with industrial trucks. This structure is the most flexible and allows the shortest route for material handling. However, if AGVs are selected as the transport, a sophisticated traffic controller and vehicle scheduler is necessary. In fact, due to the difficulty of preventing congestion, blocking and conflict, not many totally automatic controlled systems have been employed to complex material handling networks. Alternative techniques have been developed to solve the above problem and are discussed below.

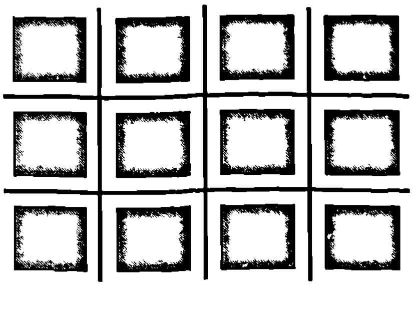


Figure 2-19 A complex network

2.4.5 Tandem Systems

The tandem configuration was suggested by Bozer and Srinivasan (1991). A tandem system is composed of a number of non-overlapping, closed loops for AGVs and each loop is traversed by a single-vehicle (Figure 2-20). Transfer stations or buffers are needed for inter loop transfers. This arrangement eliminates traffic congestion, blocking, vehicle conflicts and vehicle scheduling problems (Mahadevan and Narendran 1994). However, because of the delay in transfer buffers, the tandem system has a higher expected travel time per load and thus a greater lead time (Bischak and Stevens 1995). Because of the many loops involved, a larger number of carriers will be needed and the high utilisation of the carriers may be difficult to obtain

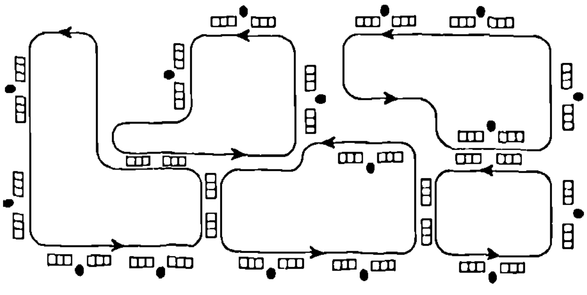


Figure 2-20 Tandem flow-path configuration (Rim and Bozer 1991)

in practice. Also, the system reliability is sensitive to carrier failure (Sinriech 1995). The vehicle speed and the job arrival distribution significantly affect the performance of a tandem system (Choi *et al* 1994).

2.4.6 Segmented Flow Topology

The concept of the segmented flow topology (SFT), developed by Sinriech and Tanchoco (1994 and 1995), is based upon the assumption that the material handling requirements exist only between a few points as defined by a given process plan. An SFT system "is comprised of one or more zones, each of which is separated into non-overlapping segments with each segment serviced by a single material handling device." (Sinriech 1995) An example of an SFT system is shown in Figure 2-21. An SFT system keeps the main advantages and drawbacks of the tandem configuration described in the previous sub-section, but SFT systems have some extra advantages. Compared with a tandem system, the shortest travelling distance within a zone can be obtained in an SFT system. An SFT system can also represent all conventional structures described earlier. Moreover, rather than limited to an AGV system, it could be a system with mixed material handling equipment.

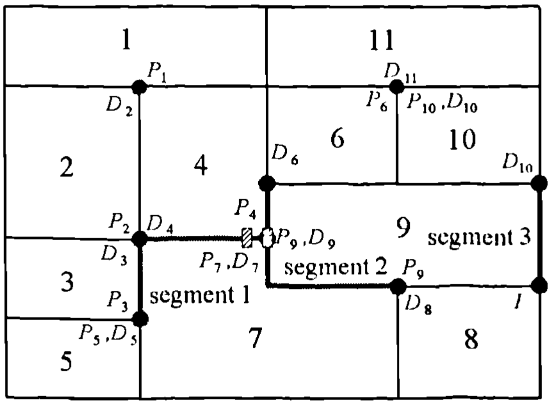


Figure 2-21 Segmented flow topology (SFT) type network (Sinriech 1995)

2.5 General Combinatorial Optimisation Methods

This section will present some algorithms for general combinatorial optimisation problems, especially popular in recent years for solving cell formation and layout problems. Other techniques can be reviewed by reference to the books of Foulds (1981), Parker and Rardin (1988).

2.5.1 Simulated Annealing Algorithms

Simulated annealing (SA) is a method that combines random selection and a hill climbing search (Poole *et al* 1998) to find global optima. In particular, it does a

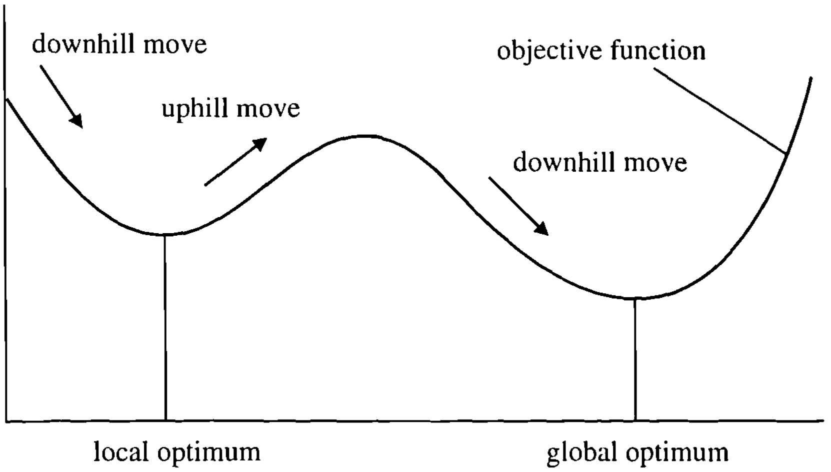


Figure 2-22 Uphill move in a simulated annealing algorithm

random walk, choosing neighbours at random and deciding at random whether to visit that neighbour. If a neighbour is visited and has a better value, the procedure continues from the neighbour. If the neighbour has a worse value, it still could be chosen with a probability depending on the temperature and the difference in the values, that allows the algorithm to go over ridges to find the global optimum as shown in Figure 2-22. It was popularised by Kirkpatrick *et al* in 1983 to solve combinatorial optimisation problems. As the name implies, it is based upon the simulation of the physical annealing process in which a solid metal is heated to a temperature, slightly higher than its re-crystallisation temperature, and then cooled down until it crystallises into a state with a different structure. (Rich and Knight 1991)

Implementation of a SA algorithm presupposes an objective function *F*(*s*), a solution space *s*, and a cooling schedule of the algorithm. A general simulated annealing algorithm is shown in Figure 2-23 (Tam 1992). An instance of a cooling schedule involves

```
$M = 0; i = 0;$
Initialise configuration $s_i$ and temperature $t_M$;
(the symbols are as defined in the adjacent text)
Repeat
    $s_j = G(s_i)$
    $\Delta = F(s_j) - F(s_i)$
    if $\Delta \le 0$ (downhill move)
        $s_i = s_j$
    else
        $s_i = s_j$ with probability $P - e^{-\Delta\ t_M}$
    end if
    $t_{M+1} = d(t_M)$ (decrease the temperature)
    $M = M + 1$
Until stopping criterion = true.
```

Figure 2-23 A general simulated annealing algorithm (Tam 1992)

the following parameters:

(1) Initial "temperature" t_0

The "temperature" is a control parameter which governs the acceptation rate of uphill moves. Typically, initial temperature t_0 is so determined that the initial probability:

$$P = 0.6$$

(2) A generation function $G(s_i)$

The generation function generates new points within the solution space as the neighbours that can be selected randomly. The pair-exchanging as in the CRAFT algorithm, for example, is a generation function that creates the neighbour solutions.

(3) A function $d(t_M)$ to decrease the temperature

Generally, the temperature decreases by a constant in each iteration as in the following equation.

$$t_{M+1} = \beta \times t_M \tag{2-25}$$

where $0 < \beta < 1$.

(4) Stopping criterion

The programme stops when a further downhill move cannot be found and the temperature is low enough, that is,

$$t_M \le \alpha \tag{2-26}$$

where α is given.

Boctor (1991 and 1996), Lee and Wang (1992), Adil *et al* (1996 and 1997), Sofianopoulou (1997), Su and Hsu (1998), Zhou and Askin (1998), Zolfaghari and Liang (1998), and others applied simulated annealing to part grouping and machine cell formation problems in group technology. Mavridou and Pardalos (1997) gave a review of SA for the layout problem. The more recent applications of SA were studied by Meller and Bozer (1996), Alsultan and Bozer (1998), Cagan *et al* (1998), Chwif *et al* (1998), Ho and Moodie (1998), Szykman *et al* (1998), Wang *et al* (1998).

2.5.2 Tabu Search Algorithms

Tabu search (TS) is a heuristic approach for solving combinatorial optimisation problems. In order to prevent cycling behaviour near a local optimum, it uses a tabu (i.e. forbidden) list to keep track of a set of solutions that the method does not want to examine in the next few iterations. The method was first developed by Glover in 1986. Skorin-Kapov (1990) and Nissen (1994) applied TS to the quadratic assignment problems. Chiang and Kouvelis (1996) improved the method of Skorin-Kapov and claimed that the improved method is capable of solving large-scale problems of up to 200 machines.

A tabu search method for layout problems starts with an initial solution, which is generated either randomly or by construction algorithms. By using pair-exchanges like that in CRAFT, the method generates from the current solution a list of candidate solutions, which are evaluated if they are not forbidden. Typically, a candidate is forbidden if it reverses a recent exchange and therefore potentially causes a cycling. The best solution is put into the tabu list and selected as a new beginning.

The main virtues of the tabu search algorithm include:

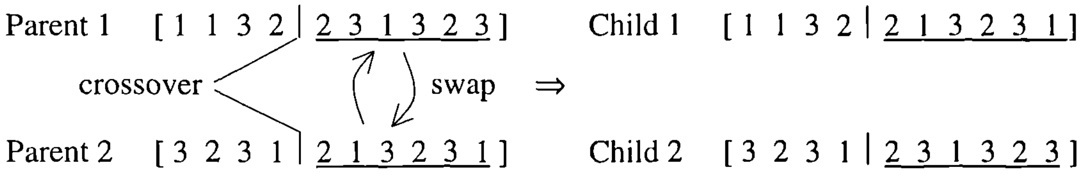
- Avoids cycling behaviour.
- Narrows the search space.
- Avoids trapping around the local optimum.

Applications of the tabu search to cell formation and layout problems were also given by Bland and Dawson (1991 and 1994), Logendran *et al* (1994), Sun *et al* (1995), Logendran and Ko (1997), Vakharia and Chang (1997), Aljaber *et al* (1997), Chiang and Chiang (1998), Chittratanawat and Noble (1999), and others.

2.5.3 Genetic Algorithms

Genetic algorithms (GAs) were first introduced by Holland in 1975 and originated from studies of cellular automata. A GA is an evolution and search procedure that is modelled on the mechanics of natural generation and selection rather than a simulated reasoning process. A genetic algorithm is a competitive method to solve large, unsmooth or noisy problems, especially to find a sufficiently good solution instead of a global optimum (Mitchell 1996). The representation of a string of genes makes it especially good for combinatorial optimisation problems, such as those in group technology or facility layout.

Like its biological counterpart, in GAs a solution space is represented by alphabets or integrals called genes. A string of genes is called a genome or a chromosome. The entire set of variables may comprise of one or more genomes. The different possible settings in the solution space for a gene are called alleles. Mutation is what prevents the loss of diversity at a given bit position, i.e., the change of each gene into another form in the alleles. A mutation rate is the probability of the change of a gene setting at a bit position. A solution set with the genes of certain settings is an individual. A group of individuals is called a population. The number of the individuals in a population is the population size. For instance, in a cell formation problem (Gupta *et al* 1995 and 1996) of ten machines grouped into three-cell, an individual could be [1 1 3 2 2 3 1 3 2 3], in which each gene represents a machine and its value or setting represents a cell in which the machine is clustered. The alleles for all the genes are [1, 2, 3]. Individuals are chosen from a population according to their fitness as parents to generate offspring. Individuals with higher fitness values (better solutions) have better chances of being selected as parents. Parents generate children by crossover, which allows the children to inherit the genes from the parents. For example, the crossover procedure of the above cell formation problem are illustrated below.


Parent 1 [1 1 3 2 | 2 3 1 3 2 3]
crossover
swap
⇒
Parent 2 [3 2 3 1 | 2 1 3 2 3 1]
Child 1 [1 1 3 2 | 2 1 3 2 3 1]
Child 2 [3 2 3 1 | 2 3 1 3 2 3]

Each gene of the individuals is mutated with a probability by another gene randomly selected in the corresponding allele. If the first gene of Child 1 and the last gene of Child 2 are mutated, the children could be as follows.

Child 1 [1 1 3 2 2 1 3 2 3 1] ⇒ [3 1 3 2 2 1 3 2 3 1]
Child 2 [3 2 3 1 2 3 1 3 2 3] ⇒ [3 2 3 1 2 3 1 3 2 2]

Successive populations are called generations. A general GA creates an initial generation, $G(0)$, and for each generation, $G(t)$, generates a new one, $G(t+1)$. Mutation is applied when a new generation is produced. The evolution procedure stops when a certain criterion is satisfied. An abstract view of the algorithm (Buckles and Petry 1992) is shown in Figure 2-24.

```
Generate initial population, G(0);
Evaluate G(0);
t = 0;
Repeat
    Select the parents from G(t);
    Determine the crossover;
    Generate children;
    Evaluate the children;
    Produce new generation G(t+1);
    t = t + 1;
Until stopping criterion = true.
```

Figure 2-24 A general genetic algorithm

In building a GA, there are six fundamental issues that affect the performance of the GA: chromosome representation, initialisation of the population, selection strategy, genetic operators, termination criteria, and evaluation measures (Joines *et al* 1996). Chromosome representation is a way to represent the design variables by genes. Suresh *et al* (1995), for instance, utilised a string of integers to represent a quadratic assignment problem in layout design, where the integers denote the facilities and their positions in the string denote the positions of the facilities in the layout. An initial population is commonly generated by creating random solutions, whose number equals the population size. Selection strategy may be twofold: selecting parents to reproduce and selecting some individuals whose life will be prolonged to the next generation. Some techniques, such as roulette wheel selection, are used to carry out the principle of "survival of the fittest". The most important GA operators are crossover and mutation. In many applications, crossover or mutation has a mechanism to handle the possible production of illegal children. For example, in a QAP problem of nine facilities and nine locations, in the representation of Suresh *et al*, a two-point crossover would generate illegal children from their parents as illustrated below.

Parent 1 $[9\ 3\ 2 \mid 5\ 1\ 4\ 7 \mid 8\ 6]$ Child 1 $[9\ 3\ \underline{2} \mid \underline{2}\ \underline{6}\ 7\ 5 \mid 8\ 6]$

Parent 2 $[4\ 8\ 3 \mid 2\ 6\ 7\ 5 \mid 9\ 1]$ Child 2 $[\underline{4}\ 8\ 3 \mid 5\ \underline{1}\ \underline{4}\ 7 \mid 9\ \underline{1}]$

Some facilities, such as machine 2 and machine 6 in Child 1, are assigned to more than one position, while some others, machine 1 and machine 4 in the example, are excluded in the layout. Illegal individuals are those that cannot represent the solutions to the problem.

Some crossovers, such as PMX (Partially Matched Crossover), OX (Order Crossover) and CX (Cycle Crossover), are specially designed to deal with the above problem (Chan and Tansri 1994, Michalewicz 1996). The details of these crossovers will be given in the next chapter. The most common termination criterion is the number of the generations evolved. The evaluation function measures the fitness of each individual. Normally, it is the objective of the problem. For instance, Joines *et al* (1996) used grouping efficacy (Kumar and Chandrasekharan 1990) as the evaluation function.

Various researchers have employed genetic algorithms in group technology. Lee *et al* (1997) presented a genetic algorithm for the cell design. They used a GA for selecting the alternative routings, in which an integral string for representing the routine for each part, while machine clustering is performed by a similarity-based method. They also proposed a GA for clustering by utilising a string of blocks for the cells. Each block contains two genes, in which the first gene shows the start machine number of the cell and the second indicates the end machine number. However, the authors did not give details of how to rearrange the machines and how to prevent generating illegal strings by the crossover. Hsu and Su (1998) proposed a genetic algorithm for machine-component grouping, considering duplicated machines in different cells. Like the representation by Gupta *et al* (1995 and 1996), the chromosome is a string of machine types and the value of each gene represents the cell in which the type of machine is grouped. Nevertheless, this representation cannot allow a type of machine to be assigned to more than one cell. Therefore, the representation cannot present the problem as the author claimed and could not derive the results given. Gravel *et al* (1998) proposed a GA to configure the machine cell and within it another GA was adopted to select the best part routes.

Tate and Smith (1995) used a GA with two chromosomes to solve the unequal area problem in a multi-bay layout (Section 2.3.2.4). One chromosome represents the sequence of departments and the other contains the breakpoints for the bays as represented in the first chromosome. Banerjee *et al* (1997) proposed a mixing of a GA and a random method. A string representing the machines is used in the GA, and the two additional strings, whose value was randomly chosen from a set of feasible values, determines the direction and length relative to the preceding department. Rajasekharan *et al* (1998) presented a two-stage genetic algorithm. The first stage of the GA determines the spatial sequence and the second designs the layout. Tam and Chan (1998) utilised *Gambler's ruin* (Sedgewick and Flajolet 1996) to encode the slicing tree structure (Tam 1992) in a parallel GA, which was implemented on a parallel computer or a cluster of workstations. Islier (1998) and Kochhar *et al* (1998) built their GAs based on the concept of space filling curves (Bozer *et al* 1994) to represent the layout. For a description of other applications of GAs to layout problems refer to the survey by Mavridou and Pardalos (1997).

2.5.4 Knowledge-based Algorithms

One of the main branches of artificial intelligence (AI) is the knowledge-based expert system. An expert system contains a knowledge database, in which the knowledge is obtained from human experts, in a way simulating human experts for various decision-making in a certain domain. "Expert systems derive their power from a great deal of domain-specific knowledge, rather than from a single powerful technique." (Rich and Knight 1991)

Generally, a knowledge-based expert system consists of a rule base, a database and an inference engine (Heraque 1989, Heragu and Kusiak 1990). In a knowledge-based expert system for layout design, the knowledge base consists of the rules for determining the type of layout and the material handling system, rules for creating initial assignments and directing searches, and rules for evaluating and analysing results. The database has the data of known facts regarding parts, process sequences, transport, machines and their dimensions. The inference engine interprets rules and makes decisions by matching the rules in the knowledge base against the information in the database. Kumara *et al* (1987) developed an expert system that considers multiple objectives and includes a learning facility to obtain the knowledge about new criteria. Later, they added a pattern recognition approach to the facility layout problem for sorting the material flow patterns (Kumara *et al* 1988). Many other expert systems, such as KBFP (Banerjee and Nof 1994), FLEXPERT (Badiru and Arif 1996), CLADES (Babu and Yao 1996), etc., have been developed for facility layout design. Kusiak (1987b, 1988), Basu *et al* (1995), Srinivasulu and Kumar (1997), and others applied expert systems to group technology.

Due to the difficulty of representing the many constrains in MHE selection by an analytical method, knowledge-based systems provide a powerful tool. Gabbert and Brown (1989), Matson *et al* (1992), Park (1996), Kim and Eom (1997) performed research in this area. Welgama and Gibson (1995b) presented a knowledge-based hybrid algorithm for the selection of material handling systems. The algorithm consists of an expert system and an optimisation approach. The knowledge systems is used to obtain a feasible set of MHE types, and then the total cost of investment and area of aisle space is minimised by applying an incorporated algorithm proposed by Webster and Reed (1971), Hassan and Hogg (1985). Arizne *et al* (1989), Welgama and Gibson (1996) studied the integration of layout and MHS design by utilising knowledge-based systems.

2.5.5 Neural Network Approaches

Another branch of artificial intelligence is artificial neural networks (ANNs), which sometimes is simplified to "neural nets". Neural networks are "massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organisations which are intend to interact with the objects of the real world in the same way as biological nervous systems do." (Kohonen 1988) A neural network is a collection of neurons (nodes) interconnected by adjustable weights. A positive weight presents an excitatory connection; a negative weight gives an inhibitory connection. A neuron sums all of its weighted inputs and determines the neuron's response, or output, by an activation function. Figure 2-25 shows the structure of a simple neural network with one hidden layer. Neural networks

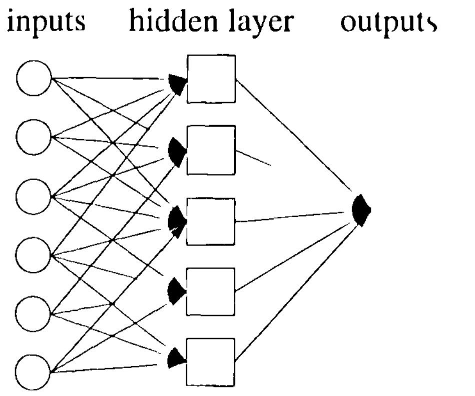


Figure 2-25 A simple neural network

have learning ability to obtain experiential knowledge from examples by adjustment of the weight values.

Kao and Moon (1991), Moon (1992), Moon and Chi (1992) applied a feedforward neural network to group technology for identifying part families. The neural network consists of three layers: a part layer, a machine layer, and a hidden layer. Each neuron in the part layer represents a part and each neuron in the machine layer represents a machine. The number of the neurons in the hidden layer equals the number of parts or the number of machines which ever is greater. The part layer connects to the hidden layer with a one-to-one correspondence, assuming that the number of parts is greater than that of machines. The machine layer connects to the hidden layer representing the part-machine requirement, i.e. the incidence matrix. The weights of the intra-layer connections are determined by a similarity coefficient described in Section 2.2.1. The operation of the neural network starts with choosing a neuron arbitrarily as a seed for each cluster. The neurons with activation values greater than a threshold are grouped together. More applications of the various neural networks in manufacturing can be found in the review of Zhang and Huang (1995). Some recent studies have been carried out by Dagli and Huggahalli (1995), Chen and Cheng (1995), Burke and Kamal (1995), Arizono *et al* (1995), Kamal and Burke (1996), Chen *et al* (1996), Zolfaghari and Liang (1997), Kao and Moon (1995, 1997 and 1998), Jang and Rhee (1997), Chu (1997), Alsultan (1997), Pilot and Knosala (1998), Enke *et al* (1998), and others. However, most of the above papers used a binary part-machine incidence matrix in their models and did not consider the operation sequences. Chang and Hsiao (1995) applied neural networks to the design of VLSI layout with rectilinear shaped departments. Chung (1999) built a cascade BAM (Bi-directional Associative Memories) neural expert system with the capability of incrementally learning for facility layout design.

2.6 Conclusions

Although much effort has been exerted in the past three decades, cell formation and layout problems are far from solved. Many of the previous studies of plant layout were concentrated on soft block layout problems (the layout problems within a cell will be discussed in Chapter 3). The design procedure for a soft block layout is typically that described by the following sequence (Meller and Gau 1996a).

Cell Formation → Block Layout → Detailed Layout

In the block layout design, a soft block is constrained to a previously specified area and certain bounds (lower and upper) to the shape ratio. This brings two problems. First, it is difficult to estimate the area with accuracy. The area will determine the distance between the blocks and the distance is very important in the evaluation of the layout. Secondly, even though the area is calculated precisely, the shape ratio could not, in most cases, be maintained in the detailed layout. The shape of the cell can only appear in a number of certain aspects, such as one row of machines or two rows of machine in a flowline cell. Therefore, there is a need for a method to design layout in a reverse sequence. That is, the design of the cell layout precedes the design of the block layout. Thus, the block layout problem becomes that of hard blocks, as in the case of VLSI design. The following is the proposed design procedure for flexible manufacturing systems.

Cell Formation → Cell Layout → Block Layout → Detailed Design

In order to evaluate layout precisely, aisle distance is required. Consequently, aisle structure and layout should be considered simultaneously. For that purpose, a slicing structure provides the best representation for the following reasons.

- Slicing structure represents layout and aisle structure simultaneously.
- It can represent the most common material handling network, including single lane layout, spine layout, matrix layout, etc. The structure also allows the SFT configuration.
- It provides simple straight aisles.

- The aisles can be classified into a number of levels in nature. It is normal that the main aisles are wider than the sub-aisles.
- The aisle distance can be automatically computed in the representation of the layout.
- The overlapping constraint and the rectangular border constraint are automatically applied.

The following chapters will implement the above ideas.

Chapter Three

Cell Formation

Cell configuration is an important pre-layout stage in the design of cellular manufacturing systems. There are many methods available for cell configuration, which have been presented in the previous chapter. This chapter presents a sequence-based material flow heuristic for machine clustering problems in group technology (GT). The algorithm links machines according to part volumes and part types based upon operation sequences, whilst ignoring the specification of the number of cells and any cell size limitation, so they are 'naturally occurring' (Mosier 1989). Then the natural cells are merged according to the intercell flow and any pre-specification or limitation, also keeping the 'natural' property. A local optimum is guaranteed by a final refinement procedure.

3.1 Background

Group technology (GT) in manufacturing systems is a management strategy which groups parts that have similar processing requirements into part families and clusters machines that fulfil these requirements into machine cells. The benefits of application of GT include: the reduction of material handling cost, the reduction of work-in-progress inventory, improvement of machine utilisation, and therefore, improvement in the productivity of the systems. This chapter focuses on the manufacturing cell formation problem by considering process sequences and flow volumes.

By focusing on a graph representation, the grouping problem is to find a number of cut sets that divide the flow graph into a pre-specified number of sub-graphs such that the flow between the sub-graphs is minimised, provided that the number of nodes in each graph does not exceed a predefined number. In the representation, each node is denoted as a machine, the weight of an edge between two nodes is the flow volume between the machines and each sub-graph is a machine cell.

According to Mosier (1989), the specification of the number of cells or limitation of the number of machines within a cell "adds undesirable subjectivity to the problem" and "inappropriately forced cluster formations may have an effect more detrimental than beneficial". Mosier continues by observing, "Clearly, successful procedures for addressing this problem will consider both work flow and control issues. The nature of the machine cell formation problem makes it desirable to seek solution techniques which identify naturally occurring cells." He concludes that, "With the machine grouping problem it is reasonable to seek a solution which minimises between-cell component part transfers. Unfortunately, this could lead to the simple-minded solution of forming only one machine cell, i.e. the whole shop." Unfortunately, most of the previous methods have to use the specification of the number of cells or a cell size limitation during the cell formation procedure.

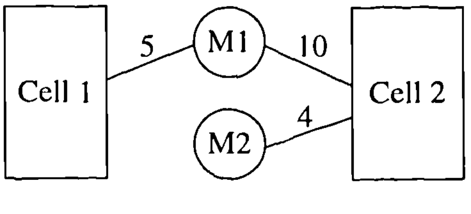


Figure 3-1 An instance of cell formation

The use of artificial limitations may cause unreasonable solution. Studying a cell formation instance illustrated in Figure 3-1, if only one machine can be added into cell 2 due to the cell size limitation, which one, M1 or M2, should be put into cell 2? According to the measure of minimising the intercell flow, M1 seems the best candidate and therefore M2 must be allocated into cell 1. The argument is, why leave M2 standing alone in cell 1. Conversely, removing cell size limitations, M2 is added into cell 2 first because it has no link to cell 1 and then M1 is added into cell 1, by the natural cell formation procedure described in the following section. A pre-specified quantity of cells has a similar problem. The argument is, why should give a pre-specification of four, if three independent cells with a reasonable cell-size exist.

One reason for limiting the cell size is the consideration of the area limitation of the shop floor and the area planning. Due to the variety of machine dimensions, the cell size limitation does not imply that the cell is limited to a certain area. Another reason is the consideration of management so that the operators, who work in the cell, can handle the number of machines. However, it is still a fuzzy limitation, because there is no significant difference between a seven-machine cell and an eight-machine cell. The third reason might be the balance of the loading, however, the cell size gives little clue of loading. Moreover, why does the loading of each cell have to be balanced? Before decisions of management issues are made, why not examine what naturally occurring cells look like, if they exit? Thus, "a more appropriate approach would be to identify 'naturally occurring' families" (Kao and Moon 1997).

The problem discussed in this chapter is how to explore 'naturally occurring' cells without pre-specification of the number of cells or cell size limitations. The proposed algorithm is based on inter-machine mass flow represented by the volume of parts, as in the case of INCFR (Okogbaa *et al* 1992) and the control issue, which is represented by the number of part types travelling between machines or cells. The more intercell mass flow, the more the transportation costs; the more part types that travel between cells, the more complex the transportation control. Thus, the aim in this research is to minimise both.

3.2 Exploring Natural Cells: A Heuristic

In an optimal solution of the cell formation problem, it can be noted that, if a machine is moved from one cell to another, the sum of the intercell flow cannot be further reduced. Thus, the idea of exploring "natural cells" is to group a number of closely related machines first, then add a machine into the cluster one at a time and finally, when the intercell flows are reduced to the minimum, a "natural cell" has been formed. For instance in Figure 3-1, M2 is regarded as more affinity to cell 2 than M1, not only because adding M2 to cell 2 reduces the total intercell flows, but also because M2 has no inter-machine flow to cell 1.

In another instance, Figure 3-2 gives an example of a two-machine natural cell. If either M3 or M4 were added into the cell, the intercell flow of the cell would increase. A cell is "natural" because the cluster is determined by the inter-machine flows only and no artificial force is applied during the machines are grouped into machine cells.

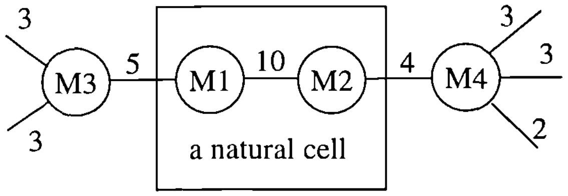


Figure 3-2 An example of a two-machine natural cell

At this point it will be useful to suggest a definition of a natural cell. A natural cell is a set of machines in a non-empty, sub-flow-graph divided by a minimal cut. A minimal cut is a cut of a graph such that the sum of the weights of the edges across the cut-line is a local minimum. This definition and a heuristic algorithm, which will be described later, can prevent a machine standing alone in a cell, unless it alone can be formed as an independent cell. Independent cells, between which there is no flow, are ideal natural cells.

The heuristic for cell formation in this chapter includes three stages: the natural cell exploring stage, the cell merging stage and the refinement stage. The details of the three stages will be described in the following sub-sections.

3.2.1 Representation of Process Sequences and Inter-machine Flow

Suppose a number of parts to be manufactured and their volumes are known and the process operations and sequences for each part are also known. Each operation has been assigned to one machine.

The process sequences of each part are recorded in a one-dimensional array per part. The element number of the array indicates the operation number; the element of the array stores the machine number, which the part visits for the operation. Examples of the representation of a process sequence are given in Appendix A. This form of presentation prevents problems that arise if two or more non-consecutive operations visit the same machine and also minimises the memory requirements for data storage (Heragu 1994).

An inter-machine flow matrix, illustrated in the example in Table 3-1, is used to indicate the material flow between one machine and another. The upper diagonal chart of the inter-machine flow matrix shows the mass flow between machines and the lower diagonal chart is used for storing the number of part types flowing between machines. Each element of the matrix is computed by the following equations:

	m1	m2	m3	m4	m5	m6	m7	m8
m1	0	50	0	15	30	0	0	0
m2	2	0	20	30	10	0	0	0
m3	0	1	0	35	0	0	0	0
m4	1	1	2	0	0	0	0	0
m5	1	1	0	0	0	35	50	0
m6	0	0	0	0	2	0	45	30
m7	0	0	0	0	2	2	0	25
m8	0	0	0	0	0	2	1	0

Table 3-1 An inter-machine flow chart

$$f_{kl}^{v} = \sum_{i=1}^{n} \sum_{j=1}^{s_i - 1} v_i \cdot x_{ij} / \rho_i \qquad (k < l)$$

and
$$f_{kl}^{p} = \sum_{i=1}^{n} \sum_{j=1}^{s_i - 1} x_{ij} \qquad (k > l) \qquad (3\text{-}1)$$

where f_{kl}^{v} is the calculated volume of the mass flow between machine k and machine l if $k < l$ and

f_{kl}^{p} is the number of part types between machine k and machine l if $k > l$.

x_{ij} represents the operation sequence expressed mathematically as:

$$x_{ij} = \begin{cases} 1 & if \quad k = mc_{i,j}, l = mc_{i,j+1} \quad or \quad l = mc_{i,j}, k = mc_{i,j+1}; \\ 0 & else. \end{cases}$$

k, l are indexes of machines.

i is index of parts.

n is the total number of parts.

j is index of process sequences.

s_i is the number of the operations within the process sequences for part *i*.

v_i is the volume of part *i*.

ρ_i is the intercell transportation lot size of part *i*.

mc_{ij} is the machine number visited by part *i* in operation *j*.

The above mathematical equation may appear complicated but the computer program for its implementation is quite simple and efficient.

3.2.2 Mathematical Model

As described earlier, the main purposes of the cellular manufacturing is the reduction of material handling cost and the simplification of production management. Therefore, the objective of good cell formation is the reduction of the cost of intercell material handling and control. The mathematical model can thus be expressed by the following equation.

$$\text{Minimise } \sum_{i=1}^{m}\sum_{j=1}^{m}[\alpha\cdot f_{ij}^{v}+(1-\alpha)f_{ij}^{p}]\cdot(1-x_{ij}) \qquad (3\text{-}2)$$

where s_i is the mass flow between machine *i* and machine *j* and,

f_{ij}^{p} is the number of part types between machine *i* and machine *j*.

α is a weight parameter.

$$x_{ij}=\begin{cases}1 & \text{if machine } i \text{ and } j \text{ are assigned to the same cell,}\\ 0 & \text{otherwise.}\end{cases}$$

subject to

$$\sum_{j=1}^{m} x_{ij} > 1 \qquad i\in\{1,\cdots,n\},$$

or

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \text{only if } f_{ij}=0 \text{ for all } i, j \text{ and } i\neq j. \qquad (3\ 3)$$

The design variable x_{ij} guarantees that each machine can be assigned in only one cell. The following constraints ensure that each cell has at least one machine and a machine can become a basic cell only if the machine has no flow from or to any other machines.

3.2.3 Natural Cell Formation

A natural cell, is a group of machines which are mostly related in material flow such that adding any one machine from outside the group will increase the flow between the group and the other machines. By way of illustration in Figure 3-3, suppose that all the machines are divided into three sets: the current cell that is being clustered, the cells already clustered and the remaining machines. A candidate machine m to be evaluated is selected from the set of the remaining machines. The mass flow between the candidate and the current cell, F^v_{mc}, and the mass flow between the candidate and the other two sets, F^v_{mr}, are calculated as follows:

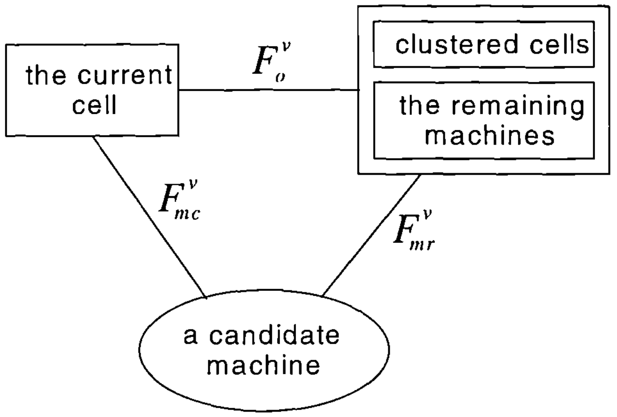


Figure 3-3 Material flow relationship

$$F^v_{mc} = \sum_{i\in C, m<i} f^v_{mi} + \sum_{i\in C, i<m} f^v_{im} \tag{3-4}$$

$$F^v_{mr} = \sum_{\substack{i\in M, i\notin C\\ m<i}} f^v_{mi} + \sum_{\substack{i\in M, i\notin C\\ i<m}} f^v_{im} \tag{3-5}$$

where i is index of parts.

m is the index of the candidate machine.

C represents the set of the current cell.

M represents the set of all machines.

The flow between the current cell and all the other machines F^v_{co} is:

$$F^v_{co} = F^v_o + F^v_{mc} \tag{3-6}$$

where F_o^v is the flow between the current cell and the rest of the machines, excluding the candidate.

If the candidate were assigned into the current cell, its size would be increased by one, effectively creating a new current cell. The flow between the new current cell and the rest of the machines F_{cn}^v would be:

$$F_{cn}^v = F_o^v + F_{mr}^v \tag{3-7}$$

Therefore, substituting from equation (3-6):

$$F_{cn}^v = F_{co}^v + F_{mr}^v - F_{mc}^v \tag{3-8}$$

$$\text{Flow ratio } r_m^v = F_{mc}^v / F_{mr}^v \tag{3-9}$$

If the flow ratio $r_m^v = F_{mc}^v / F_{mr}^v > 1$, the assignment of the candidate to the current cell will decrease the flow between the current cell and all the other machines.

Similarly, the number of part types travelling between the new cell and the rest of the machine F_{cn}^p would be:

$$F_{cn}^p = F_{co}^p + F_{mr}^p - F_{mc}^p \tag{3-10}$$

where F_{co}^p is the number of part types flowing between the current cell and all the other machines.

F_{mr}^p is the number of part types flowing between the candidate and the sets of clustered cells and the remaining machines.

$$F_{mr}^p = \sum_{\substack{i\in M, i\notin C\\ m>i}} f_{mi}^p + \sum_{\substack{i\in M, i\notin C\\ i>m}} f_{im}^p \tag{3-11}$$

F_{mc}^p is the number of part types flowing between the candidate and the current cell.

$$F_{mc}^p = \sum_{i\in C, m>i} f_{mi}^p + \sum_{i\in C, i>m} f_{im}^p \tag{3-12}$$

If the ratio $r_m^p = F_{mc}^p / F_{mr}^p > 1$, the assignment of the candidate to the current cell will decrease the movement of part types.

Define the weighted flows as follows:

$$F_{cn} = \alpha F_{cn}^{v} + (1-\alpha) \cdot F_{cn}^{p} \tag{3-13}$$

$$F_{co} = \alpha F_{co}^{v} + (1-\alpha) \cdot F_{co}^{p} \tag{3-14}$$

where α is a parameter weighted between flow volume and number of part types. $0 \leq \alpha \leq 1$.

Therefore,

$$\begin{aligned} F_{cn} &= F_{co} + \alpha(1-r_{m}^{v})F_{mr}^{v} + (1-\alpha) \cdot (1-r_{m}^{p})F_{mr}^{p} \\ &= F_{co} + [\alpha \cdot F_{mr}^{v} + (1-\alpha) \cdot F_{mr}^{p}] \cdot (1-r_{m}) \end{aligned} \tag{3-15}$$

where r_{m}^{p} is the flow ratio of part types.

$$r_{m}^{p} = \frac{F_{mc}^{p}}{F_{mr}^{p}} \tag{3-16}$$

r_{m} is the weighted flow ratio.

$$r_{m} = \frac{\alpha r_{m}^{v} F_{mr}^{v} + (1-\alpha) \cdot r_{m}^{p} F_{mr}^{p}}{\alpha \cdot F_{mr}^{v} + (1-\alpha) \cdot F_{mr}^{p}} \tag{3-17}$$

If the weighted flow ratio $r_{m} \geq 1$, the candidate should be put into the current cell. If the weighted flow decreases to the minimum, a natural cell is formed (point C in Figure 3-4). Merging any other machines with the flow ratio $r_{m} < 1$ is not acceptable because it will increase the flow (after point C in Figure 3-4).

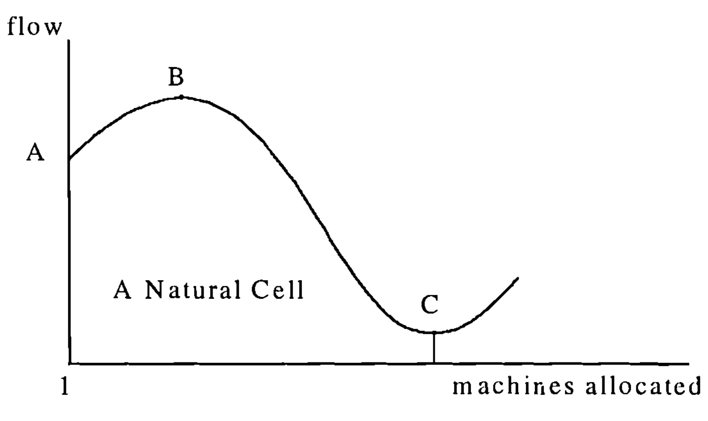


Figure 3-4 A natural cell

It should be noticed that the "natural" property is limited to a specified weight parameter. For a different weight parameter α, there will be a new machine-flow graph in which the natural cells may be different. It should also be noticed that basic cells only represent the natural clustering property and they may need merging and improving to construct the cells which the designer requires. The basic cells that are found at this stage can give a cell designer an idea which machines are strongly linked together.

3.2.4 Initial Cell

The initial cell begins with a starting machine or seed machine which becomes the first machine in the current cell. The flow ratio r_m of each unassigned machine is calculated and the machine with the largest flow is assigned into the current cell. Although this merging may initially cause the flow to increase (from A to B in Figure 3-4), the process allows machines to cluster together in a natural manner. A current cell becomes a natural cell if and only if the intercell flow of the current cell begins to reduce and reduces to a minimum.

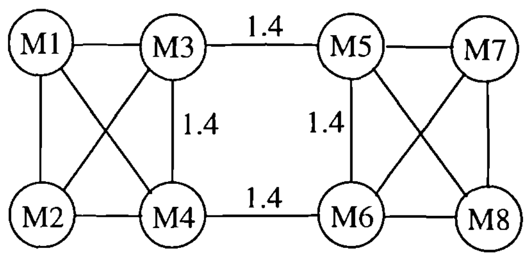


Figure 3-5 An eight-machine system

An eight-machine system illustrated in Figure 3-5 is considered as an example to demonstrate the natural cell clustering procedure previously described. First, M1 is selected as the seed machine and is put into the empty current cell. The flow ratios from equations (3-9), (3-16) and (3-17) of all other machines are then calculated. M2 is found to have the largest ratio and is consequently put into the current cell, ignoring the increase of the flow between the current cell and the remaining machines by the merge (volume of 4). The flow ratios of the remaining machines are re-computed. M3 and M4 have the equally largest flow ratio. Selecting one randomly, say M3, and putting it into the current cell, the flow volume between the current cell and the remaining machines increases to 4.8. Calculating the flow ratio again, then M4, with the largest flow ratio of 3.4/1.4, is put into the current cell. This merge causes the intercell flow to go down to 2.8. The flow ratios of the entire remaining machines are then found to be less than unity. Therefore, a natural cell of M1, M2, M3 and M4 is formed. The programme goes on to cluster the next cell of M5, M6, M7 and M8. In this case, any pre-specification of more than two cells might be inappropriate and unacceptable, because it would distort the natural property of the cell division.

In this example, Harhalakis' twofold heuristic (Harhalakis *et al* 1990) or INCFR heuristic (Okogbaa *et al* 1992) may fail to find the above solution. Both methods suggest clustering M3, M4, M5 and M6 together and the machine cannot be separated during the refinement procedure. Unfortunately, the natural cell exploration (NCE) algorithm may also fail to find the solution if M3, M4, M5 or M6 is selected as the seed machine, so it is important to find a way to solve the problem of selecting the seed machine, which will be discussed later.

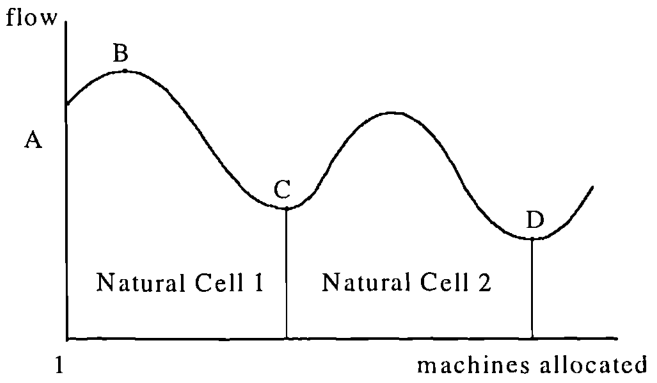


Figure 3-6 Basic natural cells

3.2.5 Basic Natural Cell Exploration Algorithm

A basic natural cell (BNC) is a natural cell that has the minimum number of machines. Point C in Figure 3-6 represents the formation of the first natural cell and point D the formation of the second. This figure can be illustrated by an eleven-machine system in Figure 3-7 as an example. The BNC formation is the first stage of the natural cell exploring (NCE) heuristic and the proposed algorithm is presented in Figure 3-8. A more detailed description of this algorithm is given below:

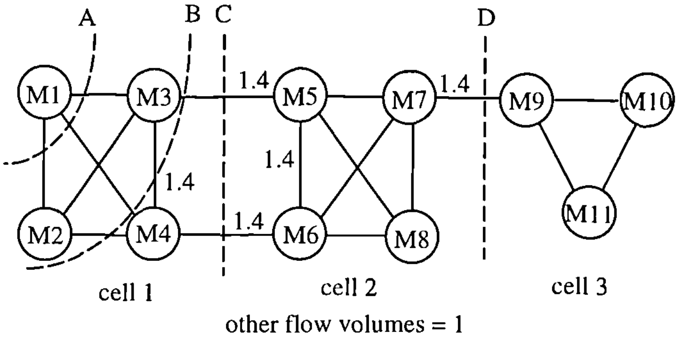


Figure 3-7 An eleven machine system

Step 1 An unassigned machine is selected as the seed machine. The seed machine is then put into the empty current cell.

Step 2 The flow between the seed machine and the other machines is calculated. If the seed machine does not receive or send to other machines, the seed machine itself forms a natural cell and the algorithm goes back to step 1.

Step 3 The flow between each unassigned machine and the current cell (F^v_{mc}, F^p_{mc}) and the flow between each unassigned machine and the remaining machines (F^v_{mr}, F^p_{mr}) are calculated by equation (3-4), (3-5), (3-10) and (3-11).

Step 4 The machine that has the maximum flow ratio r_m is selected as a candidate. The assignment of the candidate is determined by the following conditions.

A. If the ratio r_m < 1, i.e. the flow of the new current cell (F_{cn}) will increase (from A to B in Figure 3-6), and if there is no descent in the previous records, the candidate machine is added into the current cell and the algorithm recurs to step 3 until the flow goes down (after point B).

B. If $r_m \geq 1$, the candidate machine is assigned into the current cell. If $r_m > 1$, the flow F_{cn} decreases (from B to C) and a descent is recorded. The process loops back to step 3.

C. If r_m < 1 with a previous descent of the flow F_{cn}, the current cell formation is finished (point C in Figure 3-6 and 3-7). The algorithm then returns to step 1 to cluster another cell until the set of the remaining machines is empty.

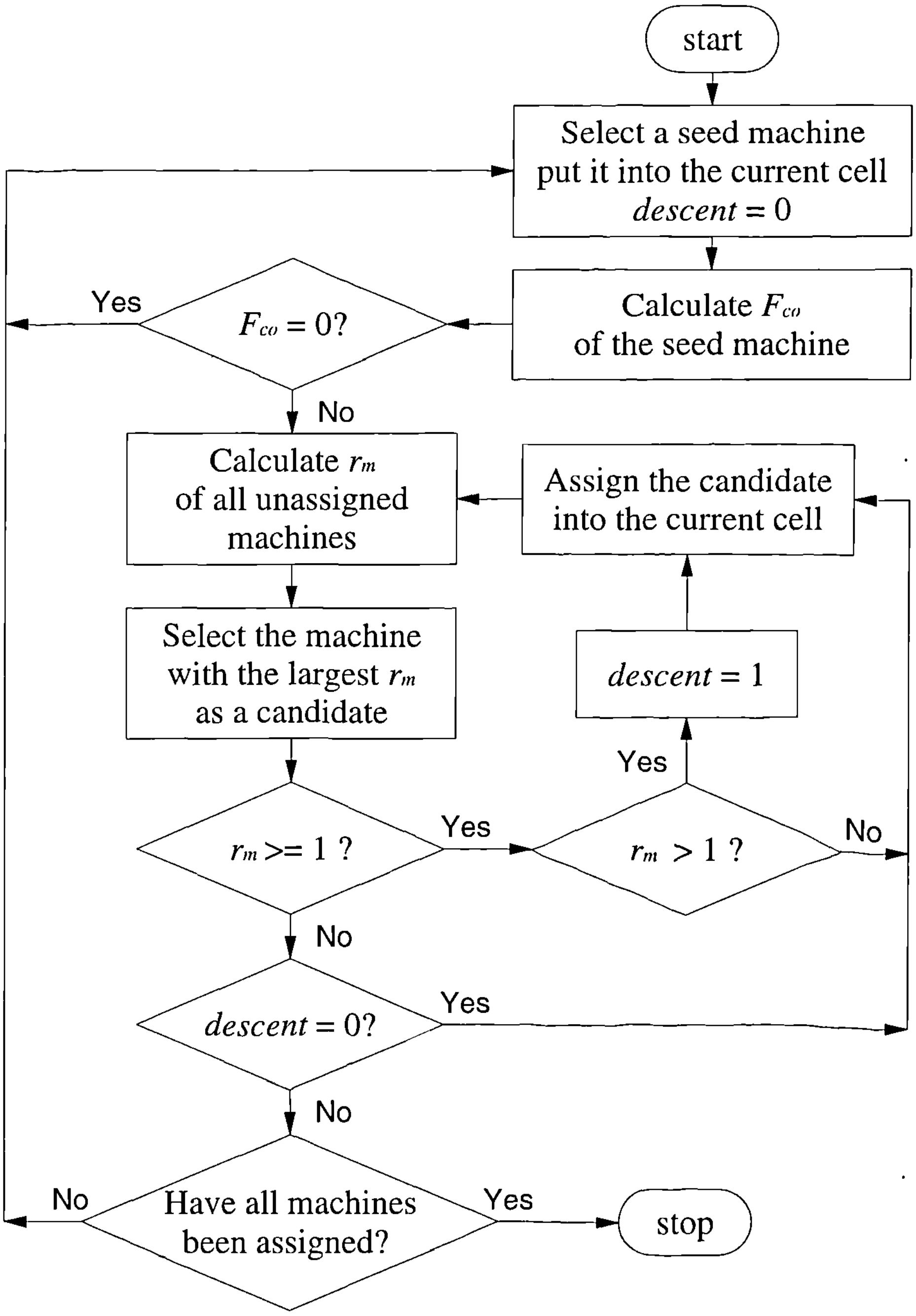

start
Select a seed machine put it into the current cell *descent* = 0
Calculate F_{co} of the seed machine
$F_{co} = 0$?
Yes
No
Calculate r_m of all unassigned machines
Assign the candidate into the current cell
Select the machine with the largest r_m as a candidate
descent = 1
Yes
$r_m >= 1$?
Yes
$r_m > 1$?
No
No
descent = 0?
Yes
No
No
Have all machines been assigned?
Yes
stop

Figure 3-8 The flow chart of the basic-natural-cell exploring algorithm

3.2.6 Cell Merging

So far there has been no limitation on the number of cells. However, for practical reasons, for example, layout etc., at the outset there may be a pre-constraint on the design to use a specified number of cells and a limitation of cell size. Thus, to complete the approach it was necessary to propose a method by which BNCs can be combined to give a system with a prescribed number of cells.

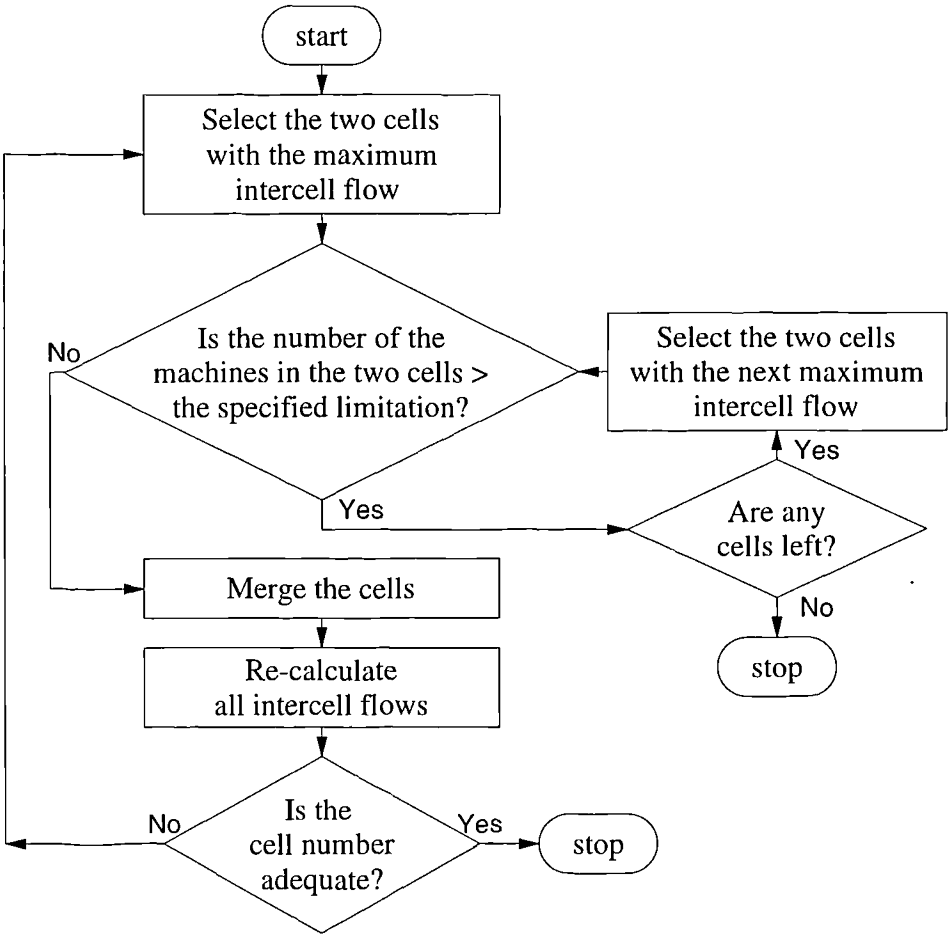

start
Select the two cells with the maximum intercell flow
Is the number of the machines in the two cells > the specified limitation?
No
Yes
Select the two cells with the next maximum intercell flow
Yes
Are any cells left?
No
stop
Merge the cells
Re-calculate all intercell flows
Is the cell number adequate?
No
Yes
stop

Figure 3-9 The cell merging procedure

In the second stage of the NCE heuristic, natural cells that satisfy the requirement on the number of cells can be formed by combinations of BNCs, as shown in the flow chart in Figure 3-9. The main procedure is that the two cells with the maximum flow between them are merged if the merging of the cells does not cause the cell size to exceed the limitation. The merging repeats till the cell number is acceptable. In this stage, only natural cells can be merged to form a larger cell. An instance of the process is given in Figure 3-7, which shows that natural cell 1 or 3 could be merged into natural cell 2, if two cells are preferred.

3.2.7 Cell Refinement

In the cell forming and cell merging stages, merging of machines or cells may change the flow property of a single machine to the clustered cells. In this case a refinement procedure is needed. A refinement procedure, which is similar to Harhalakis' (1990) and Okogbaa's (1992), re-evaluates every machine and moves it into the cell with the largest flow ratio, if the cell size limitation allows. The move is repeated until no machine can be moved. The refinement stage ensures that the solutions are local optima.

3.3 Computational Experiments

In order to evaluate the performance of the above algorithm, some examples have been considered. The algorithm was coded in C++ and implemented on a PC. The programme was tested by means of three problems based upon previously published work. Problem 1 was taken from Okogbaa *et al* (1992), problem 2 from Harhalakis (1990) and problem 3 from Seifoddini and Djassemi (1995). The detailed data for the problems can be found in Appendix A.

Problem 1. In this 18-machine problem the cell boundaries are reasonably clear; thus the cell formation by the NCE algorithm ($\alpha = 0$ or the volumes of all parts is one) is not sensitive to the selection of seed machines. For this reason, the results of the cell formation are identical by using any machine as the seed. The solution using the NCE algorithm, given in Table 3-2, is same as Okogbaa's (1992).

Total Inter-cell Flow: 20

C1:	1	12	7
C2:	4	14	2
C3:	3	6	11
C4:	9	16	5
C5:	13	10	17
C6:	15	8	18

Inter-cell Flow Chart

	1	2	3	4	5	6
1	14	2	1	1	3	2
2	2	10	1	1	1	3
3	1	1	10	1	2	0
4	1	1	1	8	1	1
5	3	1	2	1	10	0
6	2	3	0	1	0	8

Table 3-2 A solution for problem 1

Problem 2. This problem has 20 machines and 20 parts. The operation sequence data of each part is from Harhalakis' work (1990). Two solutions with the same optimised total intercell flow were found by the NCE heuristic (Table 3-3) using a single run. Solution 1 is the same as that of Harhalakis. Another solution (Solution 2) was found, which shows machine 4 can be moved into cell 3 without increasing the intercell flow.

Solution 1:	Solution 2:
Total inter-cell flow: 14	Total inter-cell flow: 14
C1: 4 6 7 13 15	C3: 6 7 13 15
C2: 2 3 5 11 14 16 17	C1: 2 3 5 11 14 16 17
C3: 1 9 10 12 18	C2: 1 4 9 10 12 18
C4: 8 19 20	C4: 8 19 20

Table 3-3 Solutions for problem 2

Total Inter-cell Flow: 2194	Inter-cell Flow Chart								
NPT Between Cells : 25		1	2	3	4	5	6	7	8
C1: 7 26 17 18 5	1	879	0	0	0	0	0	143	31
C2: 20 30 19	2	0	554	0	0	319	0	61	0
C3: 3 22 10 23 12	3	0	0	1162	952	0	110	91	0
C4: 1 11 2 21	4	0	0	11	818	0	0	0	96
C5: 9 29	5	0	3	0	0	331	0	176	0
C6: 13 24	6	0	0	1	0	0	120	76	0
C7: 8 28 27 4	7	2	1	1	0	2	1	428	139
C8: 6 16 14 25 15	8	1	0	0	1	0	0	1	288

Table 3-4 Cells formed by Jaccard's similarity coefficient (Seifoddini and Djassemi 1995)

Total Inter-cell Flow: 1922	Inter-cell Flow Chart								
NPT Between Cells : 22		1	2	3	4	5	6	7	8
C1: 7 26 17 18 5 15	1	910	0	0	0	0	143	0	0
C2: 20 30 19 29 9	2	0	1204	0	0	0	237	0	0
C3: 3 22	3	0	0	534	464	78	0	0	0
C4: 1 11 2 21	4	0	0	6	818	488	0	0	96
C5: 10 23 12	5	0	0	1	5	550	91	110	0
C6: 8 28 27 6 4 16	6	2	3	0	0	1	638	76	139
C7: 13 24	7	0	0	0	0	1	1	120	0
C8: 14 25	8	0	0	0	1	0	1	0	78

Table 3-5 Cells formed by volume based similarity coefficient (Seifoddini and Djassemi 1995)

Problem 3. This is a problem of 41parts and 30 machines. Tables 3-4 and 3-5 show the previously published solutions for cell formation using Jaccard's similarity coefficient and a production volume based similarity coefficient (Seifoddini and Djassemi 1995). The elements in the diagonal of the intercell flow matrix give the mass flows within the cells.

Total Inter-cell Flow: 2456 Inter-cell Flow Chart

NPT : 28		1	2	3	4	5	6	7	8	9	10	11
C1: 3 22	1	534	464	78	0	0	0	0	0	0	0	0
C2: 1 11 2 21	2	6	818	488	0	0	0	0	0	96	0	0
C3: 10 23 12	3	1	5	550	0	0	0	0	91	0	0	110
C4: 7 26 17 18	4	0	0	0	770	109	0	76	67	0	0	0
C5: 5 15	5	0	0	0	2	31	0	0	0	0	0	0
C6: 20 30 19	6	0	0	0	0	0	554	380	0	0	0	0
C7: 29 9 8	7	0	0	0	1	0	4	507	143	0	0	0
C8: 27 28 4	8	0	0	1	1	0	0	2	285	0	139	76
C9: 14 25	9	0	1	0	0	0	0	0	0	78	139	0
C10: 16 6	10	0	0	0	0	0	0	0	1	1	71	0
C11: 13 24	11	0	0	1	0	0	0	0	1	0	0	120

Table 3-6 Cells explored by BNCE

Total Inter-cell Flow: 1473 Inter-cell Flow Chart

NPT Between Cells : 17		1	2	3	4	5	6	7	8
C1: 3 22 1 11 2 21	1	1816	566	0	0	0	0	96	0
C2: 10 23 12	2	6	550	0	0	0	91	0	110
C3: 7 26 17 18	3	0	0	770	109	76	67	0	0
C4: 5 15	4	0	0	2	31	0	0	0	0
C5: 20 30 19 29 9 8	5	0	0	1	0	1441	143	0	0
C6: 27 28 4	6	0	1	1	0	2	285	139	76
C7: 14 25 16 6	7	1	0	0	0	0	1	288	0
C8: 13 24	8	0	1	0	0	0	1	0	120

Table 3-7 Cells with specification of cell quantity and limitation of cell size

From the present work, Table 3-6 shows the basic natural cells explored by randomly selected seed machines ($\alpha = 1$). After the cell merging stage, 11 basic cells were merged into 8 cells, in order to compare to the previously published case, with a limitation of six machines within a cell (Table 3-7). If there were no specification of the number of cells, further merging of the cells would give a better solution (Table 3-8). If the limitation of the cell size was increased to nine machines, the result of the cell merging will be as given in Table 3-9. The resulting part-machine chart is shown in Table 3-10.

```
Total Inter-cell Flow: 1254          Inter-cell Flow Chart
 NPT Between Cells : 14                 1    2    3    4    5    6
 C1:  3 22  1 11  2 21           1   1816    0    0   96  566    0
 C2:  7 26 17 18  5 15           2      0  910   76    0    0   67
 C3: 20 30 19 29  9  8           3      0    1 1441    0    0  143
 C4: 14 25 16  6                 4      1    0    0  288    0  139
 C5: 10 23 12 13 24              5      6    0    0    0  780  167
 C6:  4 27 28                    6      0    1    2    2    1  285
```

Table 3-8 The cell size is limited to six

```
Total Inter-cell Flow: 583                     Inter-cell Flow Chart
 NPT Between Cells : 7                              1    2    3    4
 C1:  3 22  1 11  2 21 10 23 12              1   2932    0    0  297
 C2:  7 26 17 18  5 15                       2      0  910   76   67
 C3: 20 30 19 29  9  8                       3      0    1 1441  143
 C4:  4 27 28 14 25  6 16 13 24              4      3    1    2  908
```

Table 3-9 The cell size is limited to nine

For the problem with 30 machines and 41 parts, the computation time (excluding data I/O time) is between 0.37 and 0.44 milliseconds on a personal computer with a 133MHz Pentium™ processor. Computation time generally is not comparable because different computers have been used and normally a workstation runs faster than a PC. To give an approximate idea of time, a simulated annealing approach uses 11 to 17 seconds to get a solution of a problem with 16 machines and 43 parts (Sofianopoulou 1997).

	machine number																														
	20	30	19	29	9	8	7	26	17	18	5	15	3	22	1	11	2	21	10	23	12	13	24	16	6	14	25	28	27	4	vol.
P1	1	2	3	4		5																									115
P3	1	2	3	4	5																										120
P21		1	2	3	4																										84
P9				1		2																									61
P13				1	2																										127
P22				1																											120
P30				1																											55
P29			1			2																						3			61
P14						2				1																					76
P16							1	2	3	4																					87
P27							1	2	3	4																					97
P34							1	2		3	4																				60
P36							1	2	3		4																				49
P17											1	2																			31
P37												1																			81
P7									1																					2	67
P32													1	2	3		4	5													31
P33															1	2	3	4													93
P10														1	2	3	4														8
P41															1	2	3														31
P31													1	2	3			4		5											87
P23													1	2					3		4										78
P40													1	2				3			4										110
P39													1	2	3	4	5		6	7	8										84
P12													1	2	3		4	5	6	7	8										144
P11																	1				2										63
P18																	1									2					96
P2																			1	2											16
P20																					1										136
P19																					1	2									110
P38																						1	2								120
P6																								2	1						71
P26																								2		3				1	139
P4																										1	2				78
P5																										1					91
P8						1																						2	3		82
P35																												1		2	128
P15																													1	2	75
P25																						2								1	76
P28																														1	69
P24																					1									2	91
	20	30	19	29	9	8	7	26	17	18	5	15	3	22	1	11	2	21	10	23	12	13	24	16	6	14	25	28	27	4	

Table 3-10 The result chart of the problem 3

3.4 Discussion of Seed Selection and Searching Strategy

This section will discuss the impact of seed selection upon the cell being clustered and the issue of searching strategy which influences the efficiency of the algorithm and the quality of the results.

3.4.1 Selection of Seed Machine

It was observed in problem 3 that the seed machine played an important role in determination of natural cell size. Selecting the machine with a high flow as the seed would sometimes form a smaller cell and conversely a low-flow machine selection would form a larger cell. The algorithm using machines with minimum flow as the seeds, with the cell size limitation of 6 machines, directly gave the same solution as the results by cell merging (shown in Table 3-8). In contrast, for the problem of the eight-machine system in Figure 3-5, the selection of one of the machines M3, M4, M5 or M6 as the seed machine would lead to the solution of forming a single machine cell, which includes the whole system. This happens because these machines have the largest flow ratio between them. Subsequent allocation of all other machines will therefore be merged into the current cell. However, the seed selection of any machine in the system, other than these machines, will not cause the above problem. Thus, selection of the seed machines cannot depend on the total flow of a machine alone. Before the problem of the seed machine selection can be solved, an exploration procedure is needed in order to find proper natural cells.

3.4.2 Searching Strategy

A searching strategy for the natural cells concerns itself with the effectiveness of the algorithm. Three search strategies will be analysed in this sub-section. They are random search, exhaustive search and comprehensive search.

3.4.2.1 Random Search

Random search selects a machine randomly as a seed from the remaining machines to form the next cell. This procedure repeats until all the machines are grouped. For any candidate machine, $(m-1)$ computer operations are needed to calculate the flow between the candidate and the current cell, F_{mc}, and the flow between the candidate and the other two sets, F_{mr}. For any selected seed, $(m-1)$ computer operations are also needed to calculate the flow F_{co}. When merging a candidate into the current cell, all remaining machines of k need to be evaluated in order to obtain the one with the largest flow ratio. The adding is repeated until no candidate, whose the largest flow ratio is greater than unity, can be found. After the current cell has been formed, a new seed is then selected. The procedure repeats until all machines have been clustered. Thus, $k\cdot(m-1)$ calculations are needed in order to add a machine into the current cell. In the whole cell-formation procedure, k reduces from $m-1$ to zero. Therefore, the total number of operations can be calculated, as shown below.

$$\sum_{k=1}^{m-1} k\cdot(m-1) = m\cdot(m-1)^2/2 \qquad (3\text{-}17)$$

That is, the computation time is $O(m^3)$.

The symbol $O(m)$, a mathematical term used in algorithm analysis, means the total number of computational operations in the algorithm do not exceed a linear function of m. Expressed more precisely in mathematical language, a function $f(n)$ is $O(g(n))$ if there are positive constants c_1 and c_2 such that $f(n)\le c_1\cdot g(n)+c_2$ for all value of n (Tarjan 1983).

The quality of the results from this searching strategy is on a random basis, due to the fact that seed machines are selected randomly.

3.4.2.2 Exhaustive Search for Natural Cells

In an exhaustive search for natural cells, the number of searches will be the product of the total number of machines, m, and m minus the number of machines in the first cell, if the size of all the first cells are same, and so on. In the case of only a simple cell in the shop, the exhaustive search enumerates every machine as the seed. Thus, the number of calculations would be $m^2\cdot(m-1)^2/2$. In the worst case, each machine forms a cell and every machine has to be selected as the next seed. Thus, the total number of solutions obtained is the factorial of the total number of machines. The number of operation to get a solution is equal or less than $m\cdot(m-1)^2/2$. Therefore, the total number of computer operations in an exhaustive search is $O(m^3 \cdot m!)$. Although it is a huge computation, the procedure, in this case, is repeatedly to find the same solution.

3.4.2.3 Comprehensive Search

A comprehensive search uses each machine as the seed to form the first cell and randomly selects the seeds to form the following cells until all machines are clustered. Thus the procedure repeats exactly m times as the random search does. Therefore, the computation time is $O(m^4)$.

The comprehensive searching strategy gives every machine the possibility to be the seed machine and covers the cells formed by all possible seeds with a relative small amount of computation. The strategy gives a high opportunity to find the global optimum, though it is not guaranteed. Therefore, this search strategy is highly recommended.

The advantage of the strategy is that many solutions can be found in a run of the programme, though most of them may be identical, and the designer can easily pick the most suitable.

3.5 Conclusions

A manufacturing cell formation method has been developed in this chapter. The cells clustered by the method are represented in a more natural way than by the previous methods in which the specified number of cells need to be given and the number of machines in a cell is constrained by a pre-specified upper limit. The cells grouped by the method show the natural basic property of the relationship between machines and parts. If each cell is regarded as a machine, the size of the machine-clustering problem could be dramatically reduced by the NCE method. Consequently, the NCE method could be used to solve problems with a large number of parts and/or machines.

Tests showed that the NCE method is effective in minimising intercell material movement. It was observed that most solutions derived from different seeds were identical, which indicates that the method still has potential efficiency to be exploited by careful selection of seed machines; a topic which needs further study.

The heuristic described in this chapter is far from all-powerful or universal, but it is able to deal with some circumstances, which other approaches have failed to handle. Hence, the method is a supplement to the existing algorithms. The weight parameter α for calculating flow between machines is determined by the cell designer in the approach introduced here, although it should be determined by the influence of the involved costs. The impact of the weight parameter α on the cell formation and the improvement of the search quality also need further study.

Unfortunately, the research presented in this chapter did not include problems such as job assignment and machine duplication. Ho and Moodie (1996) provided a very good representation for job assignment and alternative routing. The machine duplication problem could be solved by utilising the concept of complex nodes proposed by Wu (1998). The minimisation of investment and machine utilisation could be performed during the merge of duplicated machines and cells. It is suggested that this could be a topic of further research.

Chapter Four

Cell Layout

In this chapter a particular layout form – single-roadway layout, which is the most commonly used within machine cells, is studied. This layout can be represented as a single-row layout, a double-row layout or a single loop (U-shaped) layout, in which the main material handling system (MHS), supplementary material handling equipment (MHE) and the flow direction are also taken into consideration. Road distance is applied in the evaluation of the layout and an order-based genetic algorithm is used to optimise a multi-objective to deal with different types of material flows and material handling systems.

4.1 Introduction

In the design of cellular manufacturing systems, the facility layout problems described in the literature can be divided into two classes: block layout problems and detailed layout problems (Meller and Gau 1996a). A block layout problem concerns itself with centroid to centroid distances between the departments and the areas of the departments, while ignoring the aisle structures, pick-up/drop-off points and the exact shape of the departments. The aisle structures, etc. are only of concern in the detailed layout problem. As discussed in Chapter 2, without considering the aisle structure, any method for designing a manufacturing layout is inadequate.

In the current research, a different classification is used, which depends on whether an aisle structure is given or not. In this thesis, a cell layout is regarded as the layout with given aisle structure, because the material handling system and the aisle structure within a cell usually have been determined before the design of the layout is carried out. A plant layout is the layout whose aisle structure needs to be designed instead of a pre-determined pattern. The concurrent design of layout and aisle structure will be discussed in the next chapter.

The simplest structure of a layout is a row layout, in which all departments are arranged into a single line. The problem of designing row layout is known as the row layout problem (RLP). Although it looks much simpler than a QAP or a cell formation problem, a row layout problem with the objective of minimising the total backtracking distance is still NP-complete (Kouvelis *et al* 1995). Therefore, there is no algorithm available to ensure the optimum solution within polynomial time. Much work has been done in the past two decades and various methods have been used for attempting to solve the RLP (Hassan 1994). Sarker *et al* (1994) studied the measures in the row layout problem by assuming that the widths of the machines are equal, i.e. the travel distance from one machine to the next are the same, a somewhat unrealistic assumption.

Heuristics are a type of commonly used approach to solve facility layout problems. Heragu and Kusiak (1988) developed a heuristic for RLP by determining the adjacency of machines according to the flow density. Their method can only be used for bi-directional MHE because a symmetric flow matrix is applied in the heuristic. Braglia (1997) extended the heuristic of Heragu and Kusiak to more general RLPs by adopting a known heuristic for the flow shop scheduling problem. Kouvelis *et al* (1995, 1996) studied the row layout problem and developed heuristic algorithms to minimise the total backtracking distance of material handling devices by the assumption of the equal distance between adjacent locations. Tang and Abdel-Malek (1996) used a three-phase hierarchical method to generate layout for manufacturing cells by a flow-network-oriented approach. In the hierarchical method, the flow network of all cells was firstly transferred into a master flow network. Then the flow pattern and the aisle structure were decided according to the master flow network. Finally, cells and storage areas were located and allocated, to complete the layout. In studies of Tang and Abdel-Malek, it is can be seen how important the aisle structure is in a manufacturing layout design. Benson and Foote (1997) also emphasised that aisle distances should be used.

Genetic algorithms were also used for layout problems with given aisle structures. Some studies showed that GAs are suitable for such problems, especially to find a sufficiently good solution instead of a guaranteed global optimum (in fact, no algorithm available can guarantee a global optimum in polynomial time). For example, a genetic search was found superior to a random search in a 32-cell problem (Banerjee *et al* 1997). Banerjee and Zhou (1995), and Banerjee *et al* (1997) used genetic algorithms to solve cell layout problems for a material flow path with a single loop. They decomposed the problem into two sub-problems: optimising the sequence of traversal of stations based upon inter-station flow and then optimising the layout according to the optimised sequence. The problem with this method is that the final layout solution depends on the optimal sequence and it cannot be proved that the optimal sequence will result in the optimum layout. Cheng *et al* (1996b), Cheng and Gen (1998) also tried to solve the loop layout problem by genetic algorithms. They concentrated on minimising the traffic congestion, which was expressed as the total number of times the parts traversed the loop. Delmaire *et al* (1997) applied linear programming within genetic algorithms, using fixed aisle skeletons including spine, T and O shapes. The ordering of the cells was given by the genetic process and the linear programming optimised the dimensions and positions of the input/output stations for the given ordering of the cells.

The selection of material handling systems also has a notable impact on layout design. Aneke and Carrie (1986) reported four different movements in a flow line: repeat, in-sequence, bypassing (skipping) and backtracking (Figure 4-1). The impact of the MHE effects the following aspects:

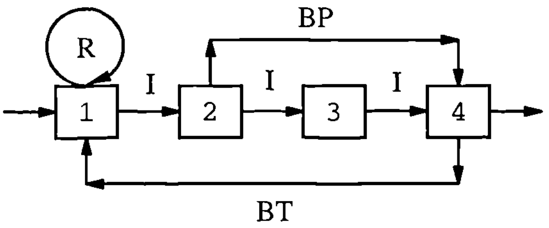


Figure 4-1 Four different flow movements in a flow line: repeat (R), in-sequence (I), bypassing (BP) and backtracking (BT) (Aneke and Carrie 1986)

a) The direction of material movement

If a one-direction conveyer is selected as the main transport, the machines in the layout should be so arranged that backtracking flows are minimised, ideally zero, because the conveyer cannot handle backtracking. If backtracking cannot be eliminated, additional transport must be provided or the intercell transport system be used. Either could be costly. In contrast, it is not a problem in a shuttle AGV system, which can handle materials in two directions. However, an AGV has its own limitations.

b) Material handling distances between two machines

The road distance from machine *i* to machine *j* in a one way loop generally differs from the road distance from machine *j* to machine *i*. In a bi-direction case, the distance between two machines is always the same.

c) The area of material handling systems

In most circumstances the area occupied by the material handling systems cannot be ignored. A roadway in the layout may affects the physical relationship between machines, such as distance and material handling behaviours.

d) The material handling cost rate

Investment and running cost change with the selection of the type of transportation system, which may be used to deal with the different types of flows. In practice, the handling of the in-sequence flows from one machine to the next in an AGV or track car system is often performed by feeding mechanisms instead of using the main transport system. Such devices are used to reduce the work-in-progress (WIP) and avoid frequent loading and unloading. In a one direction conveyer system, as previously mentioned, backtracking cost are normally much higher than the cost to handle the forward flows. Therefore, in layout design different types of flows should be treated separately.

This chapter focuses only on the single roadway layout problem within a manufacturing cell in which there is only one roadway for material handling. A single roadway layout is the most common layout form for a machine cell in cellular manufacturing. In a single roadway layout, four types of flows between machines can be observed: in-sequence flows, skipping flows, backtracking flows and crossing-road flows. Each type of flow may be dealt with by a different type of material handling system. Crossing-road flows may occur only in a layout of double rows of machines.

4.2 Single-roadway Cell Layout

A single roadway layout may include a single roadway, in which one row of machines is arranged on one side of the roadway, or two rows of machines on both sides, with, perhaps, some supplementary material handling equipment. The roadway may be folded to form an L-shaped layout or looped into a U-shaped layout.

4.2.1 Assumptions

It is assumed that the machines and the material flow between the machines, within a cell, have been determined. There is one in-coming buffer and one out-going buffer within the cell and all parts to be processed in the cell will go through the buffers before or after processing within the cell. Therefore, the dimensions of the machines and the magnitude of the flow are known.

In this study, it is also postulated that there is only one aisle within a cell and every machine can be accessed from the aisle. The dimension of a machine along the length of the aisle is denoted as the width of the machine and the dimension from the front to the rear of the machine as the length of the machine (see Figure 4-2). The pick-up and drop-off points are at the middle of the width on the front line of the machine, which can be extended to any point along the width. In this way, the layout design is simplified to a one-dimensional problem. The orientation of a machine is so arranged that the material flow through the machine is parallel to the aisle; that is, the front of the machine faces the aisle.

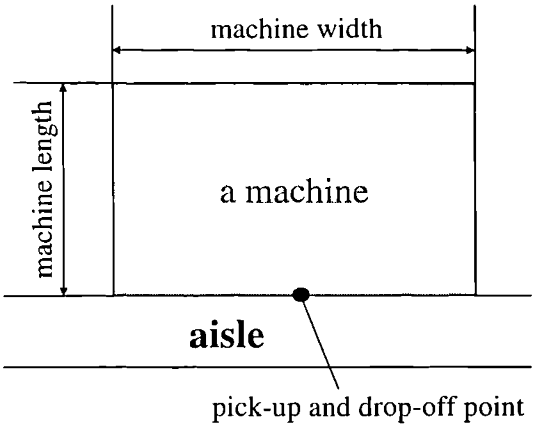


Figure 4-2 The orientation of a machine

4.2.2 Single Roadway Cell Layout

Three typical types of manufacturing cells are studied: single-row layout, double-row layout and U-shaped layout. In a single row problem, all the machines are queued into a line on one side of the aisle (Figure 4-3). In a double row problem, the machine line is folded back on the other side of the aisle (Figure 4-4). The in-coming and out-going buffers are put at the ends of the line. A conveyer or a shuttle car is often used as the main transport in a single or double row layout. The problem is a bi-direction problem

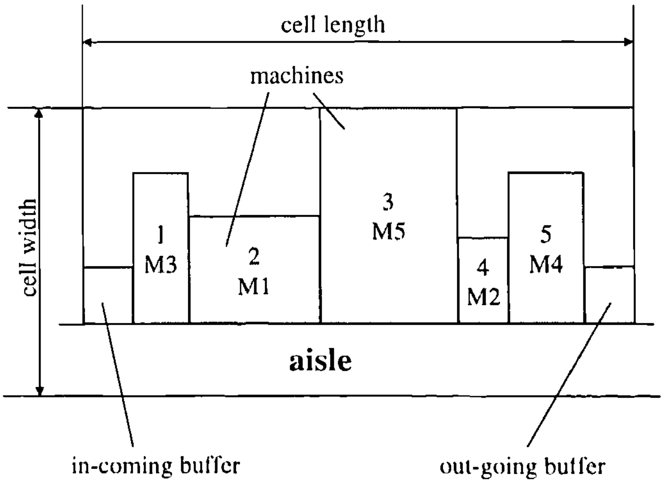


Figure 4-3 A single-row layout

when the main transport is a shuttle car type, or a uni-direction problem if a conveyer system is used.

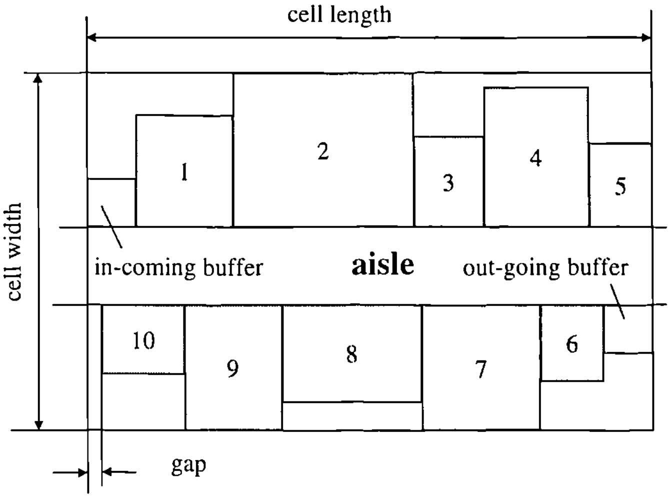


Figure 4-4 A double-row layout

A U-shaped layout can be viewed as a folded double-row layout. In a U-shaped layout, the aisle within the cell is a closed loop, some machines are put in the central rectangular area of the loop and the machines on the outside of the aisle are arranged to form a U-shape. The in-coming buffer and the out-going buffer are positioned at the ends of the U-shape, beside the main roadway (Figure 4-5).

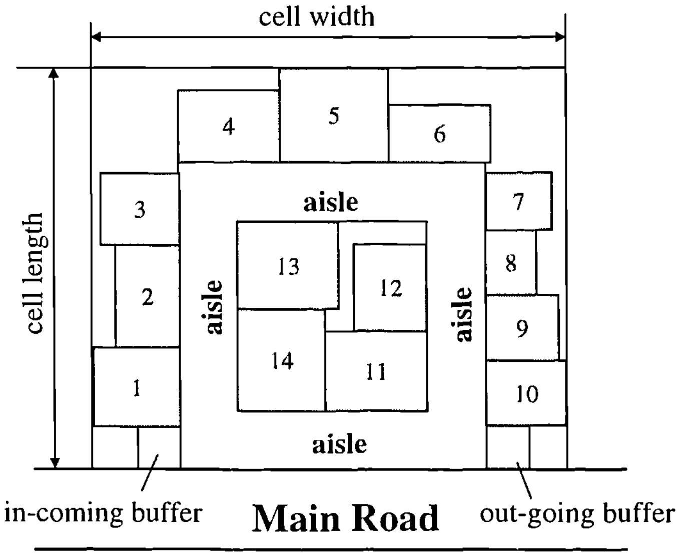


Figure 4-5 A U-shaped layout

4.2.3 Intracell Flow

Due to the different material handling equipment used within each machine cell, the flow volumes for the equipment of a cell need to be calculated by the following equation, which is similar to the equation for computing the intercell flows in Chapter 3.

$$f_{kl} = \sum_{i=1}^{n} \sum_{j=1}^{s_i - 1} v_i \cdot x_{ij} / \rho_i' \tag{4-1}$$

where f_{kl} is the volume of the mass flow between machine k and machine l.

x_{ij} represents the operation sequence expressed mathematically as:

$$x_{ij} = \begin{cases} 1 & if\ k = mc_{i,j}, l = mc_{i,j+1} \\ 0 & else. \end{cases}$$

k, l are indexes of machines,

k, $l = 0$ is denoted as the raw-material warehouse, and

k, $l = m + 1$ is denoted as the finished product warehouse.

i is index of parts.

n is the total number of part types.

j is index of process sequences.

s_i is the number of the operations within the process sequences for part i.

v_i is the volume of part i.

ρ_i' is the transportation lot size of part i within the cell.

mc_{ij} is the machine number visited by part i in operation j,

$mc_{ij} = 0$ is denoted as the raw-material warehouse, and

$mc_{ij} = m + 1$ is denoted as the finished product warehouse.

m is the number of machines.

The flow volume from the in-coming buffer to the machines in the cell and the flows from the machines to the out-going buffer are also calculated by the equations as follows.

$$f_{\phi k} = \sum_{\substack{i=1 \\ i \notin C}}^{m+1} f_{ik} \quad \text{and} \quad f_{k\Theta} = \sum_{\substack{i-1 \\ i \notin C}}^{m+1} f_{ki} \tag{4-2}$$

where $f_{\phi k}$ is the flow volume from the in-coming buffer to machine *k*.

$f_{k\Theta}$ is the flow volume from machine *k* to the out-going buffer of the cell.

C is the set of machines within the cell.

4.2.4 Aisle Distance

As addressed by Ho and Moodie (1998), some distance between neighbouring machines is necessary for the safe operation, the maintenance, the storage of work-in progress (WIP) or the feeder attached to the machine. Therefore, the down-flow travel distance from machine *i* to machine *j* could be computed as,

$$d_{ij} = \sum_{(k,l)\in R_{ij}} (D_k + D_l)/2 + \max\{dc_k, dp_l\} \tag{4-3}$$

where D_k and D_l are the widths of machine *k* and machine *l* respectively.

R_{ij} is the set containing every pair of the adjacent machines along the route from machine *i* to machine *j*.

dc_k is the minimum safe distance on the input side of machine *k*.

dp_l is the minimum safe distance on the output side of machine *l*.

$dc_k = dp_k$ for all *k* within the cell, if bi-directional MHE is used.

The backtracking distance depends on the directional character of the MHE. If a bi-directional device serves the cell, the backtracking distance is equal to the down-flow travel distance. That is,

$$d_{ji} = d_{ij} \tag{4-4}$$

When uni-directional equipment is applied, backtracking has to travel a much longer distance. Firstly, the backtracking parts are carried to the out-going buffer in the down-flow direction. Then, the intercell MHE has to be used to transport them back to the in-coming buffer, and finally, they are sent to the destination machine. In this case, the backtracking distance will be computed by the following equation.

$$d_{ji} = L_c - d_{ij} + p \cdot L_c = (1+p) \cdot L_c - d_{ij} \tag{4-5}$$

where d_{ji} and d_{ij} are the backtracking distance and the down-flow travel distance between machine *i* and machine *j* respectively.

L_c is the length of the cell.

p is a penalty parameter, $p \geq 0$, whose value depends on the ratio of intercell and intracell transportation cost and the real distance travelled from the out-going buffer to the in-coming buffer.

In a double-row layout the distances between machine *i* and machine *j*, d_{ji} and d_{ij}, are calculated in the exactly same way as in a single-row layout, if machine *i* and machine *j* are on the same side of the roadway. When machine *i* and machine *j* are separated in different rows, d_{ji} or d_{ij} is computed as the difference in the positions along the middle line of the roadway plus the width of the road. When the roadway is folded into a shape rather than a line, such as a loop in a U-shaped layout, the distance is formulated according to the above principle.

4.2.5 Transportation Cost

The transportation cost includes material moving costs and pick-up and drop-off costs. As previously mentioned, material moving costs can be further divided according to different types of flows and they should be calculated separately and might have different weights depends on the part type, weight, size, and so on. It is supposed that the number of journeys is in proportion to the flow volume. The general objective function is used to minimise the total transportation cost. The mathematical model is given as follows:

$$\text{Minimise} \left\{ \sum_{f_q \in F} f_q \times (a_1 \times d_q + b_1) + \sum_{f_b \in F} f_b \times (a_2 \times d_b + b_2) + \sum_{f_s \in F} f_s \times (a_3 \times d_s + b_3) + \sum_{f_c \in F} f_c \times (a_4 \times d_c + b_4) \right\} \qquad (4\text{-}6)$$

where f_q, d_q are the in-sequence flow volume and the corresponding road distance.

f_b, d_b are the backtracking volume and the corresponding road distance.

f_s, d_s are the skipping volume and the corresponding road distance.

f_c, d_c are the crossing flow volume and the corresponding road distance.

F is the set of flows.

$a_1,\cdots,a_4$ are weights on the four types of flows respectively.

$b_1,\cdots,b_4$ are weights on the loading and unloading cost for each type of flow.

The road distance used in the above equation depends on roadway flow types, which are in turn decided by transport type. In this research, two types of roadway flow are presented: uni-direction, such as in a one direction conveyer system, and bi-direction, as in a shuttle car system. In a single-row or a double-row layout with a uni-directional roadway, backtracking has to be handled by other transports. That implies another roadway might be used for backtracking. Therefore, backtracking distance is calculated as the distance from the starting machine to the out-going buffer, then from the outgoing buffer to the in-coming buffer and finally from the in-coming buffer to the destination machine. In a U-shaped layout, uni-directional flow is always clockwise or anticlockwise, while the distance of bi-directional flow is calculated clockwise or anticlockwise, whichever is shorter.

4.3 A Genetic Algorithm for Single-roadway Layout Problems

As previously mentioned, GAs have potential power to solve facility layout problems efficiently. Thus, an order-based GA has been selected as the main algorithm for the solution of single roadway layout problems. To construct a genetic algorithm, there are some determining factors: string representation, fitness function, initial population, genetic operators and the parameters for the genetic evolution. These factors strongly affect the searching efficiency (Mavridou and Pardalos 1997).

4.3.1 String Representation

In genetic algorithms variables are represented as a string known as a genome. An order-based genome is frequently used in the application of genetic algorithms to the layout problem. The scheme of the string representation is problem dependent and not unique. In this research, the string order of an order-based genome indicates the locations and the numbers in the string represents the machine numbers. For example, [3 1 5 2 4] in a five-machine single-row cell represents machine 3 in location 1, machine 1 in location 2, and so forth.

The string to the double-row layout problem shown in Figure 4-4 is represented by [1 2 3 4 5 / 6 7 8 9 10]. There is a folding point at position 6 to convert the string into a double-row layout. The folding point is floating, depending on the dimensions of the machines in the string. The positions of the two buffers also float. The buffers and the folding point are so arranged that the length of the cell is minimised. A gap may exists in one of the rows, which allows the machines to be moved along the aisle in order to reduce the backtracking from or to the machines on the other side of the aisle with shorter distance than the gap.

There are four fold points in a string to represent a U-shaped layout. The U-shaped layout shown in Figure 4-5 (page 82) has four fold points proceeding machines at 4, 7, 11, and 13 respectively. The first two fold points indicate the two upper corners. The third fold point divides the machines into central block and the outskirts. The fourth point folds the central block into two rows. The determination of the four folds is based upon the idea that the area of the cell and the length of the road are near to the minima when the internal perimeter of the outskirts equals the perimeter of the central block and the road. Whenever the perimeter of the outskirts is greater than that of the central block, a machine in the outskirts is moved into the centre, as shown in Figure 4-6. The central block is optimised as in the double-row layout. The details of determining the folding points are described in the following procedure.

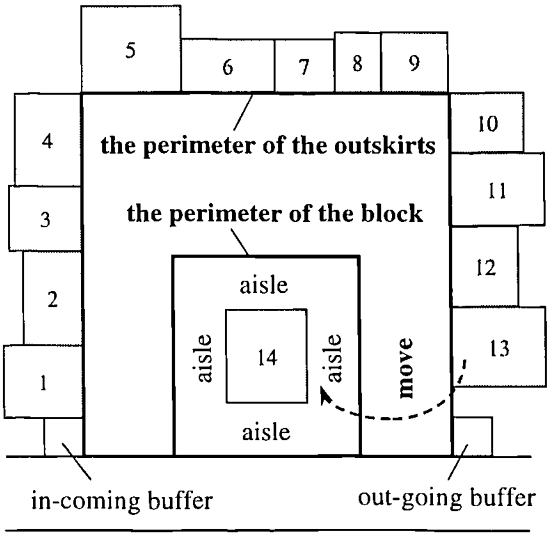


Figure 4-6 The determination of the central block

Step 1. Put the last machine, which is temporarily regarded as the third folding point, in the string into the central block.

Step 2. If there are two or more machines in the central block, decide the fourth folding point as in the case of a double-row layout, but ignoring buffers.

Step 3. Calculate the perimeter of the central block and the width of the machines from the first position to the third folding point in the string.

Step 4. If the width plus the widths of the buffers is larger than the width of the three side of the central block plus aisle widths, move the last machine to the third folding point into the central block and go to step 2, else go on to step 5.

Step 5. Decide the first and second folding points according to the central block and the aisle.

The merit of this form of the representation will be discussed later in Section 4.4.

4.3.2 Genetic Evolution

The evolution of a genetic algorithm (GA) starts with an initial population. The individuals with higher fitness values (better solutions or lower value of the objective function) have better chances of being selected as parents to generate offspring for the next generation than those with lower fitness values. The strong have a high probability to live while the weak have high probability to die off. Parents produce children by a crossover mechanism to let the children inherit good genes. To prevent relative propagation that may result in a local optimum, a mutation mechanism is applied to produce completely different chromosomes (strings) within the population.

Crossover is the most influential operator with respect to algorithm efficiency and solution quality. There are three commonly used crossovers for an ordinal chromosome: partially matched crossover (PMX), order crossover (OX) and cycle crossover (CX). For example, the following illustrates the PMX procedure in which two parents, P1 and P2, generate two children, C1 and C2, with two cut points marked by "|" and illegal genes underlined.

P1 4 8 3 | 5 1 9 | 6 7 10 2

P2 2 4 7 | 6 10 8 | 9 3 1 5

The ordinary swap builds intermediate children.

I1 4 $\underline{8}$ 3 | 6 10 8 | $\underline{6}$ 7 $\underline{10}$ 2
I2 2 4 7 | 5 1 9 | $\underline{9}$ 3 $\underline{1}$ $\underline{5}$

The ordinary exchange of sub-genes bounded by cut points produces illegal strings of repeated genes as shown in the above intermediate children. In these circumstances the repeated genes located outside the cut points are replaced by a mapping mechanism. The illegal genes are replaced by the following mapping between the two cut points.

5 ⇔ 6, 1 ⇔ 10 and 9 ⇔ 8

Therefore, the repaired strings are:

C1 4 9 3 | 6 10 8 | 5 7 1 2
C2 2 4 7 | 5 1 9 | 8 3 10 6

The repaired strings can be used for calculation of the objective function.

OX creates offspring by keeping the original sequence of the parents. The mechanism is explained as follows.

P1 4 8 3 | 5 1 9 | 6 7 10 2
P2 2 4 7 | 6 10 8 | 9 3 1 5

After swapping the segments between the cut points, the children have the following schemes, in which an "X" indicates an undetermined gene.

C1 X X X | 6 10 8 | X X X X
C2 X X X | 5 1 9 | X X X X

Removing 6, 10 and 8 from P1, and 5, 1 and 9 from P2, the remains are:

P1 4 3 5 1 9 7 2
P2 2 4 7 6 10 8 3

The above segments replace the "Xs" of the children, C1 and C2.
Therefore, the offspring are finally generated as follows.

C1 4 3 5 | 6 10 8 | 1 9 7 2
C2 2 4 7 | 5 1 9 | 6 10 8 3

CX generates the children in such a way that each gene comes from one of the parents while preserving the absolute position of the genes in the parent sequence. The procedure starts with copying the first gene of the parent P1 to the first child.

P1	4 8 3 5 1 9 6 7 10 2
P2	2 4 7 6 10 8 9 3 1 5
C1	4 X X X X X X X X X

Next, the gene in P1 which is the same as the gene at the corresponding position in P2 is copied to the child.

C1	4 X X X X X X X X 2

Repeat the above operation until no new gene can be copied. That is,

C1	4 8 X 5 X 9 6 X X 2

Then, the remaining undetermined genes are copied from P2.

C1	4 8 7 5 10 9 6 3 1 2

The same operations are used to obtain the second child.

Therefore, the two children are,

C1	4 8 7 5 10 9 6 3 1 2
C2	2 4 3 6 1 8 9 7 10 5

The different crossovers described above have appeared in the literature for solving different problems. It was reported that OX has the best performance for the travelling salesman problem (Oliver *et al* 1987). However, according to Chan and Tansri (1994) the PMX was the best among the three ordinal crossovers for facility layout problems. Therefore, the PMX was selected for this research.

The fitness of the children strings are then evaluated by the objective function discussed in the previous section. The individuals of the next generation are selected according to their fitness. The evolution continues until a pre-specified number of generations (or convergence) has been reached.

4.4 Computational Experiments and Discussion

The computer programme for this approach was written in C++ on the platform of a PC compliant computer with a Pentium™ processor. The main body of the GA program was adopted from Matthew's GAlib (Wall 1996). The parameters for genetic search were used as follows: the mutation probability was 0.005, the crossover probability 0.9, the replacement percentage 0.20. The initial population was selected randomly.

4.4.1 Single-row and Double-row Layout

The experimental problems for uni-directional single-row layout are taken from Problem 3 in Chapter 3. Table 4-1, Table 4-2 and Table 4-3 give the results of 8-cell, 6-cell and 4-cell solutions respectively, while the second cell-dimension in Table 4-3 is the dimension of the single-row U-shaped cell and the slash in the machine sequence is represented as the fold point.

Cell No.	Machine Sequences in the Row	Cost $\times 10^3$	IS	SP	BT	Dimensions (length×width)
1	3 22 1 11 2 21	147.6	2443	1023	0	225 × 60
2	10 23 12	92.9	1610	778	0	130 × 60
3	7 26 17 18	53.5	1323	319	0	175 × 55
4	5 15	10.4	283	190	0	60 × 40
5	20 30 19 29 9 8	146.4	1834	1409	0	190 × 60
6	28 27 4	103.5	1073	788	0	155 × 45
7	6 16 14 25	66.7	437	801	0	155 × 55
8	13 24	17.6	546	186	0	80 × 50

Where IS is the summation of the in-sequence volume,
SP is the summation of the skipping volume,
BT is the summation of the backtracking volume

Table 4-1 Uni-directional single-row layouts of the 8-cell solution

Cell No.	Machine Sequences in the Row	Cost $\times 10^3$	IS	SP	BT	Dimensions (length×width)
1	3 22 1 11 2 21	147.6	2443	1023	0	225 × 60
2	10 23 12 13 24	194.1	1144	1866	0	190 × 60
3	6 16 14 25	66.7	437	801	0	155 × 55
4	28 27 4	103.5	1073	788	0	155 × 45
5	7 26 17 18 5 15	97.2	1206	800	0	215 × 55
6	20 30 19 29 9 8	146.4	1834	1409	0	190 × 60

Where IS is the summation of the in-sequence volume,
SP is the summation of the skipping volume,
BT is the summation of the backtracking volume

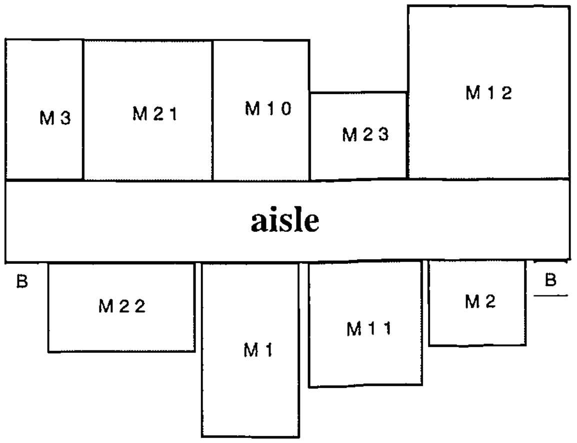
Table 4-2 Uni-directional single-row layouts of the 6-cell solution

Cell No.	Machine Sequences in the Row	Cost $\times 10^3$	IS	SP	BT	Dimensions (length×width)
1	3 22 1 11 2 / 21 10 23 12	338.6	3410	1878	0	335×60, 170×120
2	28 27 4 13 / 24 6 16 14 25	456.7	990	2626	0	350×55, 180×100
3	20 30 19 / 29 9 8	146.4	1834	1409	0	190×60, 95×105
4	7 26 17 / 18 5 15	97.2	1206	800	0	215×55, 120×100

Where IS is the summation of the in-sequence volume,
SP is the summation of the skipping volume,
BT is the summation of the backtracking volume

Table 4-3 Uni-directional single-row layouts of the 4-cell solution

No requirement for backtracking has been achieved in all of the above layout solutions, which is ideal for applying conveyers as the material handling equipment. However, the achievement of non-backtracking in a single-row layout does not always imply that the solution with non-backtracking in a double-row layout can be found. Figure 4-7 shows a solution for Cell 1 in Table 4-3 with the minimised material handling cost. The computer programme also found a non-backtracking solution for the single-row layout of the 30-machines problem in Problem 3 within a 5-minute running time. This indicates the programme is capable of solving medium sized problems.



Cost:	243.4
In-Sequence:	2767
Skipping:	1297
Crossing:	872
Backtracking:	352

Figure 4-7 A double-row solution

4.4.2 U-shaped Layout

The problem for U-shaped layout was taken from Banerjee and Zhou's work (Banerjee and Zhou 1995). The best results obtained from the GA for uni-directional and bi-directional flow are given in Figure 4-8 and Figure 4-9 respectively. In comparing the results in this research, the solutions from Banerjee and Zhou's work ignored the accessibility to the cell and did not consider the road width. In fact, the road may take a significant amount of area, which could have a great effect on the relative location of the machines and the layout.

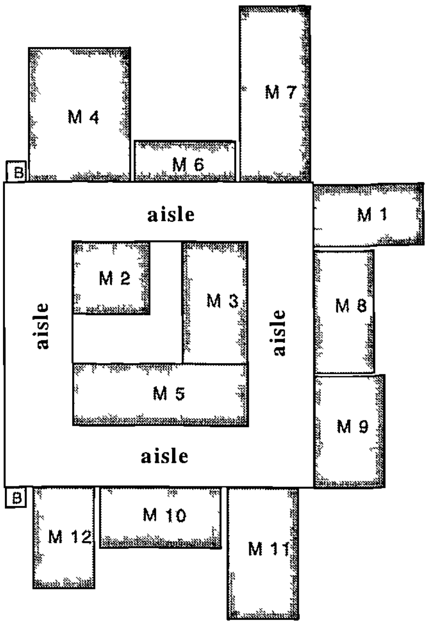


Figure 4-8 A U-shaped layout solution for uni-directional aisle

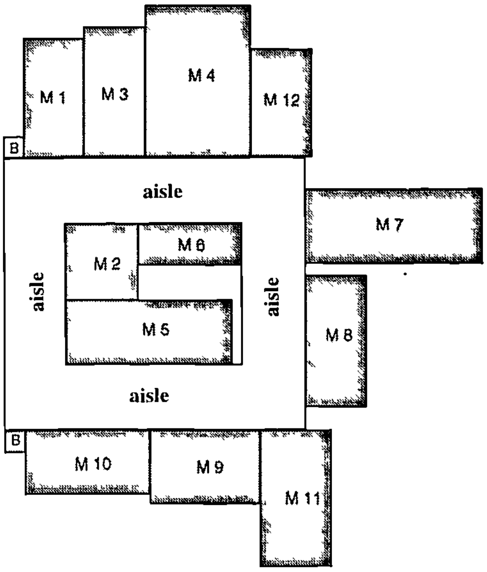


Figure 4-9 A U-shaped layout solution for bi-directional aisle

4.4.3 Efficiency of the Genetic Search

Two presentations of GA strings for double-row layout are compared in this research. One of them folds the string as described in the previous section and the other restarts the second row from the cutting point. An example of the two representations for a ten-machine problem is shown in Figure 4-10.

String 1

1	2	3	4	5
10	9	8	7	6

String 2

1	2	3	4	5
6	7	8	9	10

Figure 4-10 An example of two presentations of the genetic strings

The 9-machine double-row problem of Cell 1 in Table 4-3 is again utilised to examine the efficiency of the genetic algorithm. By an exhaustive search, the solution in Figure 4-7 has been proved to be one of the two global optima (the other optimum is the symmetry about the aisle).

The behaviour of two representations cannot be compared in a single run, because a genetic algorithm is on a stochastic basis. Thus, an optimum-hit rate was introduced for this purpose. Under the given GA parameters, the optimum-hit rate is the number of runs that achieve the optimum result in one thousand runs. According to statistics, optimum-hit rate is an estimate of the probability of reaching the optimum in a single run.

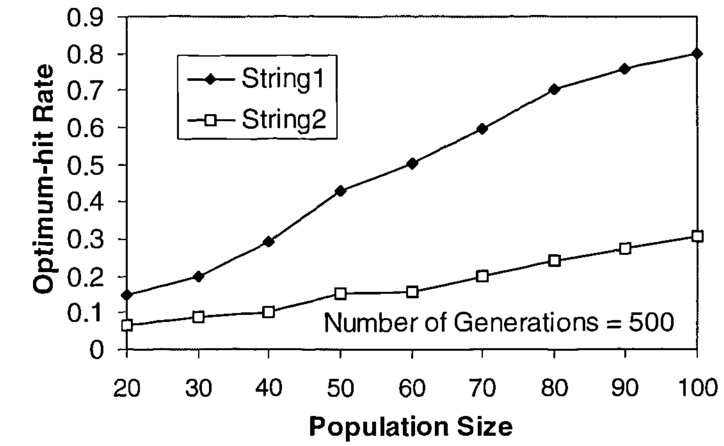


Figure 4-11 Optimum-hit rate versus population size

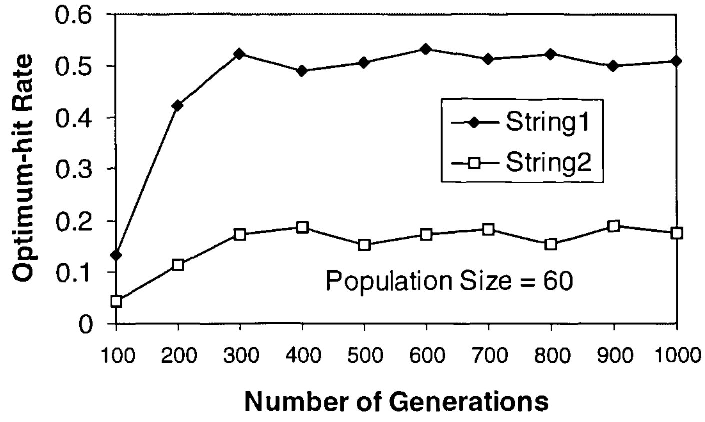


Figure 4-12 Optimum-hit rate versus generations

It was found that the efficiency of the GA presented by string 1 were much higher than that by string 2 as shown in Figure 4-11 and 4-12. It is believed that the interruption of neighbour location of the string causes the efficiency difference. In considering the ten-location strings in Figure 4-10, it would be helpful if the dimensions of the machines are assumed to be unity and the dimensions of the buffers and width of the aisle are omitted. In string 1, the distances from the incoming buffer to the machines are continuous (Figure 4-13), while the distances in string 2 are discontinuous (Figure 4-14). The interruption of the distance continuity causes the offspring to inherit less information from their parents, because the neighbour genes have more chance to be transferred to the children in the PMX.

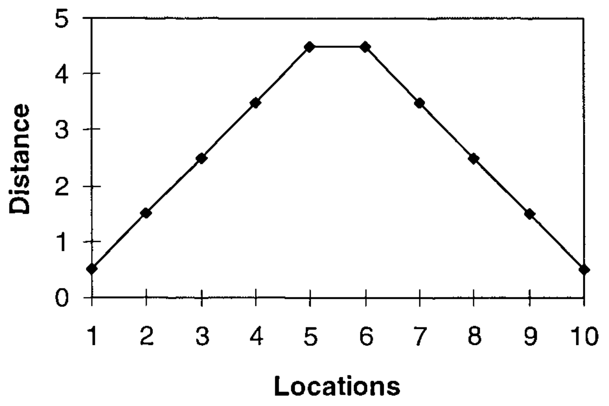


Figure 4-13 The distance of string 1

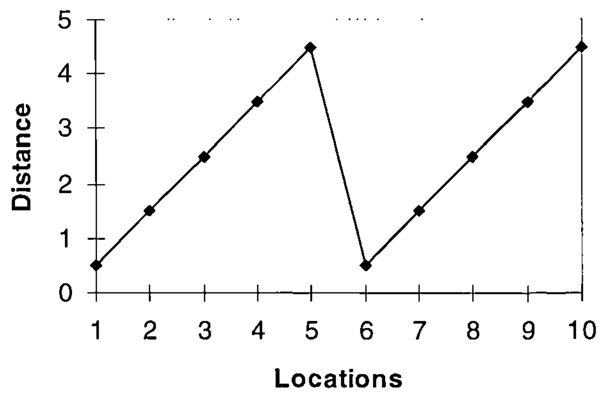


Figure 4-14 The distance of string 2

It was also found that for certain problems more generations do not significantly improve the behaviour of GAs. If the number of the generations is greater than 300 to 400 (Figure 4-12), the opportunity to obtain the optimum does not increase in this case. Further investigation showed that a large population size may increase the probability of catching the global optimum but at low levels of generations increasing population size makes things worse. However, the population size is limited by the system memory of the computer. The optimum-hit rate versus number of generations and populations is shown in Figure 4-15. The best number of generations is around 500 and the best population size is 115 (limited by the computer system) in the case of the problem.

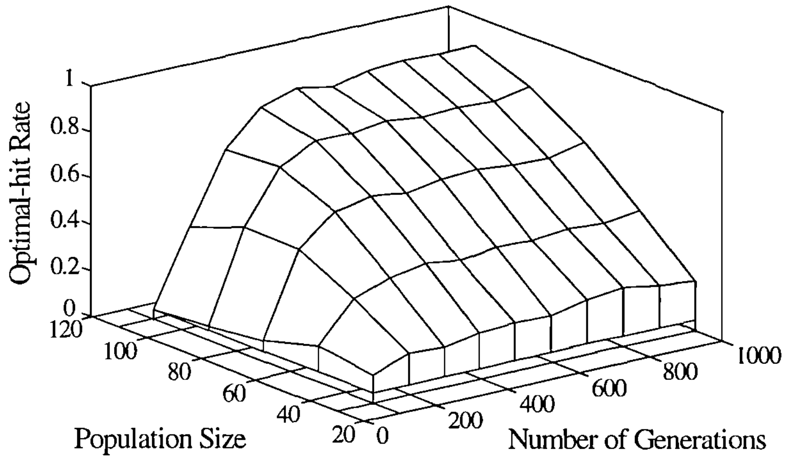


Figure 4-15 Optimum-hit rate of string 1

4.5 Summary and Conclusions

A genetic method is used for layout design for a single-roadway machine cell and an evolution model for the above problem has been developed. The model includes the simplest and the most common layout patterns, that is, single-row layout, double-row layout and U-shaped layout were investigated. Four types of material flow, produced by uni-directional and bi-directional material handling equipment, and by the main MHE and supplementary MHE, were also taken into account within the model. The model could be extended for other single-roadway layout forms, such as L-shaped layout.

A form of string representation for a genetic search was discussed. This representation reduces the discontinuation of neighbour genes. The computational tests show that it may significantly increase the search efficiency compared with the representation from earlier work. Although not discussed here, further studies may expand the order-based genome from a string to a 2D form to represent the 2D layout with less discontinuation and accordingly may need a 2D crossover.

Chapter Five

Plant Layout

The previous chapter discussed the design of a layout with a single roadway material handling system. In this chapter a more complicated and general layout form will be studied. This layout form will cover the layout forms from spine layout to matrix layout and simultaneously take the aisle structure into account. Although the title of the chapter indicates a focus on "plant layout", the methodology in this chapter can also be used in the design of machine layout within a cell.

5.1 Introduction

As discussed in Chapter 2, material-handling costs are the main concern with layout problems, while the dimensions of the departments or machines within the layout decide material-handling distances. Hence, due to the ignoring of the dimension differences of the departments, the quadratic model discussed in Chapter 2 is inadequate in most cases. With respect to the evaluation of the layout, rectilinear distance and Euclidean distance are also not satisfactory for the accurate measurement of distance, because all material handling is performed via aisles (three dimensional material handling layout is out of the scope of the research in this thesis). Therefore, a method is needed which gives consideration to unequal area departments and aisles.

Slicing layout, which will be discussed in detail in the next section, provides a good method for representation of the problem. It can represent unequal-area departments and its slicing cut-lines, which separate the departments, can be used for the aisles (Tretheway and Foote 1994). In comparison with non-slicing layouts, the aisle structure in a slicing layout is relative simple and it can represent the most common conventional aisle structures, including single-aisle layout, spine layout and matrix layout.

5.2 Preliminaries

Before studying a slicing layout, some assumptions need to be addressed. It is supposed that there are *m* departments with given dimensions to be placed in a rectangular shop floor of a plant. One end of the shop floor is next to a raw-material warehouse. The other end is near to a product warehouse. The surroundings of the shop floor are pre-defined as either roadways or walls. The roadways connecting the raw material warehouse and the product warehouse are called main aisles. The start points and the ends of main aisles are entrances and exits of the shop floor respectively. All raw materials or finished products are transported through the entrances or exits. The layout within the warehouses, that is, outside the shop floor area, is excluded in this research.

A *floorplan* is a partitioning of a rectangle into basic blocks which are also rectangles. This partitioning has the property that while horizontal and vertical lines may meet, but cannot cross (Chong and Sahni 1993). The basic blocks provide the spaces into which the machines may be placed.

A floorplan is *a slicing floorplan* if and only if either (1) it consists of a single basic block or (2) there is a line segment, or a cut-line, that divides the floorplan into two sub-floorplans such that each sub-floorplan is a slicing floorplan (Chong and Sahni 1993, Shi 1996). An example of a slicing floorplan is provided in Figure 5-1.

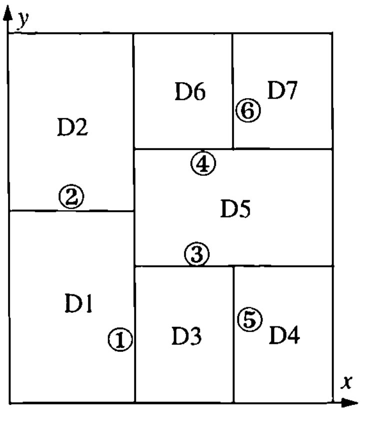


Figure 5-1 A slicing floorplan

A *vertex*, or intersection, in a floorplan is a point where a horizontal and a vertical line meet. Since horizontal and vertical lines do not cross in a floorplan, each vertex has exactly three links to other vertices, except for the corner vertices. In a floorplan, there are exactly four corner vertices each of which have two links to other vertices.

Each cut-line of the floorplan, except the borders, is denoted as being of a certain *level.* It is also supposed that, without losing generality, the slicing procedure starts from one or more vertical cut-lines. The vertical cut-lines from the lower border to the upper border of the floorplan are of level 1. The subsequent horizontal cut-lines are of level two and so on. The level of a cut-line is one level lower than the level of the cut-lines at the start point or end point whichever is lower. Thus, all cut-lines of odd level are vertical and all cut-lines of even level are horizontal. For example, cut-line ① in Figure 5-1 is of level 1. Cut-line ②, ③ and ④ are of level 2 and cut-line ⑤ and ⑥ of level 3. Cut-line ②, ③ and ④ are horizontal while the others are vertical.

A *block* of a certain level is a minimum sub-floorplan partitioned by a cut-line or cut-lines of the same level. For instance, the block composed of department 6 and 7 is a level-2 block and the block occupied by department 5 is also a level-2 block. Blocks of the same level do not overlap. That is, for example, a level-2 block cannot contain another block of level-2. The sub-floorplan composed by department 5, 6 and 7 in Figure 5-1 contains two blocks of level-2, instead of being a single level-2 block.

Therefore, a slicing floorplan can be described as a tree form, in which the shop floor is represented as the root of the tree. Each node of the tree represents a block at a certain level and the sub-trees represent the sub-blocks within the block. A parent block has at least two offspring blocks if there are any sub-blocks within the block. Each node without offspring, i.e. each external node, represents a department. The block tree of the floorplan of the example in Figure 5-1 is shown in Figure 5-2. Between any two adjacent blocks of the same level, there is a cut-line that has the same level as the blocks. A number in a pair of brackets represents a cut-line while the number gives the level of the cut-line. It can be noted that between two adjacent departments, if the departments are listed in the order of the external nodes, there is exactly one cut-line. The departments and the cut-lines can, therefore, be separated into a department string and a cut-line string without losing information for representation of a floorplan.

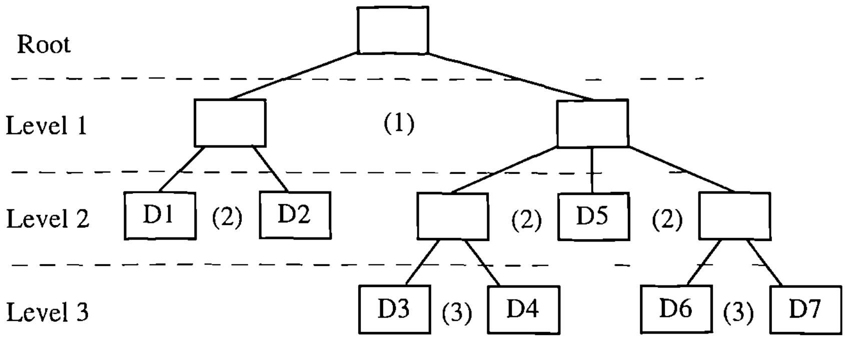


Figure 5-2 The block tree of the floorplan

The department string represents the sequence of the departments to be located. The cut-line string, which contains (m-1) integers if there are m departments in the shop floor, represents the relative position of the departments. The integer in the cut-line string represents the level of each cut-line. The departments are placed in the shop floor from bottom to top or from left to right according to the sequence of the departments and the level of the related cut-line. The pre-specified dimensions of each department decide the relative co-ordinates of the cut-lines in the shop floor to ensure each basic block provides sufficient space to accommodate the corresponding department. The details of the relative co-ordinates and the size of the floor plan will be discussed later in Section 5.4.1.

It is assumed that two delivery stations are provided for each department, one for dropping-off the materials to be processed within the department and the other for picking-up the processed products. They may be located at the front of the department (Figure 5-3), or located separately at the opposite sides of the department (Figure 5-4). The positions and the location type of the delivery stations for each department is given previously.

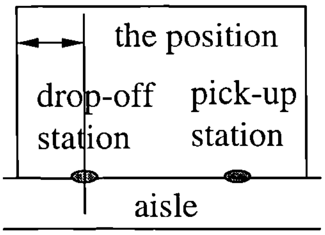


Figure 5-3 Pick-up/Drop-off stations located at the front of the department

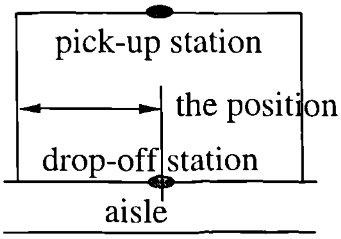


Figure 5-4 Separated Pick-up/Drop-off stations

The orientation of a department is represented by an integer in an orientation string. Each integer ranges from 0 to 3, which represents one of the four orientations of a department by clockwise rotation from 0° to 270° respectively, or the integer ranges from 0 to 7 if the reflection forms are taken into account. The orientation shown in Figure 5-3 and Figure 5-4 is defined as 0, which represent the drop-off station positioned on the lower side of the department.

Thus, a shop floor can be fully represented by six strings: a department string, a cut-line string, an orientation string, a cut-line position string, a station position string and a station type string. The last three are determined by previously given data, and hence the first three are used as design variables in this research. The design variables of example 1 are listed as follows.

Department series:	1	2	3	4	5	6	7
Cutting levels:		2	1	3	2	2	3
Orientation string:	0	0	0	0	0	0	0

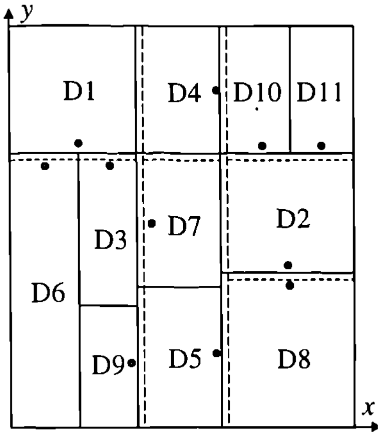


Figure 5-5 The floorplan of example 2

A more complicated example, which has two main aisles, is shown in Figure 5-5. The strings of the design variables of the layout are listed below and its block tree is illustrated in Figure 5-6.

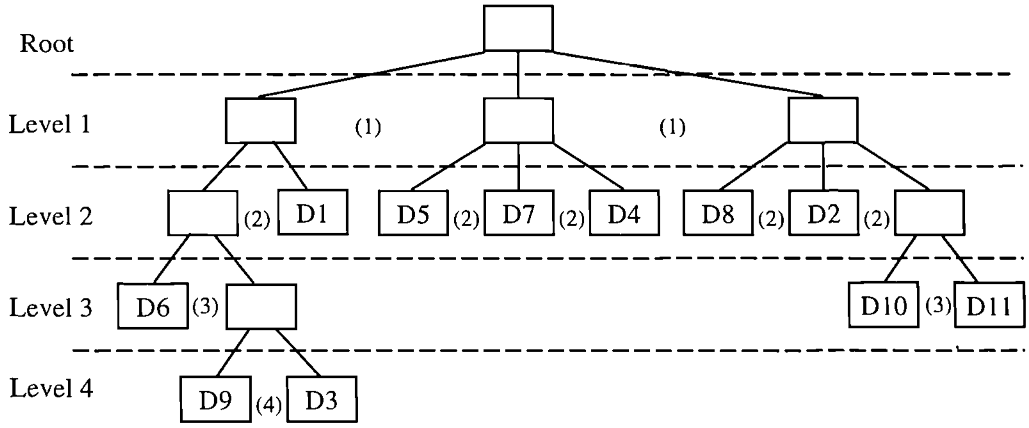


Figure 5-6 The block tree of example 2

Department series:	6	9	3	1	5	7	4	8	2	10	11
Cutting levels:	3	4	2	1	2	2	1	2	2	3	
Orientation string:	2	3	2	0	3	1	3	2	0	0	0

There are several benefits from the above representation:

1) Genetic algorithms (GAs) can easily adopt the strings for optimisation, although some rules are needed to prevent illegal forms.
2) One layout form has only one set of strings and one set of strings represent one layout.
3) The number of aisles and junctions at any level can be quantified and computed. Therefore, the complexity of the aisle structure can be evaluated.

The following sections present a methodology for layout optimisation based upon the above block-tree representation.

5.3 Proposed Methodology and Mathematical Models

This research proposes a two-stage method to solve the problem. In the first stage, or floorplan stage, the locations and the orientations of the departments are optimised with an objective of minimising the material handling cost. The second stage, aisle-structure stage, minimises the number of aisles based on the optimised layout from the first stage.

In the floorplan stage, the material flow system is represented as a three-layer system as illustrated in Figure 5-7. The first layer represents the logical flow requirements as defined by a directed graph $G_L(E,N)$. In the graph, N denotes the set of nodes of which each node represents the P/D stations of a department. E denotes the set of edges representing the flows between the departments. A weight f_{ij} is assigned to each edge (i, j) representing the flow intensity requirements from node i to node j in graph G_L. The third layer represents the possible aisles as defined by a planar graph $G_P(P,V)$, where P denotes the set of paths and V denotes the set of intersections. A weight d_{kl} is assigned to each path (k, l) representing the distance of the path from intersection k to intersection l in graph G_p. Every plane face in the path graph corresponds to a department in the middle layer. Each P/D station of a department links to two related intersections according to the orientation of the department with proper distances.

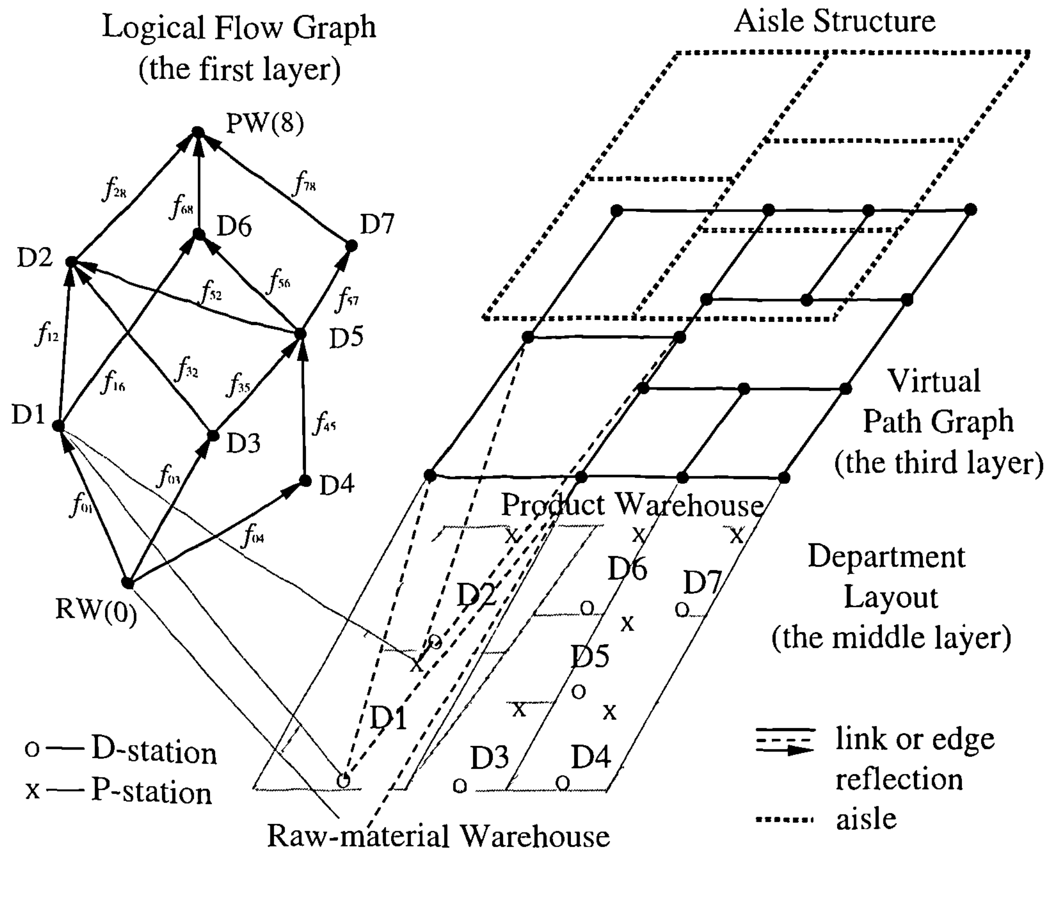


Figure 5-7 The aisle structure and material flow system

The optimisation of the floorplan is performed by a genetic algorithm, which provides the potential solutions to be evaluated. Then the floor layout is realised according to the solution strings and the corresponding path graph is built. The material handling cost is calculated as the objective function for the evaluation, in which the transportation distances are station-to-station distances through the shortest aisle path. The mathematical model of the floorplan is defined as follows.

$$\text{Minimise} \sum_{i=0}^{m}\sum_{j=1}^{m+1} f_{ij} \cdot d_{ij} \tag{5-1}$$

subject to

$$U_a \le \sum_{i=1}^{m} a_i / a^F \tag{5-2}$$

$$r_s \le \frac{L_f}{W_f} \le \frac{1}{r_s} \tag{5-3}$$

where m is the number of the departments,

i, j are index of the department,

$i = 0$ is denoted as the raw-material warehouse and,

$j = 0$ denoted as the product warehouse,

f_{ij} is the flow density form department i to department j,

d_{ij} is the shortest aisle distance from department i to department j,

a^F is the area of the floor.

a_i is the area of department i.

U_a is the area utilisation limit.

r_s is the floor shape limit.

L_f , W_f are the length and width of the floor respectively.

The model minimises the inter-department material handling cost, represented by the total volume distance. Constraint (5-2) ensures that the utilisation of the floor area is not less than a given limit. Constraint (5-3) makes the shape of the floor maintain a certain aspect ratio.

The second stage is the optimisation of aisle structure. The minimised number of aisles among the entire shortest paths is selected as the objective of the optimisation in this research. That is, for each existing flow, one path is selected among the entire possible shortest paths of the flow, provided that the number of the total aisles, of which the paths for all flow are composed, is the minimum. The mathematical model of the optimisation of aisle structure is given below.

$$\text{Minimise} \left| \bigcup_{i=0}^{m} \bigcup_{i=1}^{m+1} \{p_{ij} \in \boldsymbol{P}_{ij} : f_{ij} \neq 0\} \right| \qquad (5\text{-}4)$$

where m is the number of the departments.

i, j are indices of the department.

$i = 0$ is denoted as the raw-material warehouse and.

$j = 0$ denoted as the product warehouse.

f_{ij} is the flow density form department i to department j.

p_{ij} is the set of cut-lines for a path of $\boldsymbol{P}_{ij}$.

$\boldsymbol{P}_{ij}$ is a set of the shortest paths from department i to department j.

$| S |$ is denoted as the number of the elements in set S.

The set arithmetics refers to *Naive Set Theory* by Halmos (1960). The computer implication on details of the above two stages will be discussed in the following sections.

5.4 Floorplan

When a set of the design variables is given, it is necessary to realise the shop floor from these design variables for evaluation purpose. All cut-lines are regarded as virtual aisles during the floorplan stage. The transportation distance from a department to another department is calculated as the shortest path in the aisle network using graph theory.

5.4.1 Realisation of a Shop Floor

The task of realisation of a shop floor is to compute the position of each cut-line and the dimensions of the shop floor and sub-blocks according to the pre-specified dimensions of each department.

The realisation of a shop floor is based on the fact that each basic block should have enough room to accommodate the related department. Due to the constraints of the dimensions of the departments, the procedure is executed from bottom to top of a block tree. Two sub-blocks, which are basic blocks at the beginning, are placed horizontally or vertically according to the oddity or evenness of the cut-line between the two sub-blocks and are merged to form a parent block. The relative position of the cut-line is determined by the dimensions of the first sub-block. The virtual horizontal size of the parent block is the sum of the virtual horizontal dimensions of the sub-blocks and virtual vertical size is the greater of the two virtual vertical dimensions, if they are merged horizontally, and *vice versa*. The procedure is carried out until the dimensions of the root block, *i.e.* the dimensions of the shop floor, are determined. For example, to realise the floor layout of example 2 as shown in Figure 5-5, the departments are merged from the bottom to the root according to the tree representation in Figure 5-6. To compute the dimension of the parent block composed by D10 and D11, D10 is placed on the left of D11, because the level of the cut-line is odd, *i.e.* the cut-line is vertical. The horizontal dimension of the parent block is the horizontal dimension of D10 plus the horizontal dimension of D11 and the vertical dimension of the parent block is the greater of the vertical dimension of D10 and the vertical dimension of D11 as illustrated in Figure 5-8. The horizontal dimension and the vertical dimension of a department are decided by its orientation in the orientation string. The actual sizes of the sub-blocks are finally resolved by the size of the shop floor.

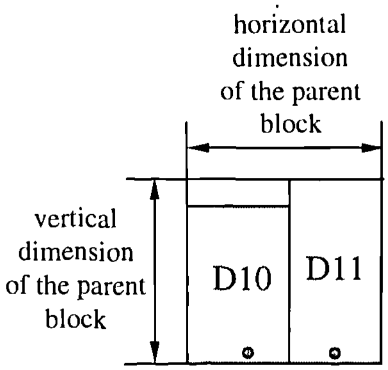

horizontal dimension of the parent block
vertical dimension of the parent block
D10
D11

Figure 5-8 The dimensions of the parent block

The following is the algorithm for realisation of floor layout. In the procedure, the main concerns are the relative positions of the cut-lines and the size of the floor. Let v denote the vertical dimension and h the horizontal dimension of the block, S the cut-line string, P the cut-line position string, d_v the vertical dimensions of departments, d_h the horizontal dimensions of departments.

Procedure: realisation(*S*)

1) Let $v = d_v$, $h = d_h$; $T = S$; $P = \{0\}$.

2) Find a cut-line c_i with the local lowest level in *T*, starting from the tail of *T*.

3) If c_i is even, $v_i = v_i + v_{i+1}$ and $h_i = \max\{h_i, h_{i+1}\}$; $p_i = v_i$.

Otherwise, $v_i = \max\{v_i, v_{i+1}\}$ and $h_i = h_i + h_{i+1}$; $p_i = h_i$.

4) Remove dimensions (v_{i+1}, h_{i+1}); remove cut-line c_i from *T*.

5) If T is empty, return v_1, h_1 and P; otherwise go to 2).

5.4.2 Construction of the Path Graph

In order to obtain the shortest paths, the floorplan needs to be converted to a path graph. Each node in the path graph represents an intersection in the floorplan. Each edge represents a piece of a cut-line between two intersections. The weight of an edge is the distance between the intersections. The construction of a path graph includes creation of the nodes of all intersections of the cut-lines, computation of the distances between the intersections and linking the related nodes to form a path graph. The proposed ideas for path-graph construction are discussed in detail below.

Firstly, all nodes of the intersections need to be defined in the graph. Note the fact that each cut-line has two end points and there is no cross intersection between any pair of the cut-lines. Thus, there are 2×(*m*-1) nodes plus four corner nodes of the border if there are *m* basic blocks in the floorplan. Further observation shows that all the intersections on a cut-line are those that the cut-line intersects with lower level cut-lines at one of the end points. Therefore, all the nodes can be defined by the two end points of the cut-lines without duplication, except the end points of the border cut-lines. The sequence of the construction procedure, like that of the realisation procedure, is executed from the bottom to the top of the block-tree.

Secondly, the intersections on a cut-line need to be found in order to give the relative links of the nodes. Along with the study of the intersection properties, the adjacency of the blocks to the cut-line is also analysed. Obviously, two neighbour blocks that have the same level as the cut-line are adjacent to the cut-line being examined. In other words, it is this cut-line that divides the block into two. For example, supposing that the cut-line to be examined is cut-line ① of the first example in Figure 5-2, the two level-1 blocks are adjacent to the level-1 cut-line. A higher level block is not related to the cut-line, so the rest is focused on the lower level blocks in the block-tree representation, as illustrated in Figure 5-9, and examined level by level. Firstly, all immediate sub-blocks of these two blocks are adjacent to the given cut-line since the division of the normal cut-lines forms them. Consequently, these normal cut-lines intersect with the cut-line being examined. Secondly, at the next level, only the far-left sub-block of the right side sub-blocks to the cut-line is neighbouring the examined cut-line, because the others are on the other side of the block in the layout. Similarly, only the far-right sub-block of the left side sub-blocks to the cut-line is neighbouring the examined cut-line. All the cut-lines of this level are parallel to the given cut-line and thus no cut-line at this level intersects with it. It should be noted that all cut-lines within the non-adjacent blocks do not have an intersection with the cut-line. Similar analysis goes on until no cut-line can be found inside the sub-blocks. Similar analysis can also be performed if the cut-line being examined is of an even level.

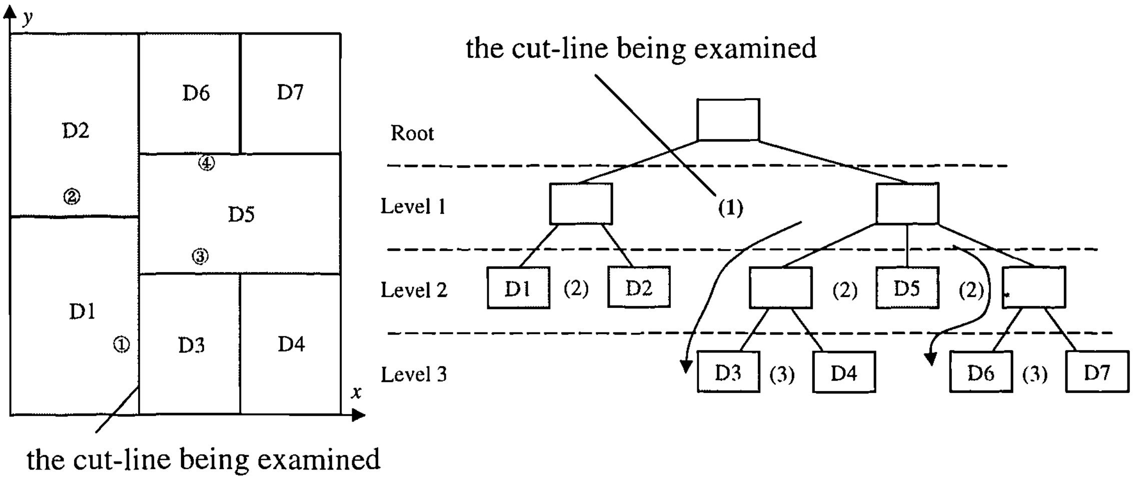


Figure 5-9 The adjacent blocks of a cut-line

The above analysis can be used as the basis for the rules that can be used to find adjacent blocks or the intersections from the block-tree representation or string representation, as follows.

1) In a slicing floorplan, the blocks to which a cut-line is adjacent are those, that,
   i. the two neighbour blocks have the same level as that of the cut-line,

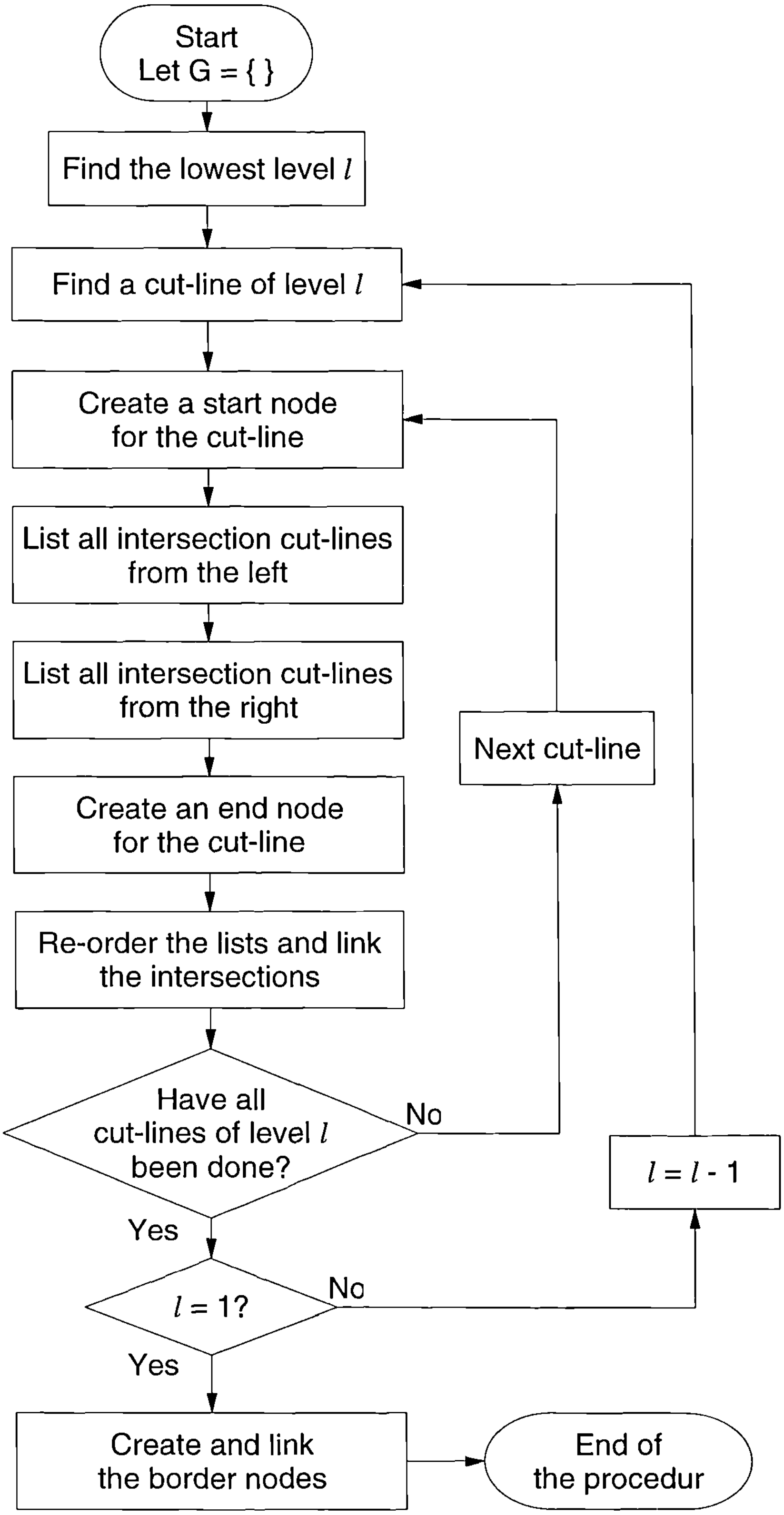

Start
Let G = { }
Find the lowest level *l*
Find a cut-line of level *l*
Create a start node for the cut-line
List all intersection cut-lines from the left
List all intersection cut-lines from the right
Create an end node for the cut-line
Re-order the lists and link the intersections
Have all cut-lines of level *l* been done?
No
Next cut-line
Yes
l = 1?
No
l = *l* - 1
Yes
Create and link the border nodes
End of the procedur

Figure 5-10 The flow chart of the path-graph construction

ii. all the immediate sub-blocks belong to an adjacent block which has the same even/odd level as the cut-line,

iii. a far right/left immediate sub-block belongs to a left/right adjacent block which has the opposite even/odd level as the cut-line.

2) In a slicing floorplan, a cut-line intersects with any cut-line between the two neighbour sub-blocks of the same level that are adjacent to the cut-line.

Next, the relative positions and the distances between the intersections need to be calculated. The relative distances of the cut-lines have already been obtained at the realisation stage, hence the distance between the left-side or right-side intersections are not difficult to compute, but they need to be combined and rearranged. Then, the intersections are linked together to form the graph.

Finally, the borders of the floor, if appropriate, are taken into account by utilising a similar procedure to that used to the cut-lines. The nodes representing the warehouses are linked to the start nodes and the end nodes of any level-1 cut-lines with zero distance. These links are defined as one-way routes to prevent materials flowing back through the warehouses as a short cut. The entire procedure of the path-graph construction is shown in Figure 5-10 in the previous page. An example of the delivery network, which is associated to the layout in Figure 5-1, is shown as in Figure 5-11.

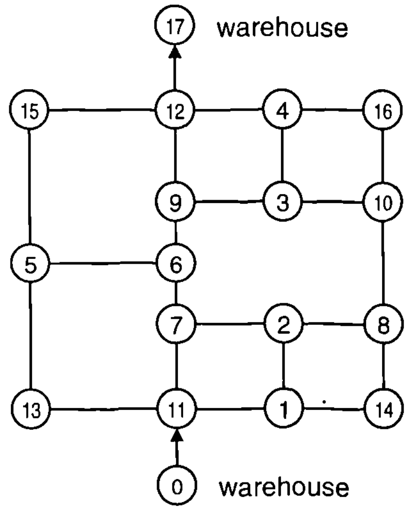


Figure 5-11 A delivery network

Three types of border of the layout are considered in this research: opened, semi-opened and closed. In an opened layout, it is assumed that the layout is surrounded by aisles, as shown in Figure 5-11. In a closed layout, it is assumed that the layout is surrounded by wall and any department cannot be accessed though the border lines, as illustrated in Figure 5-12. It is assumed that in a semi-opened layout, the two side borders are walls, while the borders between the layout and the warehouse are aisles (Figure 5-13).

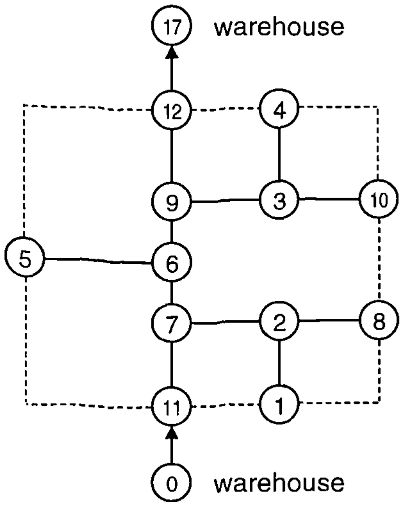


Figure 5-12 A closed delivery network

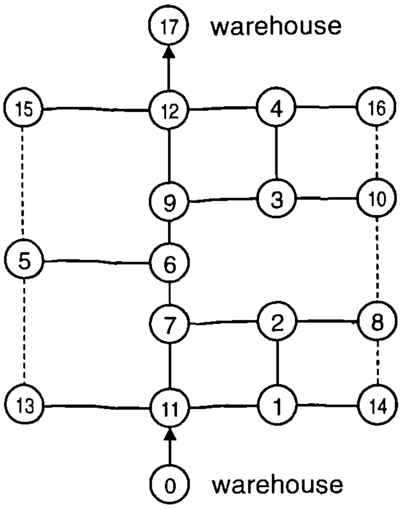


Figure 5-13 A semi-opened delivery network

5.4.3 Shortest Distances in the Path Graph

Finding the shortest paths is a typical problem in graph theory. Many methods can be utilised to calculate the shortest distance. Among them the methods of Dijkstra, Floyd and Dantzig are the most notable. As discussed before, each node in the path graph has up to three edges, except the nodes representing the warehouses. Hence, the path graph is sparse. For a sparse graph, Dijkstra's method for all-pair shortest paths has been proved with a better performance than Floyd's algorithm. The latter scans all the nodes in the graph and, if a shorter path is found, it is used to replace the original path in the path matrix, while the former scans only the linked nodes. The details of the algorithms refer to Miniek (1978), Tarjan (1983) and McHugh (1990). The computation time by Dijkstra's method is $O(mn\log n)$ (Section 3.4.2.1) and both algorithms of Floyd and Dantzig, in contrast, require $O(n^3)$, where n is the number of nodes and m is the total number of edges in G (Tarjan 1983). Therefore, Dijkstra's algorithm is used in this research. The procedure of Dijkstra's algorithm for finding all-pair shortest distances is listed as follows (Tarjan 1983), where E is denoted as the edge matrix of the graph, and D the path matrix.

Procedure: Dijkstra(D, G)

```
1) Initially, let $d[i,i]=0$, $i=1,\cdots,n$ and $d[i,j]=\propto$ for all $i \neq j$.
2) Denote $s$ as a source node and have $s$ labelled.
3) For every labelled node $w$, scan every node $v$ linked to $w$;
   If $d[s,w] > d[s,v]+e[v,w]$, let $d[s,w]=d[s,v]+e[v,w]$ and have $v$ labelled.
   Remove the label of node $w$.
4) If all the nodes in graph $G$ have been used as the source node then stop.
   Otherwise go to step 2.
```

5.4.4 Delivery Network and Material Handling Costs

Further details of the delivery network and transportation cost are taken into account in this subsection. In a transport path, any change of the cut-line routes indicates a turning point, which may require more complicated control and/or the reduction of speed. Therefore, there is a need to put some penalties on intersections. Moreover, Any P/D station needs to link to the related node of the path graph, because any delivery is taken between P/D stations rather than intersections (nodes). Thus, the station-to-station shortest paths are derived, which based upon the node-to-node shortest paths of the path-graph, plus the P/D station to node components.

5.4.4.1 Penalties on Intersections

There are two kinds of penalties on intersections considered in this research: intersection penalty and turning penalty. Intersection penalty p_1 puts a penalty on each edge in the path. Turning penalty p_2 puts a penalty on each edge from an intersection to an intersection of a lower level aisle, i.e. on the two end segments of a cut-line (as illustrated in Figure 5-14). The penalties are calculated in distance and are added to the real distance at the path-graph construction stage in Section 4.5.2.

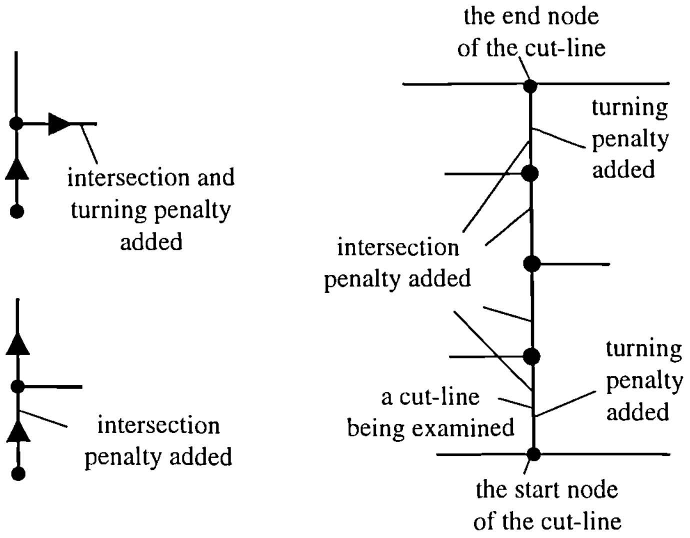


Figure 5-14 Penalties on intersections

5.4.4.2 Station-to-station Shortest Path

Generally a station-to-station path can be divided into three parts. The first part is from a pick-up station to a near intersection node. The next is from that node to another node near to the drop-off station and finally, the third part is from the second node to the destination. In this case, a P/D station links to up to two neighbour nodes. Consequently, there are only four nodes involved in the computation of the station-to-station shortest path. For any pair of pick-up or drop-off stations, the shortest distance is computed by selecting the shortest among the enumerated routes through two of the four related intersection nodes, between which the shortest distances are obtained by the algorithm described in Section 5.4.3.

5.4.4.3 Total Cost for Material Handling

The floor layout at the floorplan stage is evaluated by the total cost of the material handling between departments. The total cost of the material handling is calculated by the following equation.

$$c = E \cdot \sum_{i=0}^{m} \sum_{j=1}^{m+1} f_{ij} \times d_{ij} \tag{5-5}$$

where c is the total cost of the material handling.

E is the inter-cell material handling cost for a unit of volume distance.

i, j are indices of the department; $i = 0$ and $j = m + 1$ represent the raw material warehouse and the product warehouse respectively.

m is the number of the departments.

f_{ij} is the material flow density from department i to department j.

d_{ij} is the shortest aisle distance from the pick-up station of department i to the drop-off station of the department j.

The issue of arranging the layout plan to minimise the total inter-cell material handling cost will be discussed in the next section.

5.5 A Genetic Algorithm for the Floorplan

The genetic algorithm for the floorplan problem is based on a composite genome that consists of three chromosomes. The first chromosome is a list genome for the department string, which represents departments and their sequence. The second one is a special genome composed of the slicing cut-line string which represents aisles and layout and the last chromosome is a one-dimensional array genome representing the orientation of each department. Both list genome and one-dimensional array genome can use the standard GA operators provided by GAlib (Wall 1996), but there is no standard operator for the slicing cut-line string genome. The methods of initialisation, crossover and mutation for the cut-line string chromosome are discussed as follows.

5.5.1 Creation of a Random Initial Cut-Line String

Any cut-line string created by the GA must represent a slicing floorplan containing all departments to be located; otherwise, the cut-line string is illegal. For instance, within the left level-1 block of example 2 in Figure 5-6 (page 102) it would be illegal if the far-left level-2 cut-line changed to be that of level 3.

Assuming there are two level-0 cut-lines at the two ends of the string, a rule of a slicing cut-line string exits, which can be stated as follows. A cut-line string can represent a slicing floorplan if and only if there is at least one immediate lower cut-line, if there is any lower cut-line, between any two cut-lines. The following procedure to create a legal random cut-line string is based upon the above rule. An example is provided in Figure 5-15 to illustrate the procedure.

Procedure: initialise_cut_line

1) Suppose that the cutting string is enclosed by two 0s.
2) Define a cut-line of level 1 at a random position in the cutting string.
3) Define a cut-line of level *i* or *i*+1 randomly at a random position between two adjacent defined cut-lines if the lower level of the two defined cut-lines is *i*.
4) Repeat 3) until the string is fully defined.

The string to be determined:

0 X X X X X X X 0

Define a 1 in a random position:

0 X X 1 X X X X 0

Define two following numbers:

0 X 2 1 X X 1 X 0

Define the remaining numbers:

0 2 2 1 2 3 1 2 0

The final string:

2 2 1 2 3 1 2

Figure 5-15 An example of the creation of a random string

5.5.2 One-point Crossover for Cut-Line Strings

A heterosexual one-point crossover is used to create the offspring of the next generation for evolution. Parents are selected by the use of the Steady-State GA (Mitchell 1996, Michalewicz 1996). After swapping the genes of parent strings by use of the crossover operator, illegal offspring strings, which cannot represent.a slicing floorplan, may be produced. Therefore, the strings need to be constrained to represent a slicing floorplan. A repair procedure is used to deal with this constraint. The procedure repairs the illegal digits near the crossover in order to minimise the loss of the information from the parents. The following procedure is used to repair the illegal strings created by the crossover operator.

Procedure: cut-line_repair

1) Add two virtual cut-lines of level 0 as the initial embracing cut-lines at the both ends of the cut-line string, i.e. $l = 0$.
2) Find a lower level (l-1) cut-line between the two embracing cut-lines. Define it as the new embracing cut-line in place of the same side on the crossover point. If no cut-line of level (l-1) can be found, all numbers between the embracing cut-lines should be increased by 1. Repeat the step until a lower level cut-line is found.

3) Find a same level cut-line between the two embracing cut-lines and define it as the new embracing cut-line in the place of same side on the crossover point. Repeat 3) until no same level cut-line can be found.
4) Repeat 2) and 3) until there is no number left between the embracing cut-lines.

For example, two parents and a crossover point are as shown in Figure 5-16. Before repair, both cut-line strings created by the parents are illegal, because the sub-string between the genes of level 1 lack a level-2 gene in the string of child 1 and there is no level-1 gene in child 2. The repair procedure increases the level of the illegal genes by unity (decreases in number).

one-point crossover							
parent 1	2	1	3	| 2	2	3	
parent 2	2	3	3	| 4	1	2	
Before reparation				⇓			
child 1	2	1	3	| 4	1	2	
child 2	2	3	3	| 2	2	3	
After reparation				⇓			
child 1	2	1	2	| 3	1	2	
child 2	1	2	2	| 1	1	2	

Figure 5-16 Crossover and reparation

5.5.3 Mutation of the Genes

A mutation operator is a mechanism that prevents the loss of diversity of chromosomes at a given bit position. By utilising the tree representation form, the mutation of the cut-line chromosome moves the sub-tree up or down on a random basis, as illustrated in Figure 5-17, provided that the mutated chromosome is a legal cut-line string. The mutation procedure is listed as follows.

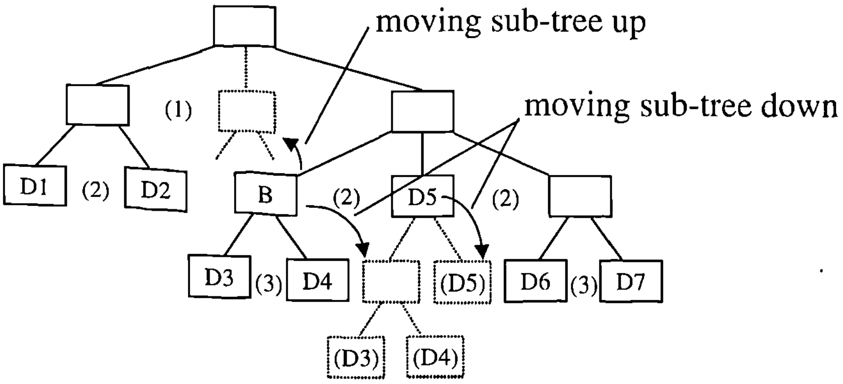


Figure 5-17 Mutation of a block tree

Procedure: slicing_mutation

1) Select a cut-line to be mutated randomly.
2) Find the block defined by the cut-line.
3) If there are at least two neighbour blocks of the same level, move the block to be mutated downward randomly, by adding 1 to the related genes of the cut-line string.
4) Otherwise, move the block upward, by subtracting 1 from the related genes of the cut-line string.
5) If the cut-line to be mutated is the only cut-line of level 1, go to 1) re-selecting a cut-line.

5.6 Slicing Aisle Structure

In the previous sections the whole slicing structure is regarded as a virtual aisle structure. In a plant, aisles not only occupy areas, but also indicate the complexity of the layout. Therefore, not all cut-lines will become aisles if they are unnecessary. This section presents a method of optimising aisle structure by removing redundant aisles.

5.6.1 Slicing Aisles

A slicing aisle is a roadway for material handling placed in the same location as a slicing cut-line. The level of the aisle is denoted as the level of the slicing cut-line. If a cut-line of level 1 is selected as an aisle, it is a main aisle of the shop floor.

A slicing-aisle structure of a layout consists of one or more slicing aisles that make all departments accessible. The lowest level of the aisles indicates the level of the slicing-aisle structure. The following studies focus only on slicing aisles and the slicing-aisle structure, or aisles and aisle structure for short. For instance, the aisle structure of the layout in Figure 5-7 (page 104), represented in string form, could be:

2 1 * 2 2 *

The two aisles of level 3 could be redundant if they are unnecessary for providing the shortest paths for material flow.

From the above example it can be seen that the aisle-structure design is just a selection procedure, deciding which cut-line given by the floorplan stage should be selected as an aisle. Therefore, a binary decision string of aisle structure, or aisle-structure string, can be used for this purpose. The decision string has the same length as the cut-line string. Each binary digit corresponds to a cut-line. 1 stands for selecting the cut-line as an aisle and 0 for not. The decision string of the above aisle structure is:

1 1 0 1 1 0

The minimum requirement for an aisle structure is that it should provide the paths for the material handling. In this research, the requirement is narrowed to the shortest paths, i.e. the aisle structure provides the shortest paths for all material flows. The problem of finding all possible shortest paths in a path graph defined by all cut-lines will be discussed in the following subsection.

5.6.2 Shortest Paths

The following algorithm is used for finding the shortest path from node p to node q (Carre 1979). The procedure returns the shortest path in a node sequence array S. Assuming that r_p is an array of single-source shortest distances and d is a matrix of all-pair shortest distances.

1) Let $s_0 = p$ and let $k = 1$.
2) Let s_k be any index such that $d[s_{k-1}, s_k] + r_p[s_k] = r_p[s_{k-1}]$.
3) If $s_k = q$ then halt; otherwise record s_k, increase k by 1, and return to 2).

The above procedure can be extended to find all possible shortest paths from a source node n_s to a destination node n_d. The following procedure uses the recursive technique to record the shortest paths by putting them into a list, where *path* is denoted as a list object containing the nodes of a path and *list* is another list object containing the paths.

```
Procedure: shortest_paths($n_s$, $n_d$, $path$)
1) for any directly connected node $n_c$, if $e_{sc} + e_{cd} = e_{sd}$ then,
    i. add the cut-line defined by $n_s$ and $n_c$ to $path$ and,
    ii. if $n_c = n_d$, insert $path$ into $list$,
    iii. otherwise, call shortest-paths($n_c$, $n_d$, $path$).
2) return.
```

5.6.3 Optimisation of Aisle Structure

The optimisation model of aisle structure in Section 5.3 works on the condition that any shortest path p_{ij} in set $\boldsymbol{P}_{ij}$ provides a path between department *i* and *j*. However, an aisle-structure string may not supply a full path for a certain flow. Thus, it is necessary to add a constraint, which guarantees the full paths for all flows, to the model. In this research, a penalty technique is utilised to deal with this constraint.

$$\text{Minimise } \sum_{k=1}^{m-1}(w_k \times a_k^s) + p^e \times \sum_{i=1}^{m}\sum_{j=1}^{m}\left\{e_{ij}^{\min} : f_{ij} \neq 0\right\} \qquad (5\text{-}6)$$

where w_k is the weight on the cost of the cut-line *k* if it becomes an aisle,

a_k^s is the *k*th element in the aisle-structure string, *k* = 1, 2, ... , *m*-1.

p^e is the penalty on the excluded aisles that is essential for a necessary path ,

$e_{ij}^{\min} = \min_{p_{ij} \in \boldsymbol{P}_{ij}} e_{ij}$

e_{ij} is the number of the cut-lines, that are not included in the aisle-structure string $\{a_k^s\}$, for path p_{ij},

p_{ij} is the set of cut-lines for a path of $\boldsymbol{P}_{ij}$,

$\boldsymbol{P}_{ij}$ is a set of the shortest paths from department *i* to department *j*,

f_{ij} is the flow density form department *i* to department *j*.

The first item in the above model represents the total cost for the aisle structure, and the second is the penalty function, which levies the aisles that are not included in the necessary path. The implication of the above model is listed as follows.

Procedure: objective_of_aisles

1) An aisle-structure string is given by an optimisation algorithm, which will be discussed in the next section.
2) The number of aisles is calculated by summation of the decision string.
3) For a non-zero flow all possible shortest paths between the pair of P/D stations are found.
4) Each shortest path is compared with the decision string. If some cut-lines for the path are not selected as aisles in the decision string, the number of such cut-lines is recorded.
5) The minimum number of the excluded cut-lines among the shortest paths is added to the objective function by multiplying by a weight factor.
6) Repeat 3) to 5) until all flows are applied.

A genetic algorithm with a binary string chromosome and a one-point crossover as described in Section 2.5.6 is used in this research to perform the optimisation procedure.

5.7 Computational Results

The algorithm for the plant layout optimisation was tested by the following problems. Problem 1 and problem 2 were specially designed so that the optimum solutions were known. The data in problem 3 is from results of the previous chapters. The efficiency of the algorithm will be discussed in Section 5.7.3 by comparison with a pure random search.

5.7.1 Problem 1

This problem contains six departments. The flows between departments and the dimensions of the departments are given as shown in Table 5-1 and Table 5-2. The department 0 and 7 are denoted as the raw-material warehouse and the product warehouse respectively.

	Departments							
	0	1	2	3	4	5	6	7
0	0	0	0	465	0	0	518	0
1	0	0	0	0	0	0	0	483
2	0	303	0	0	85	65	0	0
3	0	0	315	0	0	150	0	0
4	0	0	0	0	0	0	0	500
5	0	180	0	0	415	0	0	0
6	0	0	138	0	0	380	0	0
7	0	0	0	0	0	0	0	0

Table 5-1 The flow chart of problem 1

The algorithm successfully found the optimum solution of the layout, including the orientations of the departments and the aisle structure which contains a main aisle for material handling. The data obtained from the algorithm is listed in Table 5-3 and the layout is illustrated in Figure 5-18.

Department:	1	2	3	4	5	6
Width :	180	200	140	150	150	200
Length :	180	180	180	200	200	200
P/D type :	1	1	1	1	1	1

Table 5-2 The dimensions of the departments

Department :	3	2	1	6	5	4
Orientation :	3	3	3	1	1	1
Cut-line :	2	2	1	2	2	
Aisle Structure :	0	0	1	0	0	
Cut-line Position :	140	340	195	200	350	
Floor Dimension :	410	520				
MH Cost :	0.51116×10^6					

Table 5-3 The optimal solution of problem 1

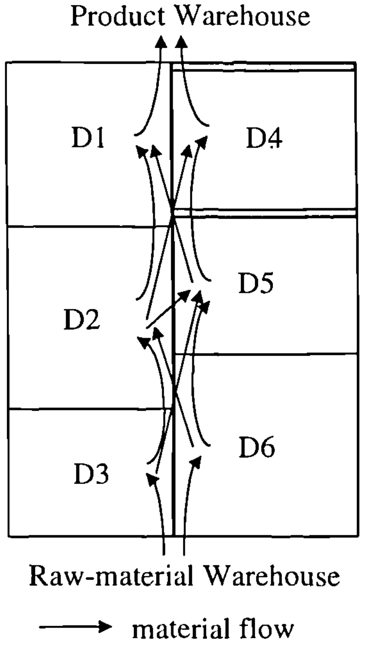


Figure 5-18 The layout of problem 1

5.7.2 Problem 2

This problem contains seven departments that have different types of P/D' stations. The flows between departments and the dimensions of the departments are given in Table 5-4 and Table 5-5.

	Departments								
	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	300	200	100	0
1	0	0	0	0	0	0	0	0	150
2	0	0	0	0	0	0	0	0	200
3	0	0	200	0	0	0	0	0	250
4	0	0	0	150	0	0	0	0	0
5	0	150	0	0	150	0	0	0	0
6	0	0	0	0	0	0	0	200	0
7	0	0	0	300	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0

Table 5-4 The flow chart of problem 2

The optimum solution of the problem was also obtained by the algorithm. It gives relative positions and orientations of the departments correctly, and suggests the aisle structure by removing the redundant aisles. Table 5-6 provides the results obtained from the algorithm and Figure 5-19 illustrates the layout reconstructed according to the data.

Department :	1	2	3	4	5	6	7
Width :	200	150	200	100	200	150	200
Length :	150	150	150	150	150	150	150
P/D type :	1	-1	1	-1	1	-1	1

Table 5-5 The dimensions of the departments

Department : 7 3 6 4 2 5 1
Orientation : 2 1 2 2 1 2
Cut-line : 3 3 3 3 1 1 1
Aisle Structure : 0 1 0 0 1 0
Cut-line Position : 200 165 150 250 330 200
Floor Dimension : 510 400
MH Cost : 0.24x 10^6

Table 5-6 The optimal solution of problem 2

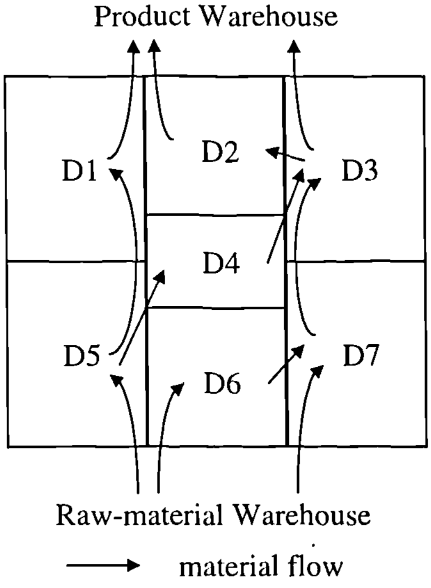


Figure 5-19 The layout of problem 2

5.7.3 Problem 3

The cell-formation problem from Seifoddini and Djassemi (1995) is again used here. The cell configuration and the layouts within the cells were presented in Chapter 3 and Chapter 4 respectively. Figure 5-20 represents an optimised layout using the genetic algorithm. The layout uses the 4-cell configuration with single-row ·cells. In comparison, the optimised layout of the 4-cell configuration with U-shape single-row cells is given in Figure 5-21. In the case of these 4-cell configurations, the latter provides the better results with respect to both intercell material handling cost and area utilisation. Although the inter-station distances are much reduced in the U-shape-cell solution, the material handling cost is not reduced significantly due to the increase of the distances between the stations of department 1 and 2 and the warehouses.

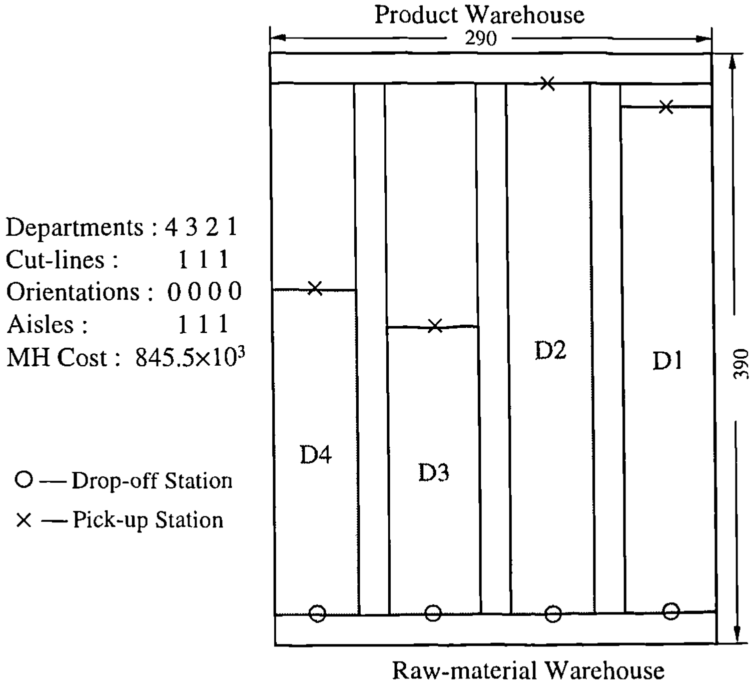


Figure 5-20 The layout of the 4-cell configuration with the single-row cells (L1)

Departments : 4 3 1 2
Cut-lines : 2 1 2
Orientations : 3 3 1 1
Aisles : 0 1 0
MH Cost : 815.5x10^3

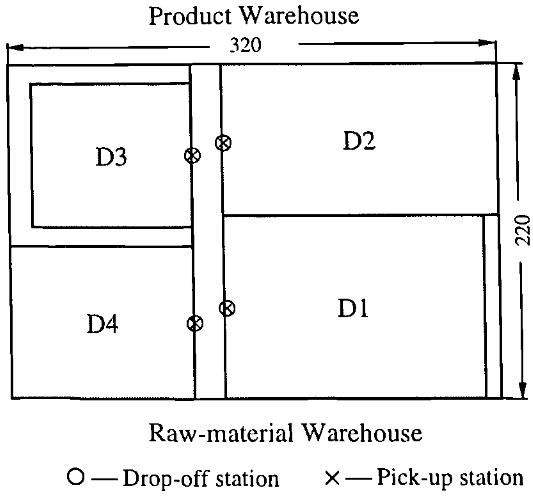


Figure 5-21 The layout of the 4-cell configuration with the U-shape cells (L2)

Departments: 2 1 3 4 6 5
Cut-lines : 1 2 3 1 1
Orientations: 0 3 0 3 0 0
Aisles : 1 1 1 1 1
MH Cost : 798.1x10^3

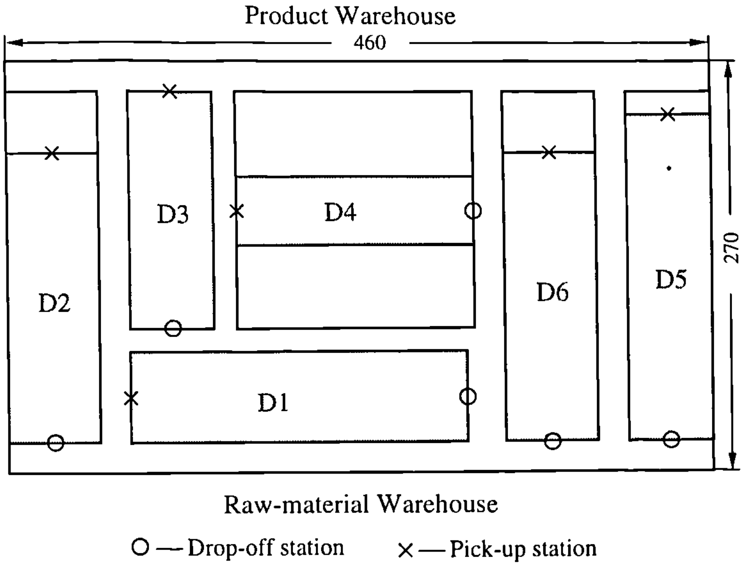


Figure 5-22 The layout of the 6-cell configuration (L3)

Departments: 1 7 6 2 8 3 4 5
Cut-lines : 1 3 2 1 1 2 1
Orientations: 0 0 0 1 1 0 3 0
Aisles : 1 1 1 1 1 1 1
MH Cost : 874.0×10^3

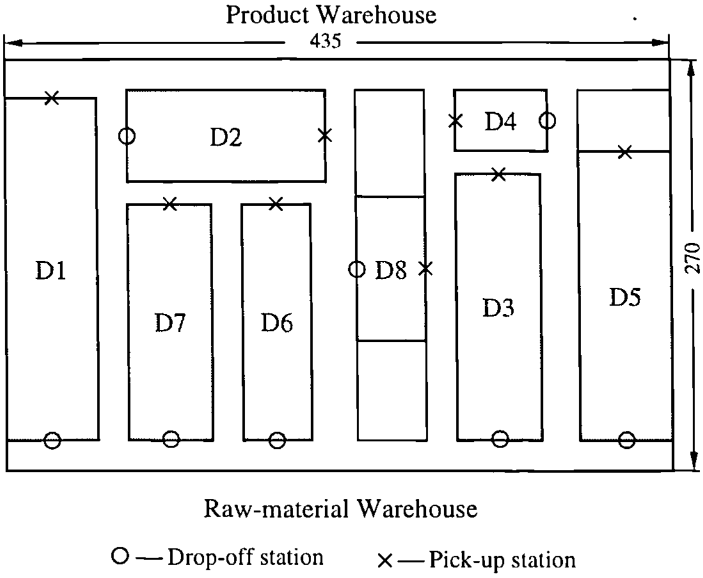
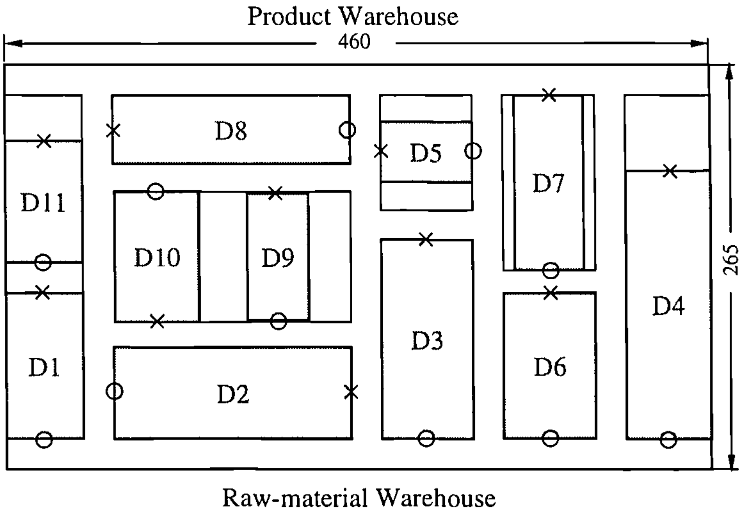


Figure 5-23 The layout of the 8-cell configuration (L4)



Departments: 1 11 2 10 9 8 3 5 6 7 4
Cut-lines : 2 1 2 3 2 1 2 1 2 1
Orientations: 0 0 1 2 0 3 0 3 0 0 0
Aisles : 1 1 1 0 1 1 1 1 1 1
MH Cost : 1060.5×10^3

× — Pick-up station
O — Drop-off station

Figure 5-24 The layout of the 11-cell configuration (L5)

Solutions	Number of Cells	Intracell MH Cost ($\times10^3$)	Intercell MH Cost ($\times10^3$)	Total Cost ($\gamma = 1$) ($\times10^3$)
L1	4	1038.9	845.5	1884.4
L2	4	1038.9	815.5	1854.4
L3	6	755.5	798.1	1553.6
L4	8	638.6	874.0	1512.6
L5	11	542.6	1060.5	1603.1

Table 5-7 The material handling costs of the solutions

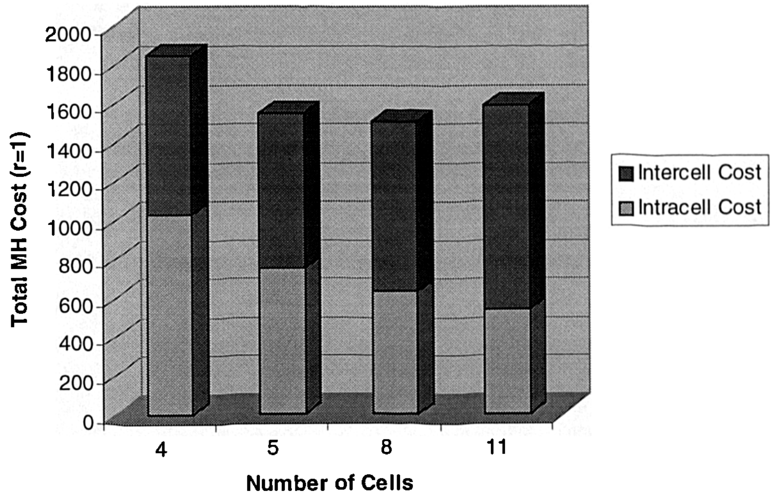


Figure 5-25 The comparison of the cell configurations

The optimised results of 6-cell, 8-cell and 11-cell layouts are illustrated in Figure 5-22, Figure 5-23 and Figure 5-24 respectively. In order to evaluate the cell configurations and their layouts, the total cost for both the intracell and intercell material handling should be taken into account. The total cost in Table 5-7 gives the sum of intercell and intracell volume distance, as illustrated in Figure 5-25, which presents the 8-cell layout as the best solution. However, in most cases, the total cost is not the simple sum of the volume distance. In general, the total cost for the material handling can be represented by a weighted sum of intercell and intracell volume distance. That is,

$$T = \alpha \cdot (V_a + \gamma \cdot V_e) \qquad (5\text{-}7)$$

where T is the total cost for material handling.

α is the cost per unit of volume distance.

V_a is the total volume distance of the intracell material handling.

V_e is the total volume distance of the intercell material handling.

γ is the ratio of the cost per unit of volume distance for intercell material handling over that for intracell material handling.

The cost rates for intercell material handling and for intracell material handling also influence the final decision of cell formation and layout. The cost rates are determined by the types of material handling systems selected within cells and at the plant level. According to the above equation, the total material handling cost associated with the inter/intra cell cost ratio of the solutions can be drawn as in Figure 5-26. Consequently, from the figure it can be seen that the 11-cell solution could be taken into consideration only when$\gamma \le 0.515$. When $0.515 \le \gamma \le 1.54$, the 8-cell layout provides the best solution. Otherwise, the 6-cell layout gives the best candidate. For any value of γ, the 6-cell layout is better than the 4-cell solutions. The layout of the 4-cell configuration with the U-shape cells has value if the area utilisation is important.

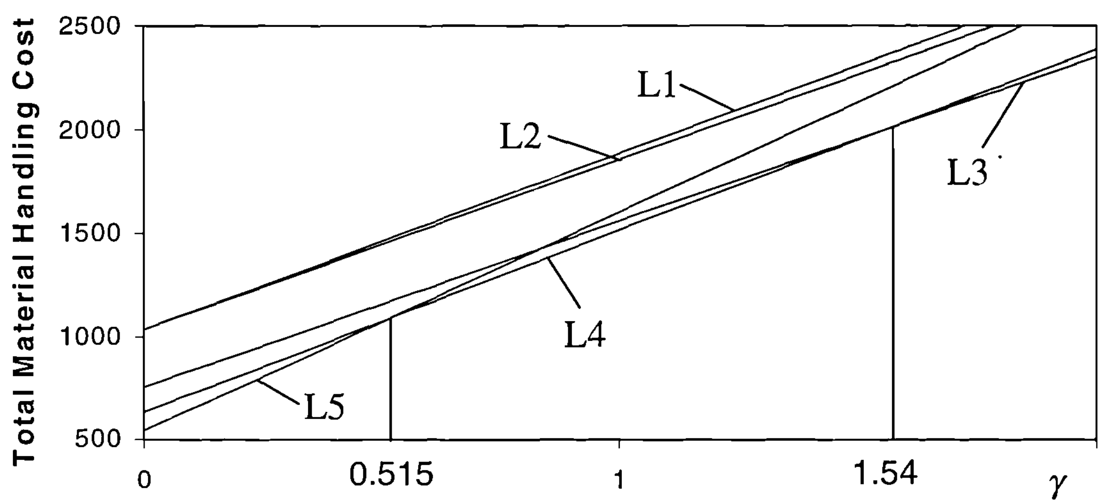


Figure 5-26 The impact of the intercell and intracell cost ratio

5.7.4 The Efficiency of the Genetic Algorithm versus a Random Search

Comparisons of the GA with a random search (RS) was performed in order to assess the efficiency of the GA for the layout problem. The comparison of the results obtained by a single run is unhelpful because both algorithms are random-based methods. The statistical data illustrated in Figure 5-27 is based upon 1000 runs for problem 1. In each run, the number of times the objective function was called by each algorithm is given as the horizontal axis in Figure 5-27. The values in brackets are the numbers of times the algorithm achieved the global minimum. The statistical data for problem 2 is represented in Figure 5-28.

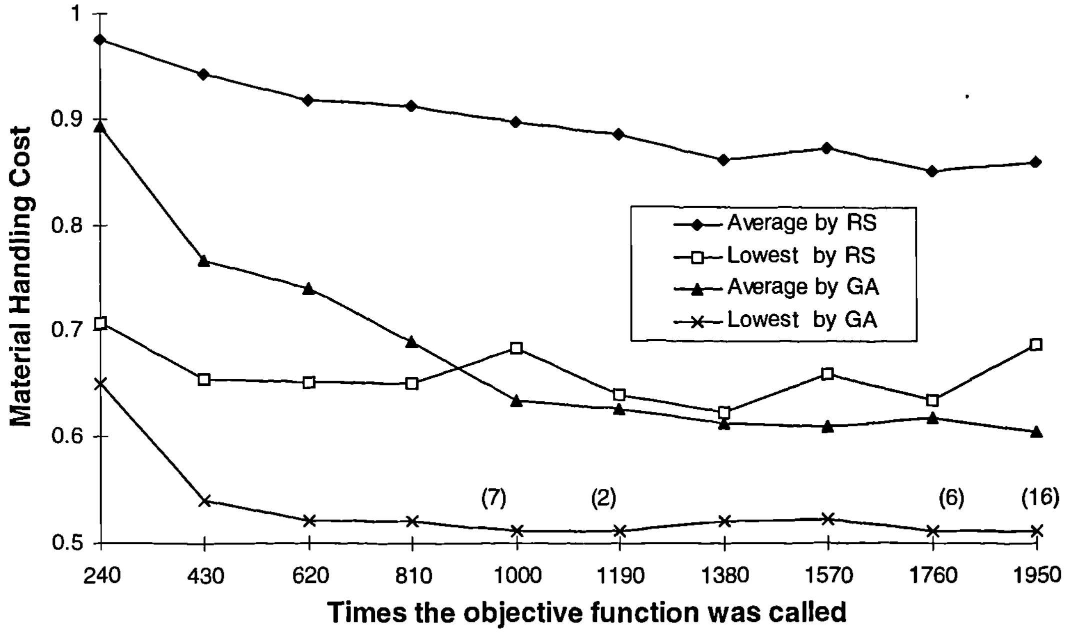


Figure 5-27 GA versus RS for problem 1

It can be observed in both figures that the results obtained by the genetic algorithm are much better than that by the random search. The average values, which tell approximately how good a result might be in a single run, acquired by the RS (the higher curves in Figure 5-27 and Figure 5-28) are much worse than that by the use of the GA. The lowest values achieved by the RS, which can only compete with the average values obtained by the GA, never reached the optima of the two problems in the experiment. The figures also reveal that even the GA has difficulties in achieving the global optimum for such simple problems.

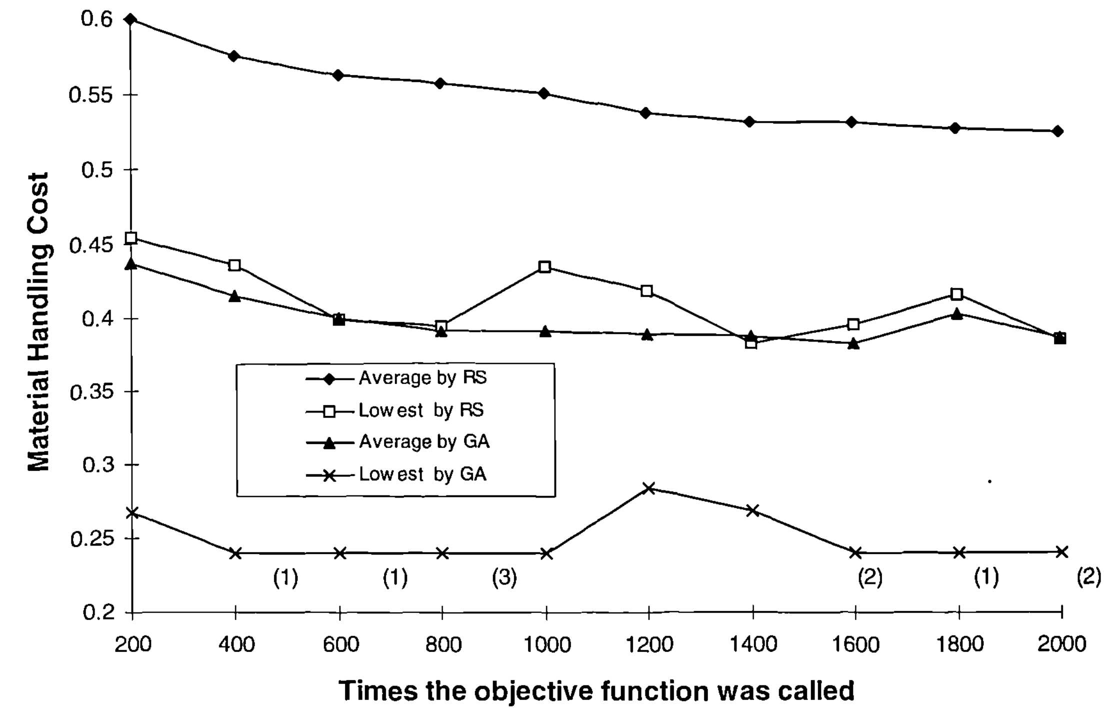


Figure 5-28 GA versus RS for problem 2

5.8 Discussion and Conclusions

Conclusions drawn from the work in this chapter will be presented as follows. Some details involved and suggestions for further study are also discussed in this section.

5.8.1 Conclusions

A method has been developed in this chapter to solve layout problems with consideration of aisles and conventional aisle structure by a representation of a slicing floorplan. The method decomposes the problem into two stages. The first stage minimises the material handling cost with aisle distance and the second stage optimises the aisle structure.

The slicing model of layout corresponds to aisle structure in a natural way. When the layout contains one aisle, it is a single-roadway layout; when it contains one main aisle and some lower level aisles, it becomes a spine layout and when it contains more than one main and lower level aisles in a certain manner, it represents a matrix layout. Unlike the method in the previous chapter, the algorithm will find the most appropriate structure for the layout, instead of a specified *priori.*

A comparison study of the genetic algorithm and the random search for the problem has been carried out. It shows that the GA has a much higher efficiency in searching for the optimum of the problem than the RS, although further study is needed to improve the efficiency of the GA.

The layout design needs pre-specified or interactive parameters such as penalties on intersections and weights on aisles of different levels.

5.8.2 Discussion and Recommendations for Further Studies

If the extension of the methodology and more details are taken into consideration, many other problems may be involved. The following sub-sections provide some suggestions for further studies.

5.8.2.1 Accessibility and Flexibility

The method in this chapter is based upon the assumption that all departments are accessible by aisles leading from or to the warehouses, giving the highest routing flexibility. Routing flexibility represents the ability of the system to manufacture products via a variety of different operation routes (Browne *et al.* 1984). In some situations, not all departments need necessarily be accessible from or to the warehouses and hence the aisles may be further reduced. Material-handling cost could accordingly be further reduced. Therefore, the aisle structure corresponds to the flexibility required and this consequently increases the complexity of the problem.

5.8.2.2 The One-way Aisles Problem and Tandem Systems

Single-direction traffic on an aisle may significantly reduce the complexity of the traffic control and make the material handling system more efficient. Sometimes it is preferred to use two separate one-way aisles instead of a two-direction aisle, even if the travel distance is a litter longer. Tandem systems, which contain several non-intersecting one-way loop aisles, are desirable for automatic guided vehicles (AGVs).

If the tandem systems are independent, the methods developed in this chapter and in the previous chapters can be used for tandem systems, by regarding each tandem system as a machine cell. The method in this chapter cannot solve the problem when the tandem systems are not independent.

A more complicated situation occurs if the aisle structure contains both two-direction aisles and one-way aisles. For example, some main aisles may be unavoidably bi-directional and the aisles linking the main aisles may be unidirectional. Also minor aisles, such as the sub-aisles in a spine layout, can be bi-directional if there is not much traffic on the road.

5.8.2.3 Non-slicing Representation

The method in this chapter is limited to a slicing model, which may not represent the department shape accurately nor fully utilise the floor area. Non-slicing floorplans represented by polar graphs (Otten 1988 and Lengauer 1990) are potential to be explored for layout problems but complex non-slicing layout is not expedient because it may involve many zigzag aisles, which increase the material handling cost and slow down the speed of the vehicles. A wheel structure in Figure 2-12 (page 34), which may contain a loop, is quite a simple and common non-slicing structure. Hopefully, a representation could include wheel structures and ease of optimisation.

5.8.2.4 Location of Warehouses

It was assumed that the raw material warehouse and the product warehouse are located at the both ends of the floor respectively and not included within the shop-floor area. However, the warehouses could be regarded as hard or soft departments so as to be included inside the floor area. To achieve this, the methodology described in this chapter needs to be extended to accommodate both hard and soft departments. Moreover, the departments with multi-stations also need to be handled in the method and this is another direction for further study.

Chapter Six

Conclusions

The conclusions drawn from this research are summarised in this chapter. The direction of further research is also suggested.

6.1 Conclusions

Manufacturing layout is one of the important features of a manufacturing system, and has a strong impact on the organisation and management of the plant. It has an effect upon adaptability and efficiency of the system, and more directly on the cost for internal material handling. Cellular manufacturing is a strategy which groups the machines required to manufacture similar components into machine cells, in order to reduce the material handling cost and setup time. The design of cellular manufacturing layout in this research was divided into three stages: cell formation stage, cell layout stage and plant layout stage.

A heuristic for manufacturing cell formation has been developed in the first stage. The cells clustered by the method are represented in a more natural way than by previously published methods, in which the specified number of cells need to be given or the number of machines in a cell is constrained by a pre-specified upper limit. The cells grouped by the proposed method show the natural basic property of the relationship between machines and parts. The computational trials showed that the method is effective in minimising intercell material movement.

In the second stage, the layout is based upon a previously specified material handling system, which contains the main handling equipment with a single-roadway and supplementary handling devices for the in-sequence material flow. A genetic method is presented for performing a design for a single-roadway machine layout. An evolutionary model for the above problem has been developed. The model includes the simplest and the most common layout patterns: single-row layout, double-row layout and U-shape layout. Four types of material flow, produced by unidirectional and bidirectional material handling equipment (MHE), and by the main MHE and the supplementary MHE respectively, were also taken into account within the model. The efficiency of two types of string representations for the genetic search was investigated. The representation that involved less interruption of neighbour genes was found to have a higher search efficiency. This could be a guideline for the selection of the representation in genetic algorithms in this application.

In the third stage, a sliced floorplan is used to represent the layout. One of the merits of this representation is that strong constraints, which prevent the departments from overlapping, can be avoided. Another merit is that the slicing cut-lines, which are simultaneously obtained from the representation, can be utilised as the aisles for the material handling system. Therefore, the aisle-structure corresponding to the layout is designed concurrently. The aisle structure created in this stage could be much more complex than that in the previous stage and this type of aisle structure, such as a spine structure or a matrix structure, is generally used for industrial trucks or AGVs.

A genetic algorithm (GA) is used for solving the problems of the layout design and aisle-structure optimisation. A representation suitable for the genetic algorithm was developed. The representation separates the different types of data into four strings representing the design variables, which are a department string, an orientation string, a cut-line string and an aisle string. A slicing layout has exactly one variable set with corresponding values and *vice versa*. A comparison study of the genetic algorithm against a random search for the problem shows that the GA has a much higher efficiency in searching for the optimum of the problem than the random search.

A comparison has been made between different solutions produced using the proposed approach based upon a problem described in previous literature. The results show that the best solution is a trade-off between the number of cells and cell size. The quality of a solution also depends on the spatial arrangement of the cells. Larger cell size may cause an increase in skipping distance, while the use of more cells needs more aisles, which occupy more area and make traffic more complex. Therefore, it is inadequate that the cell configuration is evaluated only by the intercell flow. Intercell flow, intracell flow and the spatial arrangement of the machines and cells should be considered.

In this research the selection of the process sequences, the number of cells, the type of material handling equipment and the type of cell was not automatic but was decided by the designer. The effect of layout on the WIP and setup time, which are more affected by planning and scheduling, was also not included in the current research. These could be directions for further research, of which more details are suggested in the following section.

6.2 Recommendations for Future Work

The integration of manufacturing system design could be achieved based upon this research by assuming that the optimal solution of the system is near to a combination of the optimal solutions of the sub-systems. This is still explosive for computation, due to the intractability of the optimisation of the sub-systems. Some search techniques are needed for obtaining a good solution in a reasonable time. The recommendations for further research of the sub-systems were provided in the last sections of the previous chapters. The following is the suggestion for the optimisation of the whole system.

The suggested procedure for the design of flexible manufacturing systems is illustrated in Figure 6-1. An overall evaluation could be extended on the basis of the work of Bramham and Appleton (Bramham and Appleton 1995). Simulation may be needed to obtain some time-related production targets, such as lead time, throughput and WIP, because these data are difficult to obtain by arithmetic computation. Simulation experiments, which are not included in this thesis, showed the behaviour of a system is also highly impacted by the planning and scheduling of the system, especially when numerous products or components are manufactured in the system simultaneously. Therefore, the optimisation of the planning and scheduling is also necessary before the simulation runs. However, simulation needs intensive computation.

In order to reduce the number of system evaluations by simulation, a Taguchi method aggregating with artificial intelligence (AI) could be used. A Taguchi method can provide robust design and does not evaluate the objective intensively, therefore, the method is suggested for analysis of the result. Research using Taguchi methods has been performed in the past few years (Wild and Pignatiello 1991, Mayer and Benjamin 1992, Chen and Chen 1995, Benjamin *et al* 1995, Shang and Tadikamalla 1998). The analysed results are stored in a database for the use of an AI system, or expert system, to provide the best direction for the next search, in which the overall cost could be reduced. The hybrid approach could reduce the computational effort to a minimum.

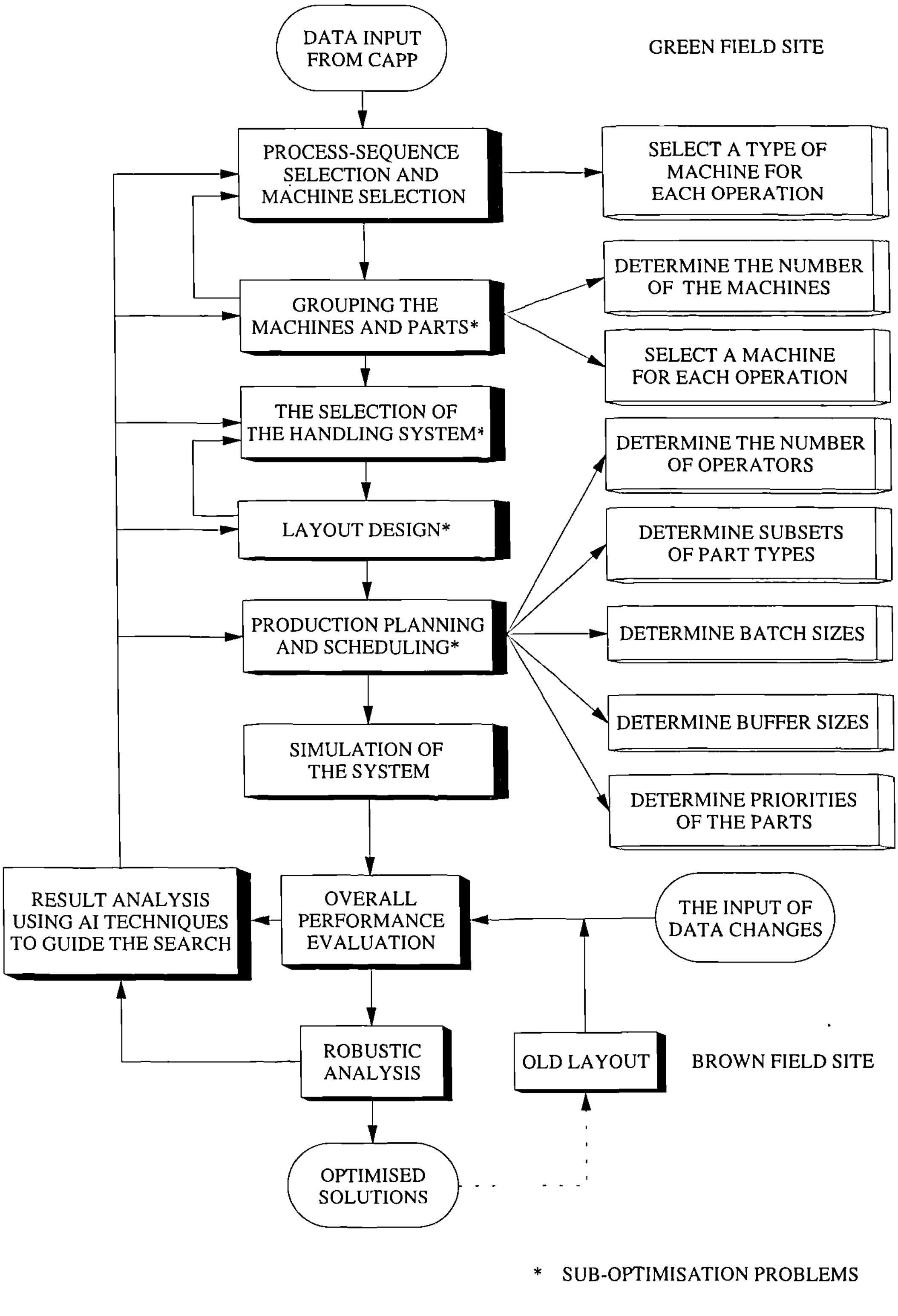

DATA INPUT FROM CAPP
GREEN FIELD SITE
PROCESS-SEQUENCE SELECTION AND MACHINE SELECTION
SELECT A TYPE OF MACHINE FOR EACH OPERATION
DETERMINE THE NUMBER OF THE MACHINES
GROUPING THE MACHINES AND PARTS*
SELECT A MACHINE FOR EACH OPERATION
THE SELECTION OF THE HANDLING SYSTEM*
DETERMINE THE NUMBER OF OPERATORS
LAYOUT DESIGN*
DETERMINE SUBSETS OF PART TYPES
PRODUCTION PLANNING AND SCHEDULING*
DETERMINE BATCH SIZES
DETERMINE BUFFER SIZES
SIMULATION OF THE SYSTEM
DETERMINE PRIORITIES OF THE PARTS
RESULT ANALYSIS USING AI TECHNIQUES TO GUIDE THE SEARCH
OVERALL PERFORMANCE EVALUATION
THE INPUT OF DATA CHANGES
ROBUSTIC ANALYSIS
OLD LAYOUT
BROWN FIELD SITE
OPTIMISED SOLUTIONS
* SUB-OPTIMISATION PROBLEMS

Figure 6-1 The integration of manufacturing system design

References

Adil, GK, Rajamani, D, and Strong, D, (1996) Cell-formation considering alternate routeings, *International Journal of Production Research*, **34**(5), 1361-1380.

Adil, GK, Rajamani, D, and Strong, D, (1997) Assignment allocation and simulated annealing algorithms for cell formation, *IIE Transactions*, **29**(1), 53-67.

Afentakis, P, Millen, RA, and Solomon, M, (1990) Dynamic layout strategies for flexible manufacturing systems, *International Journal of Production Research*, **28**(2), 311-323.

Agarwal, A, and Sarkis, J, (1998) Review and analysis of comparative performance studies on functional and cellular manufacturing layouts, *Computers & Industrial Engineering*, **34**(1), 77-89.

Akturk, MS, (1996) A note on the within-cell layout problem-based on operation sequences, *Production Panning & Control*, **7**(1), 99-103.

Akturk, MS, and Balkose, HO, (1996) Part-machine grouping using a multiobjective cluster-analysis, *International Journal of Production Research*, **34**(8), 2299-2315.

Al-Hakim, LA, (1992) A modified procedure for converting a dual graph to a block layout, *International Journal of Production Research*, **30**(10), 2467-2476.

Aljaber, N, Baek, W, and Chen, CL, (1997) A tabu search approach to the cell formation problem, *Computers & Industrial Engineering*, **32**(1), 169-185.

Allenbach, R, and Werner, M, (1990) Facility layout program, *Computers & Industrial Engineering*, **19**(1-4), 290-293.

Alsultan, KS, (1997) A hard clustering approach to the part family formation problem, *Production Panning & Control*, **8**(3), 231-236.

Alsultan, KS, and Bozer, YA, (1998) Network configuration and machine layout in fixed-path material handling systems, *Annals of Operations Research*, **76**, 21-54.

Amirahmadi, F, and Choobineh, F, (1992) A heuristic grouping procedure for component family formation in a cellular manufacturing environment, *Computers & Industrial Engineering*, **23**(1-4), 69-72.

Aneke, NAG, and Carrie, AS, (1986) A design technique for the layout of multi-product flowlines, *International Journal of Production Research*, **24**(3), 471-481.

Apple, JM, (1977) *Plant Layout and Material Handling*, Wiley, New York.

Appleton, E, and Williams, DJ, (1987) *Industrial Robot Applications*, Open University Press, Milton Keynes.

Arinze, B, Banerjee, S, and Sylla, C, (1989) Knowledge based layout and material handling - a methodology for knowledge based decision support for facilities layout planning, *Computers & Industrial Engineering*, **17**, 31-36.

Arizono, I, Kato, M, Yamamoto, A, and Ohta, H, (1995) A new stochastic neural-network model and its application to grouping parts and tools in flexible manufacturing systems, *International Journal of Production Research*, **33**(6), 1535-1548.

Armour, GC, and Buffa, ES, (1963) A heuristic algorithm and simulation approach to relative location of facilities, *Management Science*, **9**(1), 294-309.

Arvindh, B, and Irani, SA, (1994) Cell-formation - the need for an integrated solution of the subproblems, *International Journal of Production Research*, **32**(5), 1197-1218.

Babu, AJG, and Yao, DY, (1996) An expert facility layout system - an object-oriented approach, *International Journal of Computer Applications in Technology*, **9**(2-3), 131-143.

Badiru, AB, and Arif, A, (1996) Flexpert - facility layout expert-system using fuzzy linguistic relationship codes, *IIE Transactions*, **28**(4), 295-308.

Balakrishnan, J, and Cheng, CH, (1998) Dynamic layout algorithms: a state-of-the-art survey, *Omega-International Journal of Management Science*, **26**(4), 507-521.

Balasubramanian, KN, and Panneerselvam, R, (1993) Covering technique-based algorithm for machine grouping to form manufacturing cells, *International Journal of Production Research*, **31**(6), 1479-1504.

Ballakur, A, and Steudel, HJ, (1987) A within-cell utilization based heuristic for designing cellular manufacturing systems, *International Journal of Production Research*, **25**(5), 639-665.

Banerjee, P, Montreuil, B, Moodie, CL, and Kashyap, RL, (1990) A qualitative reasoning-based interactive optimization methodology for layout design, *Proceedings of 1990 International Industrial Engineering Conference*, 230-235.

Banerjee, P, and Nof, SY, (1994) Knowledge-based facility planning - a review and a framework, *Journal of Intelligent Manufacturing*, **5**(6), 399-409.

Banerjee, P, and Zhou, Y, (1995) Facilities layout design optimization with single-loop material flow path configuration, *International Journal of Production Research*, **33**(1), 183-203.

Banerjee, P, Zhou, Y, and Montreuil, B, (1997) Genetically assisted optimïzation of cell layout and material flow path skeleton, *IIE Transactions*, **29**(4), 277-291.

Basu, A, Hyer, N, and Shtub, A, (1995) An expert-system based approach to manufacturing cell design, *International Journal of Production Research*, **33**(10), 2739-2755.

Bazaraa, MS, (1975) Computerised layout design: a brand and bound approach, *IIE Transaction*, **7**(4), 432-437.

Benarieh, D, and Chang, PT, (1994) An extension to the p-median group technology algorithm, *Computers & Operations Research*, **21**(2), 119-125.

Benjamin, PC, Erraguntla, M, and Mayer, RJ, (1995) Using simulation for robust system-design, *Simulation*, **65**(2), 116-128.

Benson, B, and Foote, BL, (1997) DoorFAST: a constructive procedure to optimally layout a facility including aisles and door locations based on an aisle flow distance metric, *International Journal of Production Research*, **35**(7), 1825-1842.

Bischak, DP, and Stevens, KB, (1995) An evaluation of the tandem configuration automated guided vehicle system, *Production Planning & Control*, **6**(5), 438-444.

Black, JT, (1991) *The Design of the Factory with a Future*, McGraw-Hill, New York.

Bland, JA, and Dawson, GP, (1991) Tabu search and design optimization, *Computer-Aided Design*, **23**(3), 195-201.

Bland, JA, and Dawson, GP, (1994) Large-scale layout of facilities using a heuristic hybrid algorithm, *Applied Mathematical Modelling*, **18**(9), 500-503.

Boctor, FF, (1991) A linear formulation of the machine-part cell-formation problem, *International Journal of Production Research*, **29**(2), 343-356.

Boctor, FF, (1996) The minimum-cost, machine-part cell-formation problem, *International Journal of Production Research*, **34**(4), 1045-1063.

Boswell, SG, (1992) TESSA - a new greedy heuristic for facilities layout planning, *International Journal of Production Research*, **30**(8), 1957-1968.

Bozer, YA, and Srinivasan, MM, (1991) Tandem configuration for automated guided vehicle systems and the analysis of single vehicle loops, *IIE Transactions*, **23**(1), 72-82.

Bozer, YA, and Meller, RD, (1997) A reexamination of the distance-based facility layout problem, *IIE Transactions*, **29**(7), 549-560.

Bozer, YA, Meller, RD, and Erlebacher, SJ, (1994) An improvement-type layout algorithm for multiple floor facilities, *Management Science*, **40**, 918-932.

Braglia, M, (1997) Heuristics for single-row layout problems in flexible manufacturing problems, *Production Panning & Control*, **8**(6), 558-567.

Bramham, J, and Appleton, E, (1995) The use of optimisation techniques in the design of manufacturing systems, *ICAM'95 International Conference in Advanced Manufacturing*, University of Sunderland, Sept., 1995.

Browne, J, Dubois, D, Rathmill, K, Sethi, SP, and Stecke, KE, (1984) Classification of flexible manufacturing systems, *The FMS Magazine*, **2**(2), 114-117.

Buckles, BP, and Petry, FE, (1992) *Genetic Algorithms*, IEEE Computer Society Press, Los Alamitos, Calif.

Burbidge, JL, (1963) Production flow analysis, *The Production Engineer*, **42**(12), 742-752.

Burbidge, JL, (1971a) Production flow analysis, *The Production Engineer*, **50**, 139-152.

Burbidge, JL, (1971b) *The Production Planning*, Hieneman, London.

Burbidge, JL, (1974) Production flow analysis on the computer, *Production Engineer*, **53**.

Burbidge, JL, (1975) *The Introduction of Group Technology*, Hieneman, London.

Burke, L, and Kamal, S, (1995) Neural networks and the part family machine group formation problem in cellular manufacturing - a framework using fuzzy art, *Journal of Manufacturing Systems*, **14**(3), 148-159.

Cagan, J, Degentesh, D, and Yin, S, (1998) A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout, *Computer-Aided Design*, **30**(10), 781-790.

Cambron, KE, and Evans, GW, (1991) Layout design using the analytic hierarchy process, *Computers & Industrial Engineering*, **20**(2), 211-229.

Carre, B, (1979) *Graphs and Networks*, Clarendon Press, Oxford.

Carrie, AS, (1975) The layout of multi-product-lines, *International Journal of Production Research*, **13**(6), 541-557.

Chan, KC, and Tansri, H, (1994) A study of genetic crossover operations on the facilities layout problem, *Computers & Industrial Engineering*, **26**(3), 537-550.

Chandrasekharan, MP, and Rajagopalan, R, (1986) An ideal seed non-hierarchical clustering algorithm for cellular manufacturing, *International Journal of Production Research*, **24**(2), 451-464.

Chandrasekharan, MP, and Rajagopalan, R, (1987) ZODIAC - an algorithm for concurrent formation of part-families and machine-cells, *International Journal of Production Research*, **25**(6), 835-850.

Chang, RI, and Hsiao, PY, (1995) VLSI cell placement on arbitrarily-shaped rectilinear regions using neural networks with calibration nodes, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E**78**a(12), 1777-1784.

Chen, DS, Chen, HC, and Park, JM, (1996) An improved art neural-net for machine cell-formation, *Journal of Materials Processing Technology*, **61**(1-2), 1-6.

Chen, LH, and Chen, YH, (1995) A computer-simulation-oriented design procedure for a robust and feasible job-shop manufacturing system, *Journal of Manufacturing Systems*, **14**(1), 1-10.

Chen, SJ, and Cheng, CS, (1995) A neural-network-based cell-formation algorithm in cellular manufacturing, *International Journal of Production Research*, **33**(2), 293-318.

Cheng, Ch, (1992) Algorithms for grouping machine groups in group technology, *Omega International Journal of Management*, **20**(4), 493-501.

Cheng, CH, Madan, MS, and Motwani, J, (1996a) Knapsack-based algorithms for designing cellular manufacturing systems, *Journal of the Operational Research Society*, **47**(12), 1468-1476.

Cheng, RW, and Gen, M, (1998) Loop layout design problem in flexible manufacturing systems using genetic algorithms, *Computers & Industrial Engineering*, **34**(1), 53-61.

Cheng, RW, Gen, MS, and Tozawa, T, (1996b) Genetic algorithms for designing loop layout manufacturing systems, *Computers & Industrial Engineering*, **31**(3-4), 587-591.

Chhajed, D, Montreuil, B, and Lowe, TJ, (1992) Flow network design for manufacturing system layout, *European Journal of Operational Research*, **57**, 145-161.

Chiang, WC, and Chiang, C, (1998) Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation, *European Journal of Operational Research*, **106**(2-3), 457-488.

Chiang, WC, and Kouvelis, P, (1996) An improved tabu search heuristic for solving facility layout design- problems, *International Journal of Production Research*, **34**(9), 2565-2585.

Chittratanawat, S, and Noble, JS, (1999) An integrated approach for facility layout, p/d location and material handling system design, *International Journal of Production Research*, **37**(3), 683-706.

Choi, HG, Kwon, HJ, and Lee, J, (1994) Traditional and tandem AGV system layouts - a simulation study, *Simulation*, **63**(2), 85-93.

Chong, K, and Sahni, S, (1993) Optimal realizations of floorplans, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **12**(6), 793-801.

Chow, WS, and Hawaleshka, O, (1993) Minimizing intercellular part movements in manufacturing cell- formation, *International Journal of Production Research*, **31**(9), 2161-2170.

Chu, CH, (1997) An improved neural network for manufacturing cell formation, *Decision Support Systems*, **20**(4), 279-295.

Chung, YK, (1999) Application of a cascade BAM neural expert system to conceptual design for facility layout, *Computers & Mathematics with Applications*, **37**(1), 95-110.

Chwif, L, Barretto, MRP, and Moscato, LA, (1998) A solution to the facility layout problem using simulated annealing, *Computers in Industry*, **36**(1-2), 125-132.

Cimikowski, R, and Mooney, E, (1997) Proximity-based adjacency determination for facility layout, *Computers & Industrial Engineering*, **32**(2), 341-349.

Crama, Y, and Oosten, M, (1996) Models for machine-part grouping in cellular manufacturing, *International Journal of Production Research*, **34**(6), 1693-1713.

Dagli, C, and Huggahalli, R, (1995) Machine-part family formation with the adaptive resonance theory paradigm, *International Journal of Production Research*, **33**(4), 893-913.

Dahel, NE, (1995) Design of cellular manufacturing systems in tandem configuration, *International Journal of Production Research*, **33**(8), 2079-2095.

De-Guzman, MC, Prabhu, N, and Tanchoco, JMA, (1997) Complexity of the AGV shortest path and single-loop guide path layout problems, *International Journal of Production Research*, **35**(8), 2083-2092.

Delmaire, H, Langevin, A, and Riopel, D, (1997) Skeleton-based facility layout design using genetic algorithms, *Annals of Operations Research*, **69**, 85-104.

Demes, GH, Fenves, SJ, Grossmann, IE, Hendrickson, CT, Mitchell, TM, Prinz, FB, Siewiorek, DP, Subrahmanian, E, Talukdar, S, and Westerberg, AW, (1993) The engineering design research center of Carnegie Mellon University, *Proceedings of the IEEE*, **81**(1).

Deutsch, SJ, Freeman, SF, and Helander, M, (1998) Manufacturing cell formation using an improved p-median model, *Computers & Industrial Engineering*, **34**(1), 135-146.

Domschke, W, and Krispin, G, (1997) Location and layout planning - a survey, *OR Spektrum*, **19**(3), 181-194.

Drezner, Z, (1987) An heuristic procedure for the layout of a large number of facilities, *Management Science*, **33**, 907-915.

Dutta, HK, and Sahu, S, (1982) A multigoal heuristic for facilities design problem: MUGHAI, *International Journal of Production Research*, **20**, 147-154.

Eades, P, Foulds, LR, and Giffin, J, (1982) An efficient heuristic of identifying a maximum weight planar subgraph, in Billington, EJ *et al* (eds.), *Combinatorial Mathematics IX. Lecture Notes in Mathematics, 952*, Springer, Berlin, 239-251.

Enke, D, Ratanapan, K, and Dagli, C, (1998) Machine-part family formation utilizing an art1 neural network implemented on a parallel neuro-computer, *Computers & Industrial Engineering*, **34**(1), 189-205.

Farrington, PA, and Nazemetz, JW, (1998) Evaluation of the performance domain of cellular and functional layouts, *Computers & Industrial Engineering*, **34**(1), 91-101.

Flanders, RE, (1925) Design manufacture and production control of a standard machine, *Transactions of ASME*, **46**, 691-738.

Foulds, LR, (1981) *Optimization Techniques: an introduction*, Springer-Velag, New York.

Foulds, LR, and Robinson, DF, (1978) Graph theoretic heuristics for the plant layout problem, *International Journal of Production Research*, **16**, 27-37.

Frazier, GV, and Gaither, N, (1991) Seed selection procedures for cell-formation heuristics, *International Journal of Production Research*, **29**(11), 2227-2237.

Fu, MC, and Kaku, BK, (1997) Minimizing work-in-process and material handling in the facilities layout problem, *IIE Transactions*, **29**(1), 29-36.

Gabbert, P, and Brown, DE, (1989) Knowledge-based computer-aided-design of materials handling systems, *IEEE Transactions on Systems Man and Cybernetics*, **19**(2), 188-196.

Gaither, N, (1992) *Production and Operations Management*, 5th ed., The Dryden Press, New York.

Gallagher, CC, and Knight, WA, (1973) *Group Technology*, Butterworths, London.

Gallagher, CC, and Knight, WA, (1986) *Group Technology Production Methods in Manufacture*, Ellis Horwood, New York.

Garey, MR, and Johnson, DS, (1979) *Computers and Intractability: a guide to the theory of NP-completeness*, W. H. Freeman, New York.

Giffin, JW, (1984) Graph theoretic techniques for facilities layout, *PhD Dissertation*, University of Canterbury, New Zealand.

Glover, F, (1986) Future paths for integer programming and links to artificial intelligence, *Computers & Operation Research*, **5**, 533-549.

Gomory, RE, and Hu, TC, (1961) Multi-terminal network flows, *Journal of SIAM*, **9**, 551-570.

Gravel, M, Nsakanda, AL, and Price, W, (1998) Efficient solutions to the cell-formation problem with multiple routings via a double-loop genetic algorithm, *European Journal of Operational Research*, **109**(2), 286-298.

Green, R, and Al-Hakim, LA, (1985) A heuristic for facilities layout planning, *Omega*, **13**, 469-474.

Gupta, YP, Gupta, MC, Kumar, A, and Sundram, C, (1995) Minimizing total intercell and intracell moves in cellular manufacturing - genetic algorithm approach, *International Journal of Computer Integrated Manufacturing*, **8**(2), 92-101.

Gupta, Y, Gupta, M, Kumar, A, and Sundaram, C, (1996) A genetic algorithm-based approach to cell composition and layout design-problems, *International Journal of Production Research*, **34**(2), 447-482.

Halmos, PR, (1960) *Naive Set Theory*, D. Van Nostrand Company.

Harhalakis, G, Nagi, R, and Proth, JM, (1990) An efficient heuristic in manufacturing cell-formation for group technology applications, *International Journal of Production Research*, **28**(1), 185-198.

Harmonosky, CM, and Tothero, GK, (1992) A multifactor plant layout methodology, *International Journal of Production Research*, **30**(8), 1773-1789.

Harrington, J, (1984) *Understanding the Manufacturing Process: key to successful CAD/CAM implementation*, Marcel Dekker, New York.

Hassan, MMD, (1994) Machine layout problem in modern manufacturing facilities, *International Journal of Production Research*, **32**(11), 2559-2584.

Hassan, MMD, (1995) Layout design in group technology manufacturing, *International Journal of Production Economics*, **38**(2-3), 173-188.

Hassan, MMD, and Hogg, GL, (1985) A construction algorithm for the selection and assignment of material handling equipment, *International Journal of Production Research*, **23**, 381-392.

Hassan, MMD, and Hogg, GL, (1989) On converting a dual graph into a block layout, *International Journal of Production Research*, **27**, 1149-1160.

Hassan, MMD, Hogg, GL, and Smith, DR, (1986) SHAPE: a construction algorithm for area placement evaluation, *International Journal of Production Research*, **24**(5), 1283-1295.

Heragu, SS, (1989) Knowledge-based approach to machine layout, *Computers & Industrial Engineering*, **17**(1-4), 37-42.

Heragu, SS, (1994) Group technology and cellular manufacturing, *IEEE Transactions on Systems Man and Cybernetics*, **24**(2), 203-215.

Heragu, SS, and Kakuturi, SR, (1997) Grouping and placement of machine cells, *IIE Transactions*, **29**(7), 561-571.

Heragu, SS, and Kusiak, A, (1988) Machine layout problem in flexible manufacturing systems, *Operations Research*, **36**(2), 258-268.

Heragu, SS, and Kusiak, A, (1990) Machine layout: an optimization and knowledge-based approach, *International Journal of Production Research*, **28**(4), 615-635.

Ho, YC, and Moodie, CL, (1996) Solving cell-formation problems in a manufacturing environment with flexible processing and routeing capabilities, *International Journal of Production Research*, **34**(10), 2901-2923.

Ho, YC, and Moodie, CL, (1998) Machine layout with a linear single-row flow path in an automated manufacturing system, *Journal of Manufacturing Systems*, **17**(1), 1-22.

Holland, JH, (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Horowitz, E, and Sahni, S, (1976) *Fundamentals of Data Structures*, Pitman, Bath.

Houshyar, A, (1991) Computer aided facility layout: an interactive multi-goal approach, *Computers & Industrial Engineering*, **20**(2), 177-186.

Houshyar, A, and White, B, (1997) Comparison of solution procedures to the facility location problem, *Computers & Industrial Engineering*, **32**(1), 77-87.

Hsu, CM, and Su, CT, (1998) Multi-objective machine-component grouping in cellular manufacturing: a genetic algorithm, *Production Panning & Control*, **9**(2), 155-166.

Ioannou, G, and Minis, I, (1998) A review of current research in manufacturing shop design integration, *Journal of Intelligent Manufacturing*, **9**(1), 57-72.

Irani, SA, Cavalier, TM, and Cohen, PH, (1993) Virtual manufacturing cells - exploiting layout design and intercell flows for the machine sharing problem, *International Journal of Production Research*, **31**(4), 791-810.

Irvine, SA, and Rinsma-Melchert, I, (1997) A new approach to the block layout problem, *International Journal of Production Research*, **35**(8), 2359-2376.

Islier, AA, (1998) A genetic algorithm approach for multiple criteria facility layout design, *International Journal of Production Research*, **36**(6), 1549-1569.

Jang, IH, and Rhee, JT, (1997) Generalized machine cell formation considering material flow and plant layout using modified self-organizing feature maps, *Computers & Industrial Engineering*, **33**(3-4), 457-460.

Joines, JA, Culbreth, CT, and King, RE, (1996) Manufacturing cell design - an integer programming-model employing genetic algorithms, *IIE Transactions*, **28**(1), 69-85.

Kamel, M, Ghenniwa, H, and Liu, T, (1994) Machine assignment and part-families formation using group technology, *Journal of Intelligent Manufacturing*, **5**(4), 225-234.

Kamal, S, and Burke, LI, (1996) Fact - a new neural-network-based clustering-algorithm for group technology, *International Journal of Production Research*, **34**(4), 919-946.

Kao, Y, and Moon, YB, (1991) A unified group technology implementation using the backpropagation learning rule of neural networks, *Computers & Industrial Engineering*, **20**(4), 425-437.

Kao, Y, and Moon, YB, (1995) Feature-based memory association for group technology, *Computers & Industrial Engineering*, **29**, 171-175.

Kao, Y, and Moon, YB, (1997) Part family formation by memory association, *International Journal of Advanced Manufacturing Technology*, **13**(9), 649-657.

Kao, Y, and Moon, YB, (1998) Feature-based memory association for group technology, *International Journal of Production Research*, **36**(6), 1653-1677.

Kern, GM, and Wei, JC, (1991) The cost of eliminating exceptional elements in group technology cell-formation, *International Journal of Production Research*, **29**(8), 1535-1547.

Kim, CB, Foote, BL, and Pulat, PS, (1995) Cut-tree construction for facility layout, *Computers & Industrial Engineering*, **28**(4), 721-730.

Kim, KS, and Eom, JK, (1997) An expert system for selection of material handling and storage systems, *International Journal of Industrial Engineering-Applications and Practice*, **4**(2), 81-89.

Kirkpatrick, S, Gelatt, CD, and Vecchi, MP, (1983) Optimization by simulated annealing, *Science*, **220**, 671-680.

Kochhar, JS, Foster, BT, and Heragu, SS, (1998) HOPE: a genetic algorithm for the unequal area facility layout problem, *Computers & Operations Research*, **25**(7-8), 583-594.

Kohonen, T, (1988) An introduction to neural computing, *Neural Network*, **1**, 3-16.

Kouvelis, P, Chiang, WC, and Yu, G, (1995) Optimal-algorithms for row layout problems in automated manufacturing systems, *IIE Transactions*, **27**(1), 99-104.

Kouvelis, P, and Chiang, WC, (1996) Optimal and heuristic procedures for row layout problems in automated manufacturing systems, *Journal of the Operational Research Society*, **47**(6), 803-816.

Kumar, CS, and Chandrasekharan, MP, (1990) Grouping efficacy - a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, *International Journal of Production Research*, **28**(2), 233-243.

Kumar, KR, Kusiak, A, and Vannelli, A, (1986) Grouping of parts and components in flexible manufacturing systems, *European Journal of Operational Research*, **24**, 387-397.

Kumar, KR, and Vannelli, A, (1987) Strategic subcontracting for efficient disaggregated manufacturing, *International Journal of Production Research*, **25**(12), 1715-1728.

Kumara, SRT, Kashyao, RL, and Moodie, CL, (1987) Expert system for industrial facilities layout planning and analysis, *Computers & Industrial Engineering*, **12**(2), 143-152.

Kumara, SRT, Kashyao, RL, and Moodie, CL, (1988) Application of expert systems and pattern recognization methodologies to facilities layout planning, *International Journal of Production Research*, **26**(5), 905-930.

Kusiak, A, (1985) The part families problem in flexible manufacturing systems, *Annals of Operations Research*, **3**, 279-300.

Kusiak, A, (1987a) The generalized group technology concept, *International Journal of Production Research*, **25**(4), 561-569.

Kusiak, A, (1987b) Artificial-intelligence series .8. an expert system for group technology, *Industrial Engineering*, **19**(10), 56-61.

Kusiak, A, (1988) EXGT-s - a knowledge based system for group technology, *International Journal of Production Research*, **26**(5), 887-904.

Kusiak, A, and Heragu, SS, (1987a) Group technology, *Computers in Industry*, **9**(2), 83-91.

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

Langevin, A, Montreuil, B, and Riopel, D, (1994) Spine layout design, *International Journal of Production Research*, **32**(2), 429-442.

Lauther, U, (1980) A min-cut placement algorithm for general cell assemblies based on a graph representation, *Journal of Digital Systems*, **4**, 21-34.

Lee, H, and Garcia-Diaz, A, (1993) A network flow approach to solve clustering problems in group technology, *International Journal of Production Research*, **31**(3), 603-612.

Lee, H, and Garcia-Diaz, A, (1996) Network flow procedures for the analysis of cellular manufacturing systems, *IIE Transactions*, **28**(4), 333-345.

Lee, MK, Luong, HS, and Abhary, K, (1997) A genetic algorithm based cell design considering alternative routing, *Computer Integrated Manufacturing Systems*, **10**(2), 93-108.

Lee, S, and Wang, HP, (1992) Manufacturing cell formation: a dual-objective simulated annealing approach, *International Journal of Advanced Manufacturing Technology*, **7**, 314-320.

Lee, SD, and Chen, YL, (1997) A weighted approach for cellular manufacturing design: minimizing intercell movement and balancing workload among duplicated machines, *International Journal of Production Research*, **35**(4), 1125-1146.

Lengauer, T, (1990) *Combinatorial Algorithms for Integrated Circuit Layout*, Wiley, New York.

Leung, J, (1992) A new graph theoretic heuristic for facility layout, *Management Science*, **38**, 554-605.

Logendran, R, (1991a) Impact of sequence of operations and layout of cells in cellular manufacturing, *International Journal of Production Research*, **29**(2), 375-390.

Logendran, R, (1991b) Effect of the identification of key machines in the cell-formation problem of cellular manufacturing systems, *Computers & Industrial Engineering*, **20**(4), 439-449.

Logendran, R, (1992) Simultaneous machine-part grouping approach in manufacturing cells, *Computers & Industrial Engineering*, **23**(1-4), 77-80.

Logendran, R, and Ko, CS, (1997) Manufacturing cell formation in the presence of flexible cell locations and material transporters, *Computers & Industrial Engineering*, **33**(3-4), 545-548.

Logendran, R, Ramakrishna, P, and Sriskandarajah, C, (1994) Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans, *International Journal of Production Research*, **32**(2), 273-297.

Mahadevan, B, and Narendran, TT, (1994) A hybrid modeling approach to the design of an AGV-based material handling-system for an FMS, *International Journal of Production Research*, **32**(9), 2015-2030.

Malmborg, CJ, (1994) A heuristic model for simultaneous storage space allocation and block layout planning, *International Journal of Production Research*, **32**(3), 517-530.

Manda, BS, and Palekar, US, (1997) Recent advances in the design and analysis of material handling systems, *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, **119**(4b), 841-848.

Massay, LL, Udoka, SJ, and Ram, B, (1995) Robotic system-design - a hierarchical simulation-based approach, *Computers & Industrial Engineering*, **29**, 699-703.

Matson, JO, Mellichamp, JM, and Swaminathan, SR, (1992) EXCITE - expert consultant for in-plant transportation equipment, *International Journal of Production Research*, **30**(8), 1969-1983.

Mavridou, TD, and Pardalos, PM, (1997) Simulated annealing and genetic algorithms for the facility layout problem: a survey, *Computational Optimization and Applications*, **7**(1), 111-126.

Mayer, RJ, and Benjamin, PC, (1992) Using the Taguchi paradigm for manufacturing system-design using simulation experiments, *Computers & Industrial Engineering*, **22**(2), 195-209.

McAuley, J, (1972) Machine grouping for efficiency production, *Production Engineer*, **51**, 53-57.

McHugh, JA, (1990) *Algorithmic Graph Theory*, Prentice Hall, Englewood Cliffs, NJ.

Meller, RD, and Bozer, YA, (1996) A new simulated annealing algorithm for the facility layout problem, *International Journal of Production Research*, **34**(6), 1675-1692.

Meller, RD, and Gau, KY, (1996a) The facility layout problem - recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, **15**(5), 351-366.

Meller, RD, and Gau, KY, (1996b) Facility layout objective functions and robust layouts, *International Journal of Production Research*, **34**(10), 2727-2742.

Merker, J, and Wascher, G, (1997) Two new heuristic algorithms for the maximal planar layout problem, *OR Spektrum*, **19**(2), 131-137.

Michalewicz, Z, (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer-Verlag, Berlin.

Miltenburg, J, and Zhang, W, (1991) A comparative evaluation of nine well-known algorithms for solving the cell formation in group technology, *Journal of Operations Management*, **10**(1), 44-72.

Miniek, E, (1978) *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York.

Mitchell, M, (1996) *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, MA.

Mitrofanov, SP, (1958) *The Scientific Principles of Group Technology*, Leningrad.

Montreuil, B, and Ratliff, HD, (1989) Utilizing cut trees as design skeletons for facility layout, *IIE Transactions*, **21**, 136-143.

Moon, YB, (1992) Establishment of a neurocomputing model for part family machine group identification, *Journal of Intelligent Manufacturing*, **3**(3), 173-182.

Moon, YB, and Chi, SC, (1992) Generalized part family formation using neural network techniques, *Journal of Manufacturing Systems*, **11**(3), 149-159.

Mosier, CT, (1989) An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell-formation problem, *International Journal of Production Research*, **27**(10), 1811-1835.

Mosier, CT, Yelle, J, and Walker, G, (1997) Survey of similarity coefficient based methods as applied to the group technology configuration problem, *Omega-International Journal of Management Science*, **25**(1), 65-79.

Muther, R, and Wheeler, JD, (1962) Simplified systematic layout planning, *Factory*, **120**(8, 9, 10), 68-77, 111-119, 101-113.

Nagi, R, Harhalakis, G, and Proth, JM, (1990) Multiple routeings and capacity considerations in group technology applications, *International Journal of Production Research*, **28**(12), 2243-2257.

Nair, GJ, and Narendran, TT, (1998) CASE: a clustering algorithm for cell formation with sequence data, *International Journal of Production Research*, **36**(1), 157-179.

Ng, S, (1993) Worst-case analysis of an algorithm for cellular manufacturing, *European Journal of Operational Research*, **69**(3), 384-398.

Nissen, V, (1994) Solving the quadratic assignment problem with clues from nature, *IEEE Transactions on Neural Networks*, **5**(1), 66-72.

Noble, JS, Klein, CM, and Midha, A, (1998) An integrated model of the material handling system and unit load design problem, *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, **120**(4), 802-806.

Nugent, CE, Vollman, RE, and Ruml, J, (1968) An experimental comparison of techniques for the assignment of facilities to locations, *Operations Research*, **16**, 150-173.

Okada, K, Yamanouchi, T, and Kambe, T, (1998) Rectilinear shape formation method on block placement, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, **E81a**(3), 446-454.

Okogbaa, OG, Chen, MT, Changchit, CW, and Shell, RL, (1992) Manufacturing system cell-formation and evaluation using a new intercell flow reduction heuristic, *International Journal of Production Research*, **30**(5), 1101-1118.

Oliver, IM, Smith, DJ, and Holland, JRC, (1987) A study of permutation crossover operations on the travelling salesman problem, *Proceedings of the 2nd International Conference of Genetic Algorithms*, 224-230.

Otten, R, (1982) Automatic floorplan design, *Proceedings of the 19th Design Automation Conference*, 261-267.

Otten, R, (1983) Efficient floorplan design, *Proceedings of International Conference on Computer Design*, 499-453.

Otten, R, (1988) Graphs in floorplan design, *International Journal of Circuit Theory Applications*.

Pan, PC, and Liu, CL, (1995) Area minimization for floorplans, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **14**(1), 123-132.

Pan, PC, Shi, WP, and Liu, CL, (1996) Area minimization for hierarchical floorplans, *Algorithmica*, **15**(6), 550-571.

Park, YB, (1996) ICMESE: intelligent consultant system for material handling equipment selection and evaluation, *Journal of Manufacturing Systems*, **15**(5), 325-333.

Parker, RG, and Rardin, RL, (1988) *Discrete Optimization*, Academic Press, Boston.

Parsaei, HR, and Galbiati, LJ, (1987) Facilities planning and design with microcomputers, *Computers & Industrial Engineering*, **13**(1-4), 332-335.

Pearl, J, (1984) *Heuristics: intelligent search strategies for computer problem solving*, Addison-Wesley, Reading, MA.

Peters, BA, and Yang, T, (1997) Integrated facility layout and material handling system design in semiconductor fabrication facilities, *IEEE Transactions on Semiconductor Manufacturing*, **10**(3), 360-369.

Pilot, T, and Knosala, R, (1998) The application of neural networks in group technology, *Journal of Materials Processing Technology*, **78**(1-3), 150-155.

Poole, D, Mackworth, A, and Goebel, R, (1998) *Computational Intelligence*, Oxford University Press, Oxford.

Rajagopalan, R, and Batra, JL, (1975) Design of cellular production systems: a graph-theoretic approach, *International Journal of Production Research*, **13**(6), 567-579.

Rajasekharan, M, Peters, BA, and Yang, T, (1998) A genetic algorithm for facility layout design in flexible manufacturing systems, *International Journal of Production Research*, **36**(1), 95-110.

Rebaudengo, M, and Reorda, MS, (1996) GALLO: a genetic algorithm for floorplan area optimization, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **15**(8), 943-951.

Reisman, A, Kumar, A, Motwani, J, and Cheng, CH, (1997) Cellular manufacturing: a statistical review of the literature (1965- 1995), *Operations Research*, **45**(4), 508-520.

Rich, E, and Knight, K, (1991) *Artificial Intelligence*, McGraw-Hill, New York.

Riggs, JL, (1987) *Production Systems: planning, analysis and control*, Wiley, New York.

Rim, SC and Bozer, YA, (1991) The bidirectional circular layout problem, Working Paper, Industrial Engineering Program, FU-20, University of Washington, Seattle, WA, 98195.

Rinsma, F, Giffin, JW, and Robinson, DF, (1990) Orthogonal floorplans from maximal planar graphs, *Environment and Planning B: Planning and Design*, **17**, 57-71.

Rosenblatt, MJ, (1979) The facility layout problems: a multi-goal approach, *International Journal of Production Research*, **17**(4), 323-332.

Rosenblatt, MJ, and Sinuany-Stern, Z, (1986) A discrete efficient frontier approach to the plant layout problem, *Material Flow*, **2**(4), 277-281.

Sahni, S, and Gonzalez, T, (1976) P-complete approximation problem, *Journal of Association for Computer Machinery*, **23**(3), 555-565.

Salvendy, G, (1982) *Handbook of Industrial Engineering*, John Wiley & Sons, New York.

Sarker BR, and Mondal, S, (1999) Grouping efficiency measures in cellular manufacturing: a survey and critical review, *International Journal of Production Research*, **37**(2), 285-314.

Sarker, BR, (1996) The resemblance coefficients in group technology - a survey and comparative-study of relational metrics, *Computers & Industrial Engineering*, **30**(1), 103-116.

Sarker, BR, Wilhelm, WE, and Hogg, GL, (1994) Measures of backtracking and bidirectional flow in one-dimensional machine location-problems, *Production Panning & Control*, **5**(3), 282-291.

Sarker, BR, and Xu, Y, (1998) Operation sequences-based cell formation methods: a critical survey, *Production Panning & Control*, **9**(8), 771-783.

Savsar, M, (1991) Flexible facility layout by simulation, *Computers & Industrial Engineering*, **20**(1), 155-165.

Schroeder, RG, (1993) *Operations Management*, McGraw-Hill, New York.

Sedgewick, R, and Flajolet, P, (1996) *An Introduction of Analysis of Algorithms*, Addison-Wesley, Reading, MA.

Seehof, JM, and Evans, WO, (1967) Automatic layout design program, *The Journal of Industrial Engineering*, **18**(12), 690-695.

Seifoddini, H, (1990) Machine-component group-analysis versus the similarity coefficient method in cellular manufacturing applications, *Computers & Industrial Engineering*, **18**(3), 333-339.

Seifoddini, H, and Djassemi, M, (1991) The production data based similarity coefficient versus Jaccards similarity coefficient, *Computers & Industrial Engineering*, **21**(1-4), 263-266.

Seifoddini, H, and Djassemi, M, (1995) Merits of the production volume based similarity coefficient in machine cell-formation, *Journal of Manufacturing Systems*, **14**(1), 35-44.

Seifoddini, H, and Wolfe, PM, (1986) Application of the similarity coefficient method in group technology, *IIE Transactions*, **18**, 271-227.

Selim, HM, Askin, RG, and Vakharia, AJ, (1998) Cell formation in group technology: review, evaluation and directions for future research, *Computers & Industrial Engineering*, **34**(1), 3-20.

Selvam, RP, and Balasubramanian, KN, (1985) Algorithmic grouping of operation sequences, *Engineering Costs and Production Economics*, **9**(1-3), 125-134.

Shafer, SM, and Rogers, DF, (1993a) Similarity and distance measures for cellular manufacturing .1. a survey, *International Journal of Production Research*, **31**(6), 1133-1142.

Shafer, SM, and Rogers, DF, (1993b) Similarity and distance measures for cellular manufacturing .2. an extension and comparison, *International Journal of Production Research*, **31**(6), 1315-1326.

Shang, JS, and Tadikamalla, PR, (1998) Multicriteria design and control of a cellular manufacturing system through simulation and optimization, *International Journal of Production Research*, **36**(6), 1515-1528.

Shanker, K, and Agrawal, AK, (1997) Models and solution methodologies for the generalized grouping problem in cellular manufacturing, *International Journal of Production Research*, **35**(2), 513-538.

Shi, WP, (1996) A fast algorithm for area minimization of slicing floorplans, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **15**(12), 1525-1532.

Sinriech, D, (1995) Network design models for discrete material flow systems - a literature-review, *International Journal of Advanced Manufacturing Technology*, **10**(4), 277-291.

Sinriech, D, and Tanchoco, JMA, (1994) SFT - segmented flow topology, in Tanchoco, JMA (ed.), *Material Flow Systems in Manufacturing*, Chapman & Hall, London.

Sinriech, D, and Tanchoco, JMA, (1995) An introduction to the segmented flow approach for discrete material flow systems, *International Journal of Production Research*, **33**(12), 3381-3410.

Sirinaovakul, B, and Thajchayapong, P, (1996) An analysis of computer-aided facility layout techniques, *International Journal of Computer Integrated Manufacturing*, **9**(4), 260-264.

Skorin-Kapov, J, (1990) Tabu search applied to the quadratic assignment problem, *ORSA Journal on Computing*, **2**(33-45).

Sofianopoulou, S, (1997) Application of simulated annealing to a linear model for the formulation of machine cells in group technology, *International Journal of Production Research*, **35**(2), 501-511.

Srinivasulu, M, and Kumar, S, (1997) Expert system for group technology, *Defence Science Journal*, **47**(1), 83-88.

Stockmeyer, L, (1983) Optimal orientations of cells in slicing floorplan designs, *Information and Control*, **57**(2-3), 91-101.

Su, CT, and Hsu, CM, (1998) Multi-objective machine-part cell formation through parallel simulated annealing, *International Journal of Production Research*, **36**(8), 2185-2207.

Sun, D, Lin, L, and Batta, R, (1995) Cell-formation using tabu search, *Computers & Industrial Engineering*, **28**(3), 485-494.

Suresh, G, Vinod, VV, and Sahu, S, (1995) A genetic algorithm for facility layout, *International Journal of Production Research*, **33**(12), 3411-3423.

Szykman, S, Cagan, J, and Weisser, P, (1998) An integrated approach to optimal three dimensional layout and routing, *Journal of Mechanical Design*, **120**(3), 510-512.

Taboun, SM, Sankaran, S, and Bhole, S, (1991) Comparison and evaluation of similarity measures in group technology, *Computers & Industrial Engineering*, **20**(3), 343-353.

Tam, KY, (1990) An operation sequence based similarity coefficient for part families formations, *Journal of Manufacturing Systems*, **9**(1), 55-68.

Tam, KY, (1992) A simulated annealing algorithm for allocating space to manufacturing cells, *International Journal of Production Research*, **30**(1), 63-87.

Tam, KY, and Chan, SK, (1998) Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings, *International Journal of Production Research*, **36**(12), 3253-3272.

Tam, KY, and Li, SG, (1991) A hierarchical approach to the facility layout problem, *International Journal of Production Research*, **29**(1), 165-184.

Tang, C, and Abdel-Malek, LL, (1996) A framework for hierarchical interactive generation of cellular layout, *International Journal of Production Research*, **34**(8), 2133-2162.

Tarjan, RE, (1983) *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia.

Tate, DM, and Smith, AE, (1995) Unequal-area facility layout by genetic search, *IIE Transactions*, **27**(4), 465-472.

Teitz, MB, and Bart, P, (1968) Heuristic methods for estimating the generalized vertex median of a weighted graph, *Operations Research*, **16**, 955-961.

Tempelmeier, H, and Kuhn, H, (1993) *Flexible Manufacturing Systems*, John Wiley & Sons, New York.

Tompkins, JA, and Spain, JD, (1983) Utilization of spine concept maximizes modularity in facilities planning, *Industrial Engineering*, **15**(3), 34-42.

Tong, X, (1991) SECOT: a sequential construction technique for facility design, *Doctoral Dissertation*, University of Pittsburgh.

Tretheway, SJ, and Foote, BL, (1994) Automatic computation and drawing of facility layouts with logical aisle structures, *International Journal of Production Research*, **32**(7), 1545-1555.

Vakharia, AJ, and Chang, YL, (1997) Cell formation in group technology: a combinatorial search approach, *International Journal of Production Research*, **35**(7), 2025-2043.

Vakharia, AJ, and Wemmerlov, U, (1990) Designing a cellular manufacturing system - a materials flow approach based on operation sequences, *IIE Transactions*, **22**(1), 84-97.

Vakharia, AJ, and Wemmerlov, U, (1995) A comparative investigation of hierarchical clustering techniques and dissimilarity measures applied to the cell formation, *Journal of Operations Management*, **13**, 117-138.

Vannelli, A, and Kumar, KR, (1986) A method for finding minimal bottle-neck cells for grouping part-machine families, *International Journal of Production Research*, **24**, 387-400.

Ventura, JA, Chen, FF, and Wu, CH, (1990) Grouping parts and tools in flexible manufacturing systems production planning, *International Journal of Production Research*, **28**, 1039-1056.

Verma, P, and Ding, FY, (1995) A sequence-based materials flow procedure for designing manufacturing cells, *International Journal of Production Research*, **33**(12), 3267-3281.

Vonderembse, MA, and While, GP, (1991) *Operations Management - concepts, methods, and strategies*, 2nd ed., West, St. Paul.

Wall, M, (1996) Matthew's GAlib, *http://lancet.mit.edu/ga/*.

Wang, J, and Roze, C, (1995) Formation of machine cells and part families in cellular manufacturing - an experimental-study, *Computers & Industrial Engineering*, **29**, 567-571.

Wang, TY, Lin, HC, and Wu, KB, (1998) An improved simulated annealing for facility layout problems in cellular manufacturing systems, *Computers & Industrial Engineering*, **34**(2), 309-319.

Wascher, G, and Merker, J, (1997) A comparative evaluation of heuristics for the adjacency problem in facility layout planning, *International Journal of Production Research*, **35**(2), 447-466.

Watson, KH, and Giffin, JW, (1997) The vertex splitting algorithm for facilities layout, *International Journal of Production Research*, **35**(9), 2477-2492.

Webster, DB, and Reed, R, (1971) A material handling system selection model, *IIE Transactions*, **3**, 13-21.

Wei, JC, (1987) The development and evaluation of a capacity-constrained, multiobjective cell formation heuristic model, *Doctoral Dissertation*, Texas A&M University.

Wei, JC, and Gaither, N, (1990) A capacity constrained multiobjective cell-formation method, *Journal of Manufacturing Systems*, **9**(3), 222-232.

Welgama, PS, and Gibson, PR, (1993) A construction algorithm for the machine layout problem with fixed pick-up and drop-off points, *International Journal of Production Research*, **31**(11), 2575-2590.

Welgama, PS, Gibson, PR, and Alhakim, LAR, (1994) Facilities layout - a knowledge-based approach for converting a dual graph into a block layout, *International Journal of Production Economics*, **33**(1-3), 17-30.

Welgama, PS, and Gibson, PR, (1995a) Computer-aided facility layout - a status-report, *International Journal of Advanced Manufacturing Technology*, **10**(1), 66-77.

Welgama, PS, and Gibson, PR, (1995b) A hybrid knowledge-based optimization system for automated selection of materials handling-system, *Computers & Industrial Engineering*, **28**(2), 205-217.

Welgama, PS, and Gibson, PR, (1996) An integrated methodology for automating the determination of layout and materials handling system, *International Journal of Production Research*, **34**(8), 2247-2264.

Wild, RH, and Pignatiello, JJ, (1991) An experimental-design strategy for designing robust systems using discrete-event simulation, *Simulation*, **57**(6), 358-368.

Wu, N, (1998) A concurrent approach to cell formation and assignment of identical machines in group technology, *International Journal of Production Research*, **36**(8), 2099-2114.

Wu, NQ, and Salvendy, G, (1993) A modified network approach for the design of cellular manufacturing systems, *International Journal of Production Research*, **31**(6), 1409-1421.

Zhang, HC, and Huang, SH, (1995) Applications of neural networks in manufacturing - a state-of-the-art survey, *International Journal of Production Research*, **33**(3), 705-728.

Zhou, M, and Askin, RG, (1998) Formation of general GT cells: an operation-based approach, *Computers & Industrial Engineering*, **34**(1), 147-157.

Ziai, MR, and Sule, DR, (1988) Microcomputer facility layout design, *Computers & Industrial Engineering*, **15**(1-4), 259-263.

Zolfaghari, S, and Liang, M, (1997) An objective-guided oitho-synapse Hopfield network approach to machine grouping problems, *International Journal of Production Research*, **35**(10), 2773-2792.

Zolfaghari, S, and Liang, M, (1998) Machine cell/part family formation considering processing times and machine capacities: a simulated annealing approach, *Computers & Industrial Engineering*, **34**(4), 813-823.

Appendix

Simulation Data

A.1 Problem 1 - A Cell Formation Problem of 10 Parts and 18 Machines (Okogbaa *et al* 1992)

Part Number	Volume	Process Sequence
1	1	9, 16, 5, 13, 10, 17
2	1	3, 6, 11, 9, 16, 5, 1, 12, 7
3	1	1, 12, 7, 2, 14, 4, 13, 10, 17
4	1	3, 6, 11, 1, 12, 7, 18, 8, 15, 9, 16, 5
5	1	3, 6, 11, 13, 10, 17, 1, 12, 7, 2, 14, 4
6	1	1, 12, 7, 13, 10, 17, 3, 6, 11
7	1	9, 16, 5, 2, 14, 4, 18, 8, 15
8	1	1, 12, 7, 13, 10, 17
9	1	2, 14, 4, 18, 8, 15
10	1	3, 6, 11, 2, 14, 4, 18, 8, 15, 1, 12, 7

A.2 Problem2 - A Cell Formation Problem of 20 Parts and 20 Machines (Harhalakis *et al* 1990)

Part Number	Volume	Process Sequence	Part Number	Volume	Process Sequence
1	1	12, 1, 9, 18, 20	11	1	11, 14, 3
2	1	11, 3, 2	12	1	9, 18, 5, 12, 1
3	1	8, 20, 19	13	1	6, 7, 15, 17
4	1	3, 11, 2, 10	14	1	8, 10, 1, 2
5	1	4, 15, 6, 7	15	1	13, 14, 16, 17

6	1	11, 14, 16, 17, 5	16	1	15, 7, 6, 19
7	1	5, 16, 17	17	1	9, 1, 12
8	1	15, 13, 7, 9, 4	18	1	8, 19, 20, 10
9	1	18, 9, 11, 12, 1	19	1	3, 2, 11, 5
10	1	19, 20, 8	20	1	18, 10, 1, 12

A.3 Problem 3 - A Problem of 41 Parts and 30 Machines

A.3.1 Parts and Their Process Sequence (Seifoddini and Djassemi 1995)

Part Number	Volume	Process Sequence	Part Number	Volume	Process Sequence
1	115	20, 30, 19, 29, 8	22	120	29
2	16	10, 23	23	78	3, 22, 10, 12
3	120	20, 30, 19, 29, 9	24	91	12, 4
4	78	14, 25	25	76	4, 13
5	91	14	26	139	4, 16, 14
6	71	6, 16	27	97	7, 26, 17, 18
7	67	17, 4	28	69	4
8	82	8, 28, 27	29	61	19, 8, 28
9	61	29, 8	30	55	29
10	8	22, 1, 11, 2	31	87	3, 22, 1, 21, 23
11	63	2, 12	32	31	3, 22, 1, 2, 21
12	144	3, 22, 2, 21, 10, 23, 12	33	93	1, 11, 2, 21
13	127	29, 9	34	60	7, 26, 18, 5
14	76	18, 8	35	128	28, 4
15	75	27, 4	36	49	7, 26, 17, 5
16	87	7, 26, 17, 18	37	81	15
17	31	5, 15	38	120	13, 24
18	96	2, 14	39	84	3, 22, 1, 11, 2, 10, 23, 12
19	110	12, 13	40	110	3, 22, 21, 12
20	136	12	41	31	1, 11, 2
21	84	30, 19, 29, 9			

A.3.2 The Dimensions of the Machines

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Length	50	20	40	30	30	25	25	30	20	40
Width	30	30	25	45	20	35	50	40	20	30
	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
Length	20	50	35	25	30	45	35	45	25	50
Width	35	50	25	40	20	30	45	35	25	25
	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30
Length	40	20	25	40	30	30	30	35	35	40
Width	40	45	30	35	30	25	40	50	35	25

A.3.3 Intercell Flows

A.3.3.1 Intercell Flows of the 4-Cell Configuration

	C1	C2	C3	C4	PW
RW	1178	847	825	548	0
C1	0	297	0	0	881
C2	0	0	0	0	1354
C3	0	143	0	0	758
C4	0	67	76	0	405

A.3.3.2 Intercell Flows of the 6-Cell Configuration

	C1	C2	C3	C4	C5	C6	PW
RW	825	473	240	487	548	825	0
C1	0	566	96	0	0	0	163
C2	0	0	0	91	0	0	1024
C3	0	0	0	0	0	0	475
C4	0	76	139	0	0	0	573
C5	0	0	0	67	0	76	405
C6	0	0	0	143	0	0	758

A.3.3.3 Intercell Flows of the 8-C ell Configuration

	C1	C2	C3	C4	C5	C6	C7	C8	PW
RW	825	353	436	112	825	487	240	120	0
C1	0	566	0	0	0	0	96	0	163
C2	0	0	0	0	0	91	0	110	718
C3	0	0	0	109	76	67	0	0	184
C4	0	0	0	0	0	0	0	0	221
C5	0	0	0	0	0	143	0	0	758
C6	0	0	0	0	0	0	139	76	573
C7	0	0	0	0	0	0	0	0	475
C8	0	0	0	0	0	0	0	0	306

A.3.3.4 Intercell Flows of the 11-Cell Configuration

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	PW
RW	542	283	353	436	112	380	445	487	169	71	120	0
C1	534	464	78	0	0	0	0	0	0	0	0	0
C2	0	818	488	0	0	0	0	0	96	0	0	163
C3	0	0	550	0	0	0	0	91	0	0	110	718
C4	0	0	0	770	109	0	76	67	0	0	0	184
C5	0	0	0	0	31	0	0	0	0	0	0	221
C6	0	0	0	0	0	554	380	0	0	0	0	0
C7	0	0	0	0	0	0	507	143	0	0	0	758
C8	0	0	0	0	0	0	0	285	0	139	76	573
C9	0	0	0	0	0	0	0	0	78	0	0	404
C10	0	0	0	0	0	0	0	0	139	71	0	71
C11	0	0	0	0	0	0	0	0	0	0	120	306