國立交通大學

工業工程與管理學系

博士論文

單元形成問題之求解模式與演算法

Models and Solution Methods for Cell Formation Problems

研 究 生：張欽智

指導教授：鍾淑馨 博士

吳泰熙 博士

中 華 民 國 九 十 九 年 十二 月

單元形成問題之求解模式與演算法

Models and Solution Methods for Cell Formation Problems

研 究 生：張欽智　　Student：Chin-Chih Chang

指導教授：鍾淑馨　博士　　Advisor：Dr. Shu-Hsing Chung

吳泰熙　博士　　Dr. Tai-Hsi Wu

國 立 交 通 大 學

工業工程與管理學系

博士論文

A Thesis Submitted to
Department of Industrial Engineering and Management
College of Management
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Industrial Engineering

December 2010
Hsinchu, Taiwan, Republic of China

中華民國九十九年十二月

單元形成問題之求解模式與演算法

學 生：張欽智　　指導教授：鍾淑馨　博士

吳泰熙　博士

國 立 交 通 大 學

工業工程與管理學系

摘要

單元形成問題是單元製造系統(CMS)設計最重要且最複雜的部分，可分成考量二元零件－機器關係矩陣的標準單元形成問題及考量多項生產資料之廣義單元形成問題二大類。雖然在標準單元形成問題部分，已經有很多方法被提出，但在廣義單元形成問題部分，鮮少有方法同時整合 CMS 設計的三大步驟：單元形成、單元佈置及單元內機器擺設並考量生產資料，含多途程、需求量、零件加工順序及機器可靠度。

有鑑於此，本論文首先結合相似係數法及萬用啟發式演算法，包括模擬退火法、仿水流優化演算法和禁忌搜尋演算法，發展出有效的混合式兩階段方法來有效求解標準單元形成問題。對於廣義單元形成問題，本文整合了 CMS 設計的三大步驟並考量生產資料，含多途程、需求量、零件加工順序及機器可靠度，提出一個兩階段的多目標數學規劃模式。接著，採用以廣義相似係數法及萬用啟發式演算法為基底的混合式演算法來有效的求解此多目標數學規劃模式。

不同於其他的方法，本論文所提出的單元形成方法不但整合了 CMS 設計的三大步驟且單元數可經由決策者輸入或根據最佳目標解自動產生。實驗分析和比較的結果展現本文所提的兩階段多目標數學規劃模式及三個演算法的有效性，並顯示這些演算法可以在很短的電腦執行時間內提供一個穩健的製造單元形成規劃。

關鍵字：群組技術，單元製造系統，單元形成問題，單元佈置設計，機器可靠度，模擬退火法，仿水流優化演算法，禁忌搜尋演算法

Models and Solution Methods for Cell Formation Problems

Student：Chin-Chih Chang　　Advisors：Dr. Shu-Hsing Chung

Dr. Tai-Hsi Wu

Department of Industrial Engineering and Management

National Chiao Tung University

ABSTRACT

Cell formation problem (CFP) is the first and most difficult aspect of constructing a preliminary cellular manufacturing system (CMS). The CFP can be classified into two main categories: the standard CFP represented by a binary machine-part incidence matrix and the generalized CFP with more factors and system constraints considerations. Although many studies have been done on standard CFP, generalized CFP had received less attention. Very little has been done to integrate the three basic steps (e.g., cell formation, cell layout, and intracellular machine layout) in the design of CMS.

Based on the above discussion, a two-stage hybrid algorithm merging a similarity coefficient method (SCM)-based clustering algorithm and meta-heuristics, including simulated annealing (SA), water flow-like algorithm (WFA) and tabu search (TS) is first presented to solve standard CFP quickly and effectively. In regard to the generalized CFP, a two-stage multi-objective mathematical programming model is first formulated to integrate cell formation, cell layout, and intracellular machine layout simultaneously with the considerations of alternative process routings, operation sequences, production volume, production times, and machine reliability. A two-stage hybrid approach integrating a generalized SCM-based clustering algorithm and SA/TS/WFA method is then proposed to solve this generalized CFP model quickly and effectively.

Unlike most existing methods, the proposed approach not only integrates the three basic steps in the design of CMS but also automatically calculates and determines the number of cells (NC) to achieve the best objective value. Illustrative examples, comparisons, and experimental analyses demonstrate the effectiveness of the proposed models and solution algorithms. The proposed approaches can be used to solve real-life CFP in factories by providing robust manufacturing cell formation in a short execution time.

Keywords: Group technology (GT), Cellular Manufacturing System (CMS), Cell formation problem (CFP), Cell Layout Design, Machine Reliability, Simulated Annealing (SA), Water Flow-like Algorithm (WFA), Tabu Search (TS)

誌謝

盼了五年!我終於可以畢業了!

博士學位在學術殿堂上是許多人夢寐以求的學歷，很感恩能進入國立交通大學工業工程與管理研究所博士班進修，並順利取得博士學位。

在這五年內發生了許多事，包括大女兒張卉、二女兒張鈺、小兒子泓瑋的相繼出生及父親的去世，此刻的我內心真是五味雜陳。

回首這段日子，非常感謝吳泰熙教授與鍾淑馨教授的指導。吳老師不論在學術研究上或是為人處世方面皆不厭其煩的指正，使我獲益良多，他是我在教學與研究工作上的啟蒙導師，也是我效法的榜樣，這樣的恩德學生畢生難忘。鍾老師像母親一樣的諄諄教導，對學生的寬厚與照顧，使我永銘在心。在論文口試期間，承蒙口試委員李榮貴教授、駱景堯教授與侯建良教授寶貴的建議與指正，使得本論文內容更臻充實完備，謹在此致上由衷的感謝。

能專心完成博士學位的最大精神支柱是我的親人，感謝他們在求學期間的體諒，尤其感謝愛妻敏珠對家庭的照顧，父母親對我的殷殷期盼及岳母的諸多幫助，大哥、大嫂、大姨子、弟、妹的精神鼓勵與支持，生管實驗室學弟妹於口試期間的協助，佛菩薩的慈悲安排，讓我能無後顧之憂的完成學位。最後，僅以此論文獻給我敬愛的母親及岳母，摯愛的妻子和等不及我畢業的父親以及所有關心我的人，祝您們平安順遂。

阿彌陀佛…

欽智 合十

于 台中 2010 歲末平安夜

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Motivations

In response to various and diversified customer demands, companies must adopt innovative manufacturing strategies and technologies to achieve an efficient and flexible manufacturing system. Group technology (GT) is one approach that meets the requirements of system flexibility and product variation. The cellular manufacturing system (CMS) is one of the applications of GT principles in manufacturing. Implementation of CMS resulted in significant benefits, such as reduced material handling costs, work-in-progress inventory, throughput and set-up times, simplified scheduling, and improved quality (Wemmerlov and Hyer, 1987). Hence, it has been widely discussed and applied by researchers and practitioners in the last three decades.

A cell formation problem (CFP) is the crucial element in designing a CMS (McAuley, 1972). However, the CFP in CMS is one of the NP-hard combinational problems (Kusiak, 1990). Hence, it is difficult to obtain optimal solutions in an acceptable length of time, especially for large-sized problems. Numerous models and solution approaches have been developed to address this problem since the 1970s. Different studies have focused on different aspects of CFP. Based on production data employed in CF models, the CFP is classified into two main categories: standard CFP represented by a binary machine-part incidence matrix and generalized CFP with more factors and system constraints considerations. Although many effective heuristics or algorithms have been done on standard CFP, very little has been devoted to integrate cell formation, cell layout, and intracellular machine layout, the three basic steps in the design of CMS, simultaneously with considerations of alternative process routings, operation sequences, production volume, and machine reliability on generalized CFP; thereby limiting the practical nature of their

approaches in a real CMS environment. Moreover, most methods in the literature assume that the NC is prescribed beforehand. However, determine the proper NC in the cell formation stage is very difficult for the layout designer because he does not have any knowledge at the beginning. Hence, it is important and more practical to integrate the abovementioned factors simultaneously in the design of CMS.

Due to their excellent performance in solving combinatorial optimization problems, meta-heuristic algorithms, such as simulated annealing (SA), water flow-like algorithm (WFA), and tabu search (TS), have been the most successful solution approach to provide global or near-global optimal solutions within a reasonable computation time. On the other hand, a number of similarity coefficient method (SCM)-based approaches have been proposed, and have been shown to produce good machine-part grouping and are more flexible in incorporating various production data into the machine-part clustering process.

Thus, the major research motivations for this thesis may be summarized as follows:

(1) CMS may provide great benefits.

(2) CFP is the first and most difficult aspect of constructing a preliminary CMS.

(3) CFP is one of the NP-hard combinational problems.

(4) There are few works that integrate cell formation, cell layout, and intracellular machine layout simultaneously with considerations of alternative process routings, operation sequences, production volume, production times, machine reliability, and different cellular layout type.

(5) It is difficult for a layout designer to determine the optimum cell number beforehand.

1.2 Research Objectives

Based on the research motivations, this thesis is dedicated to merging an SCM-based clustering algorithm and meta-heuristics to develop quick and effective hybrid algorithms to solve standard CFP and generalized CFP. Specific goals are as follows: (1) to merge an

SCM-based clustering algorithm and SA/TS/WFA method to present a fast and effective two-stage hybrid algorithm to solve standard CFP; (2) to formulate a two-stage multi-objective mathematical programming model to integrate cell formation, cell layout, and intracellular machine layout simultaneously with considerations of alternative process routings, operation sequences, production volume, machine reliability, and different cellular layout type; and (3) to integrate a generalized SCM-based clustering algorithm and SA/TS/WFA method to develop a fast and effective two-stage hybrid approach to resolve the formulated two-stage multi-objective mathematical programming model.

Unlike most previous studies where the NC to be formed is prescribed beforehand, the proposed methods do not demand a priori specification of the NC. Instead, it is automatically calculated and determined such that the best objective value may be achieved. Illustrative examples will be used to demonstrate the effectiveness of the proposed methods for standard CFP and generalized CFP. Hopefully, the proposed methods can be used to solve real CFP in factories by providing robust manufacturing cell formation in a short execution time.

1.3 Research Process

To achieve the abovementioned objectives, the research process (Figure 1.1) progresses as follows:

Step 1: Identifying research problems and objectives

Issues in CFP are identified through a discussion of research motivations and the purposes of this study.

Step 2: Literature review and discussion

The literature encompasses group technology and cellular manufacturing, solution methods for CFP, performance measures for CFP, and previous work on resolving CFP.

Step 3: Formulation of mathematical models

In this step, a mathematical model in terms of maximization of grouping efficacy is formulated to express standard CFP. Then, a two-stage multi-objective mathematical programming model for generalized CFP is formulated to integrate cell formation, cell layout, and intracellular machine layout simultaneously with considerations of alternative process routings, operation sequences, production volume, production times, machine reliability, and different cellular layout type.

Step 4: Development of proposed algorithms

In order to solve standard and generalized CFP mathematical models quickly and effectively, a two-stage hybrid CF algorithm (HCFA) merging an SCM-based clustering algorithm and SA/TS/WFA method is proposed to solve the standard CFP model. Afterwards, a two-stage hybrid generalized CF algorithm (HGCFA) merging a generalized SCM-based clustering algorithm and SA/TS/WFA method is proposed to solve the generalized CFP model.

Step 5: Validation of proposed algorithms

To demonstrate the power of our proposed algorithms for standard CFP, 35 test instances represented by a binary machine-part incidence matrix drawn from the literature are used to evaluate the computational characteristics of our proposed algorithms. On the other hand, 8 test instances, two drawn from the literature and the others prepared by adding self-creating data to test instances, are used to validate the quality of our proposed algorithm for generalized CFP.

Step 6: Summaries and Conclusions

The results are summarized and the conclusions are drawn in this step.

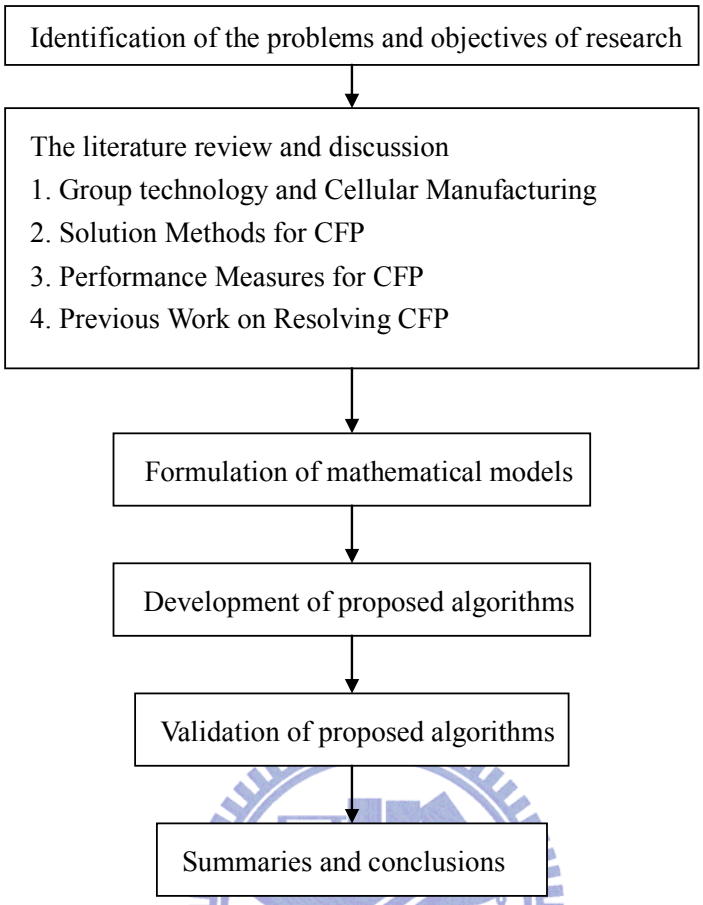


Figure 1.1 The flow chart of research

1.4 Organization

The remaining chapters are organized as follows. We present a literature review of CFP and the requisite solution techniques, including SA, TS, and WFA in Chapter 2. The mathematical models that express standard CFP and generalized CFP are formulated in Chapter 3. In Chapter 4, two hybrid meta-heuristic algorithms based on SCM-based clustering algorithm and SA/TS/WFA are proposed to solve the complex models. In Chapter 5, two numerical illustrations are given to demonstrate the effectiveness of the proposed methods for standard CFP and generalized CFP. Computational results for standard CFP and generalized CFP are shown in Chapter 6. Several strategies proposed in this thesis, together with some mechanisms, are further analyzed in Chapter 7. Conclusions of this thesis are finally drawn in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

This chapter is divided into fours sections. Section 2.1 introduces and defines GT and CM. Cell formation methods are reviewed in Section 2.2, while Section 2.3 describes the performance measures for CFP. Section 2.4 provides a review of previous work on resolving CFP.

2.1 Group Technology and Cellular Manufacturing

GT was originally introduced by Mitrovanov (1966) and was popularized in the west by Burbidge (1975). One application of GT is CM, a manufacturing philosophy in which similar parts are identified and grouped into part families, while machines are grouped into machine cells to take advantage of their similarities in manufacturing and design. Implementation of CM results in significant benefits, such as reduced material handling costs, work-in-progress inventory, throughput and set-up times, simplified scheduling, and improved quality (Wemmerlov and Hyer, 1987).

Although CM may provide great benefits, the CMS design is complex for real life problems. The design of a CMS consists of four stages as described below (Wemmerlov and Hyer, 1986).

1. CF – grouping parts with similar design features or processing requirements into part families and associated machines into machine cells.
2. Group layout – laying out machines within each cell (intra-cell layout) and cells with respect to one another (inter-cell layout).
3. Group scheduling – scheduling parts and part families for production.
4. Resource allocation – assigning tools, human and material resources.

Ideally, all of these stages should be addressed simultaneously in order to obtain the best results (Alfa *et al*., 1992). However, due to the complex nature of each stage and the

limitations of traditional approaches, this thesis will focus on stages 1 and 2. The solution methods for stages 1 and 2 will be discussed in the next section.

2.2 Solution Methods for CFP

The process of determining part families and machine groups is referred to as the CFP. It is known that the CFP in CMS is one of the NP-hard combinational problems (Ballakur and Steudel, 1987). Numerous solution approaches have been developed to address CFP since the 1970s, and these can be classified into five categories (Figure 2.1): (1) array-based methods, (2) similarity coefficient methods, (3) graph theoretic methods (4) mathematical programming methods, and (5) heuristic and meta-heuristic methods. Similarity coefficient methods and heuristic and meta-heuristic methods are related to this research and are discussed further.

(1) Array-based methods

The array-based methods attempt to allocate machines into groups and parts into associated families by appropriately rearranging the order of rows and columns to find a block diagonal form of the $a_{ki} = 1$ entries in the machine-part incidence matrix. The machine-part incidence matrix has 0 and 1 entries (a_{ki}). A '1' entry in row k and column i of the matrix indicates that part i has an operation on machine k, whereas a '0' entry indicates that it does not. Although cluster analysis methodologies are simple to implement, they have one main drawback: it usually takes into account only one objective i.e. the minimization of intercellular movements where only part operations and the machines involved are considered. Other product data (such as operational sequences and processing times) are not incorporated into the design process. Thus, solutions obtained may be valid for limited situations only.

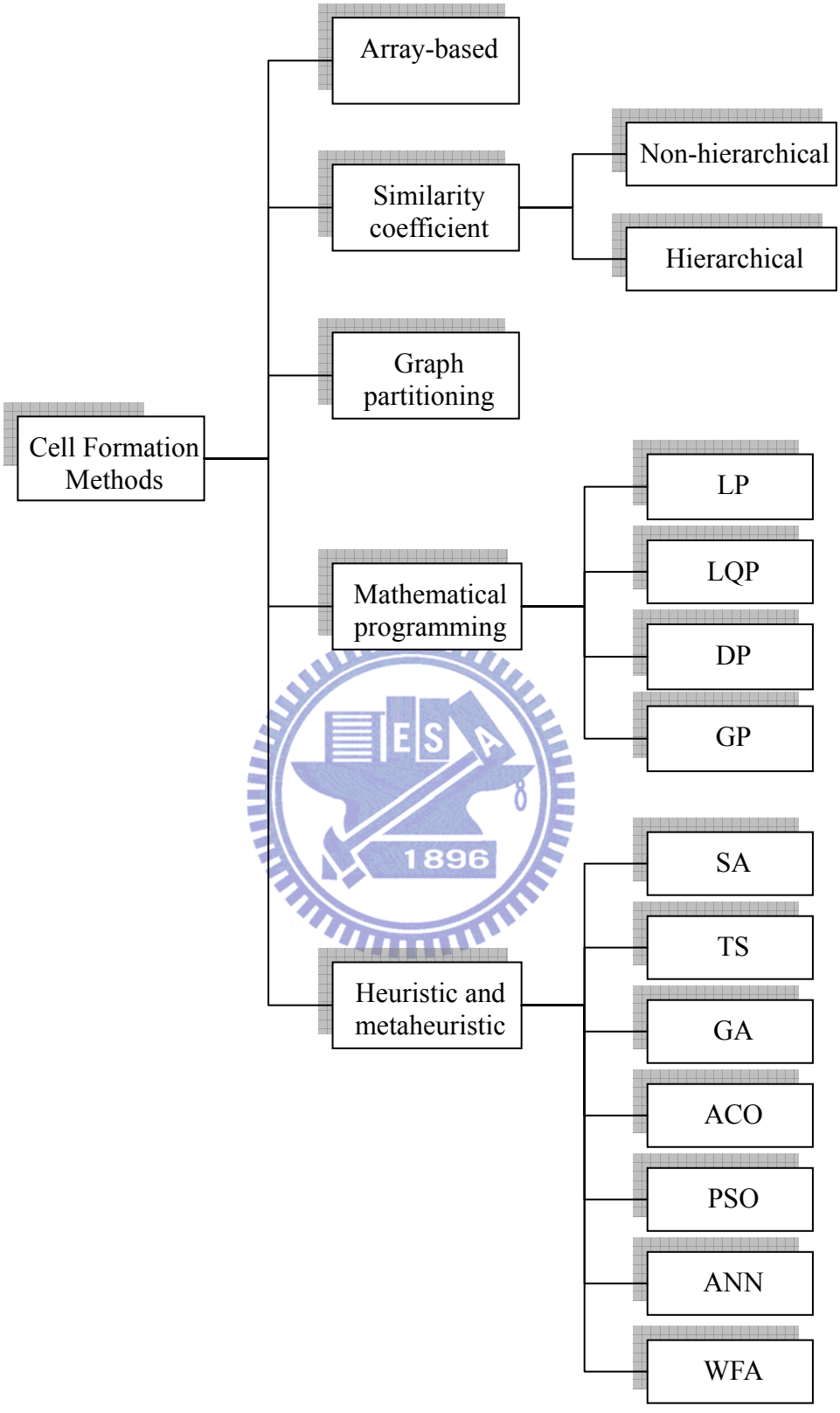


Figure 2.1 Classification of the CF solution method

(2) Graph theoretic methods

In graph partitioning approaches, the process of forming manufacturing cells starts with collecting problem data and then converting them into a weighted network diagram. Finally, the weighted network diagram is separated into several sub-groups of a machine cell. In the network diagram, nodes represent machines and arcs represent their relationships, defined as the value of total part flow between machines. In this method, the network diagram can clearly depict the flow of the machine, but other product data (such as machine capacity, processing times) are not easily incorporated. Therefore, graph partitioning approaches do not directly show the characteristics of multi-objective cell design.

(3) Mathematical programming methods

Mathematical programming methods can be presented in two parts: objective function and constraints. The establishment of objective function usually considers the factors related to manufacturing, e.g. minimizing inter-cell movement of parts, minimizing cell load unbalances, minimizing number of exceptional, and minimizing total manufacturing cost. Constraints express the content of production conditions, such as the limitation on number of machines, number of jobs, utilized time of tools, cells of machine allocation, controller's work time, and capability limitation. Mathematical programming methods can be further classified into four major groups based on the type of formulation: (1) linear programming (LP), (2) linear and quadratic integer programming (LQP), (3) dynamic programming (DP), and (4) goal programming (GP). The greatest advantage of this method is that different design objectives and constraints can be incorporated into a single formulated model. However, NP completeness of the problems makes it computationally intractable, especially for large-sized problems.

2.2.1 Similarity Coefficient Methods

SCMs are also referred to as cluster analysis-based methods in cell formation literature.

SCM are more flexible in incorporating various production data into the machine-part clustering process (Seifoddini and Tjahjana, 1999). The solution procedure of SCM usually follows a prescribed set of steps (Romesburg, 1984), the main ones being: (1) getting input data, (2) calculating the similarity coefficient, and (3) selecting a clustering algorithm to get machine cells. These steps are described next.

(1) Getting input data

Input data can be obtained from routing cards. These information are usually represented in a matrix called the machine-part incidence matrix, which is an $m \times p$ matrix with 0 or 1 entry, where m is the number of machines and p is the number of parts. Rows represent the machines and columns represent the parts. An element a_{ij} of the matrix is 1 if the jth part visits the ith machine for processing; otherwise, the value is 0.

(2) Calculating the similarity coefficient

The similarity coefficient is defined as a measure of similarity between machines, tools, design features, and so forth. Yin and Yasuda (2005) evaluated the performance of 20 well-known similarity coefficients, and found that the Jaccard similarity coefficient (Jaccard, 1908) is the most stable similarity coefficient. For this reason, we use the Jaccard similarity coefficient and the generalized similarity coefficient (Won and Kim, 1997) to calculate the similarity coefficient of standard CFP and generalized CFP, respectively. The generalized similarity coefficient is an extension of the Jaccard similarity coefficient (McAuley, 1972) and has been proposed for considering alternative process plans.

The Jaccard similarity coefficient is defined as the ratio of the number of parts visiting both machines and the number of parts visiting one of the two machines:

$$S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij} + c_{ij}} \tag{2.1}$$

where a_{ij} represents the number of parts processed by both machines i and j; while b_{ij} is the number of parts processed by machine i but not by machine j, and c_{ij} is the number of

parts processed by machine *j* but not by machine *i*.

On the other hand, the generalized similarity coefficient is formulated as:

$$S_{ij} = \frac{N_{ij}}{N_i + N_j - N_{ij}} \tag{2.2}$$

where

S_{ij} = similarity coefficient between machines *i* and *j*

$$N_i = \sum_{k=1}^{p} a_i^k,\ N_j = \sum_{k=1}^{p} a_j^k,\ N_{ij} = \sum_{k=1}^{p} a_{ij}^k$$

p = number of parts

$$a_i^k = \begin{cases} 1 & \text{if } i \in \text{ some routing of part } k \\ 0 & \text{otherwise} \end{cases}$$

$$a_j^k = \begin{cases} 1 & \text{if } j \in \text{ some routing of part } k \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij}^k = \begin{cases} 1 & \text{if } i, j \in \text{ some routing of part } k \text{ synchronously} \\ 0 & \text{otherwise} \end{cases}$$

(3) Selecting a clustering algorithm to get machine cells

When the values of the similarity coefficients have been calculated, a clustering algorithm can be selected to get machine cells. Conventional clustering algorithms are divided into two major classes: hierarchical and non-hierarchical. Hierarchical clustering for CF comprises two stages. Initially, some form of similarity or dissimilarity between machines or parts is employed in order to create machine cells or part families. Later, machines or parts are separated into a few broad cells, each of which is further divided into smaller groups and each of these further partitioned and so on until terminal groups that cannot be subdivided are generated. Essentially, hierarchical techniques can be classified into two: (a) divisive methods where the process starts with all the data (machines or parts) in a single group and a series of partitions is created until each machine (part) is in a singleton cluster and, (b) agglomerative methods where the process starts with singleton clusters and proceeds to merge them into larger partitions until a partition containing the

whole set is obtained.

Non-hierarchical clustering methods are iterative methods that also employ a measure of similarity or dissimilarity for grouping parts or machines. They begin with either an initial partition of the data set or the choice of a few seed points. In either case, the number of clusters has to be decided on beforehand.

Among the abovementioned approaches, the SCMs are more flexible in incorporating various production data into the machine-part clustering process. On the other hand, the heuristic and meta-heuristic methods are especially useful in providing near-optimum solutions within a reasonable computation time when a CFP cannot be solved using traditional methods, and thus constitute the state-of-the-art algorithm for solving CFP.

2.2.2 Heuristic and meta-heuristic methods

Heuristic and meta-heuristic methods are random heuristic search algorithms applicable to a wide variety of combinatorial optimization problems. They include SA (Su and Hsu 1998, Sofianopoulou 1999, Arkat *et al*. 2007), TS (Sun *et al*. 1995, Adenso-Diaz *et al*. 2001, Wu *et al*. 2004, Lei and Wu 2005), genetic algorithms (GA; Lee *et al*. 1997, Onwubolu and Mutingi 2001, Boulif and Atif 2006, Chan *et al*. 2008), ant colony optimization (ACO; Kao and Fu 2006), particle swarm optimization (PSO; Andres and Lozano 2006), artificial neural network (ANN; Park and Suresh 2003, Yang and Yang 2008), and WFA (Yang and Wang 2007). Although heuristic and meta-heuristic methods are not guaranteed to provide optimal solutions (they usually give sub-optimal results), they are very useful in producing acceptable solutions within a reasonable time. In fact, optimal results can only be obtained under very restricted conditions; this makes heuristic and meta-heuristic methods more practical in real-life applications. SA, TS, and WFA are relevant to this research and are discussed further.

2.2.2.1 Simulated Annealing

SA algorithm was originally proposed by Metropolis *et al*. (1953) to simulate the annealing process. Based on this pioneering work, Kirkpatrick *et al*. (1983) first introduced the general optimization algorithm of SA to solve hard combinatorial optimization problems through controlled randomization. Lundy and Mees (1986) proved that the SA algorithm converges to the global optimum with a probability close to one under certain assumptions. SA poses several advantages over other sophisticated combinatorial optimization approaches, e.g. relatively easy and quick implementation, flexibility, and transparency. Due to its ease of use and its ability to provide a good solution for real-world problems, SA is one of the most powerful and popular heuristics to solve many optimization problems. For instance, adequate results have been attained when applying SA on various combinatorial problems (Kirkpatrick *et al*. 1983, Bonomi and Lutton 1984, Aarts and Van Laarhoven 1985, Selim and Alsultan 1991, Mckendall *et al*. 2006, Yu *et al*. 2010).

The pseudo-code of the general procedure for implementing the SA algorithm in maximization problems is presented in Figure 2.2. The algorithm starts with a high temperature. After generating an initial solution (S^0), it attempts to move from the current solution (S) to one of its neighborhood solutions (S^N).Changes in objective function values ($\Delta = S^N - S$) are computed. The new solution is accepted if it results in better objective value (i.e. $\Delta > 0$). However, if the new solution yields worse value, it can still be accepted according to the probability function $p = e^{\Delta/T}$, where T is the current temperature. This check is performed by first selecting a random number (r) from (0, 1). If the value is less than or equal to the probability value (p), the new configuration is accepted; otherwise, it is rejected. By accepting worse solutions, SA can avoid being trapped on local optima. SA repeats this process L times at each temperature to reach the thermal equilibrium, where L is a control parameter usually called the Markov chain length or Epoch length. The parameter

T is gradually decreased by a cooling function as SA proceeds until the stopping condition is met.

The general scheme of SA can be stated as follows:

Step 1. Choose an initial temperature T.

Step 2. Generate a random candidate S.

Step 3. If a stopping criterion is satisfied, then stop; otherwise repeat the following steps:

Step 3.1. If "thermal equilibrium is reached," then exit this loop.

Step 3.2. Let S^N be a randomly selected neighbor of S.

Step 3.3. Generate a uniform random number r from [0, 1].

Step 3.4. Compute the changes in the objective function values: $\Delta = (S^N - S)$.

Step 3.5. If $e^{\Delta/T} > r$, then $S = S^N$.

Step 4. Let T be a new (lower) temperature value, then go to Step 3.

The annealing schedule mainly consists of (1) the initial temperature, (2) a cooling function, (3) the number of iterations to be performed at each temperature, and (4) a stopping criterion to terminate the algorithm. Performance analysis of SA had revealed several characteristics (Lin *et al*., 1993):

(1) There is a tradeoff between the quality of the final solution obtained and the execution time required. Furthermore, the execution time is sensitive to the decrement ratio of the temperature.

(2) If the temperature drops too sharply, is the algorithm becomes easily trapped in local minima.

(3) Detecting the equilibrium of the system at each temperature level is not an easy task.

(4) The total number of iterations of SA is affected by the initial temperature.

(5) If the numbers of iterations at low temperature regions are not large enough, there are still some probability of departing from good solutions.

Hence, several decisions have to be made in order to implement the conceptual

algorithm described above. These include the following:

(1) Choice of an initial temperature and the corresponding temperature decrement strategy

At a high temperature, almost all unimproved trial solutions are accepted. However, at a lower temperature, fewer unimproved trial solutions can be accepted. If the cooling speed is too fast or the initial temperature (T_0) is not high enough, this mechanism will fail to escape local minima. T_0 should be high enough that in the first iteration of the algorithm, the probability of accepting worst solutions is, at least, 80% (Kirkpatrick *et al*., 1983). The most commonly used temperature decrement function is geometric: $T = a \times T$, where $a < 1$ and constant. Typically, $0.7 \le a \le 0.95$.

(2) Choice of a criterion for detecting equilibrium

For each value of the current temperature T, the inner loop (steps 3.1 to 3.5 in the algorithm presented above) should be repeated L times in order for the system to reach "thermal equilibrium." If the search cannot reach the equilibrium state at each temperature, obtaining a globally optimum solution becomes difficult. A good criterion for thermal equilibrium can save computational effort without losing the ability of escaping from a local minimum.

(3) Choice of an adequate stopping criterion

The stopping criterion is used to stop the algorithm when there is sufficient evidence that the global optimum has been detected or that the "cost" connected with the search for a better estimate of the global optimum would be too high. The stop can also occur when some kind of "resource" has been exhausted, e.g. computer time, the total number of solutions generated, and when the desired energy level is attained (freezing temperature). The stopping criterion is always the crucial and most difficult part of the algorithm, and has great influence on overall performance.

Obviously, each of these control parameters is chosen according to the specific problem at hand. In addition to the control parameters, two other important issues that need

to be defined when adopting this general algorithm to a specific problem are the procedures to generate both the initial solution and the neighboring solutions. The details of the proposed implementation of the SA to the CFP are presented in Section 4.1.2.

```
SA_Algorithm ( )
{
   Generate an initial solution S^0.
   Let the current solution S equal to S^0.
   Let the current best solution S* equal to S^0.
   Let the current temperature T equal to the initial temperature T_0.
   WHILE(stop criterion is false)      // outer loop
   {
       Let repetition counter n = 1.
       WHILE(n < Markov chain length L)      // inner loop
       {
            Generate a random solution S^N in the neighborhood of S.
            Compute Δ = f(S^N) – f(S).
            IF ( Δ > 0  or  e^(-Δ/T) > r ∈ U(0,1) )
                Let S ← S^N.
            IF ( f(S^N) > f(S*) )
                Let S* ← S^N.
            n= n + 1.
       }
       Reduce the temperature T.
   }
}
```

Figure 2.2 Pseudo-code for general simulated annealing algorithm (Kirkpatrick *et al*., 1983)

2.2.2.2 Tabu Search

TS is a meta-heuristic approach designed to find optimal or near-optimal solutions to combinatorial optimization problems. This method has been suggested primarily by Glover *et al*. (1985) and further refined and developed by Glover (1986, 1989, and 1990). The pseudo-code for the general procedure for implementing the TS is presented in Figure 2.3.

```
TS_Algorithm ( )
{
   Generate an initial solution S^0 .
   Let the current solution  S  equal to S^0 .
   Let the current best solution S^*  equal to S^0 .
   WHILE(stop criterion is false)
   {
         Generate a best solution  S^N (S^N ∉ N^T or ∈ N^A)  in the neighborhood of  S .
         Update tabu list N^T .
         IF ( f(S^N) < f(S^*) )
            Let  S^* ← S^N .
         Let  S ← S^N .
    }
 }
```

Figure 2.3 Pseudo-code for general TS algorithm

The algorithm begins from a randomly selected or a known initial solution (S^0). From this solution, a set of neighborhood solutions of the current solution (*S)* is generated using the predefined movement strategies. The objective function is evaluated for each neighborhood of *S* and the best neighbor solution (S^N) replaces the *S* even though the best neighbor solution may be worse than the current one. In this way, the algorithm can escape from the local minima (or maxima) of the objective function. However, the algorithm may recycle. To avoid this situation, certain attributes of the last replaced solution are stored in a list, which is called a tabu list (N^T). The neighbors of *S* that satisfy conditions given by the tabu list are systematically eliminated unless they meet an aspiration criterion (N^A), so that at each iteration, the algorithm is forced to select a point not recently selected. TS has been successfully used to solve many optimization problems in a wide variety of areas, including CFP, graph coloring, traveling salesman problem, path assignment, flow shop sequencing, job shop sequencing, and dealing with learning in neural networks (Glover and Languna, 1993). More detailed discussions of the foundations of TS methodology can be found in Glover (1989, 1990) and Glover and Laguna (1997).

In general, the main components of TS are the initial solution, neighborhood and

moves, tabu list, aspiration criterion, stopping criterion, intensification, and diversification. The details of these are described next.

(1) Initial solution

The quality of the initial solution is crucial to the efficiency of TS. It is known that a good initial solution will improve the efficiency of TS. Generally, the initial solution is produced by some rules or problem-specific heuristics instead of random generation.

(2) Neighborhood and moves

The neighborhood of a solution is the set of all formations that can be arrived at by a move. Since neighborhood depends on the current solution, new neighborhood is generated every time the current solution changes. Generally, neighboring solutions can be generated by insertion method, pair-wise interchange, and adjacent interchange method. Different methods are employed according to the problem. From all neighboring solutions, the best one is chosen to move forward. However, this best solution may sometimes be in the tabu list and does not satisfy aspiration criterion. When this happens, the second best solution is chosen to move forward if it is not in the tabu list; otherwise, the third best solution is considered and so on.

(3) Tabu list

In order to prevent scheme cycling and returning to the same solutions, it is necessary to introduce a condition that prevents this from happening. This is usually carried out by not allowing reversal of moves for a certain number of iterations equal to the tabu length. These non-admissible moves within the short interval comprise the class membership of a tabu list. The size of the tabu list must be large enough to prevent cycling, but small enough to not forbid too many moves. A minimum of 7 and a maximum of 11 has been suggested for tabu length (Glover and Laguna, 1993).

(4) Aspiration criterion

The tabu restriction may be overridden if the move will result in a solution that is

better than the best solution found thus far. Thus, if a tabu move satisfies the associated aspiration criterion, it is considered admissible.

(5) Stopping criterion

The most accepted stopping criterion relies on the search being terminated if the objective function value has not improved within a certain number of iterations that is usually specified at the start of the run. Another criterion relies on the search being terminated if a maximum number of iterations has been reached to avoid an extremely long run. The problem with the latter criterion is that it is difficult to determine the maximum number of iterations because the value may either lead to premature termination or expensive termination.

(6) Intensification

The mechanism for intensification enhances the search to focus on examining elite solutions in a neighborhood. It tends to move the search to a neighboring position in the search space, and so could be considered a local search.

(7) Diversification

The mechanism for diversification allows a large jump to be made in the solution space. This ensures that large areas of the space are searched and solutions do not get stuck in local minima. This mechanism is also referred to as the restarting procedure. For each diversification process, a different initial cell formation is randomly generated. This way, the search is able to explore a large solution space, thereby enhancing the possibility of finding the optimum solution in a very short time.

TS has the following characteristics:

(1) Utilizes a flexible memory structure, which is more efficient than strict memory structure (ex: branch-and-bound) or no memory (ex: simulated annealing).

(2) Allows searching toward worse solutions in order to get rid of local optimum.

(3) Records explored solutions in tabu list in order to avoid redundant searching time.

(4) Updates the tabu list in every step to reduce the probability of redundant searching and to improve searching efficiency.

(5) Uses an aspiration criterion to relax tabu restriction and keep the searching process going.

(6) Sets upper bounds of iteration numbers passed or time elapsed to terminate the searching process.

2.2.2.3 Water Flow-like Algorithm

The design of the WFA (Yang and Wang, 2007) was inspired by the natural behavior of water flowing from higher to lower levels. On the earth's surface, a flow will split into multiple sub-flows when rugged terrains are traversed. Sub-flows, however, will merge when they arrive at the same location. Governed by gravity and driven by fluid momentum, flows can run to higher levels or run over bumps to navigate various terrains. Water flow will cease and stagnate at the locally or globally lowest depression; when the momentum left cannot expel the water out of the depression, it will stagnate at its current location. No movement is allowed until other flows merge with it or until the water evaporates into the atmosphere. When the evaporated water accumulates to some extent, it will return to the ground as several new downpour flows, such that rainfall occurs occasionally. As the solution space of a problem can be mapped to the geographical terrain, and the objective value is mapped to the altitude, each flow can then be regarded as a solution agent. Water moving to a lower position can be considered as a solution searching for the optima. Thus, the solution search process has been modeled as water flow.

Yang and Wang (2007) adopted several natural behaviors of water flow in presenting the WFA (Dougherty and Marryott, 1991). Their design ideas are summarized as follows:

(1) Driven by gravity and governed by the energy conservation law, water will constantly flow to lower altitudes. Conversely, the solution search will recursively move from inferior to superior solutions.

(2) Fluid momentum drives water forward through rough terrains. A flow will split into sub-flows when it encounters rugged terrain and when its momentum exceeds a base amount for splitting. WFA simulates this behavior as an agent forking operation; that is, more than two agents are derived from a single agent. A flow with larger momentum will generate more streams of sub-flows than one with less momentum. A flow with limited momentum will yield to the landform and maintain a single flow. Therefore, the fluid momentum of a flow is recalculated to determine the number of sub-flows that can be forked after each move.

(3) Water flows to lower altitudes and occasionally swells to higher altitudes as long as the kinetic energy is larger than the required potential energy. To avoid being trapped within a local minimum, WFA allows the water to flow to a worse location to broaden the exploration area, provided it has enough kinetic energy.

(4) A number of flows merge into a single flow when they meet at the same location. WFA reduces the number of solution agents when multiple agents result in the same objective value to avoid redundant searches.

(5) Water flows are subject to water evaporation in the atmosphere. The evaporated water will return to the ground in the form of rainfall. In WFA, a part of the water flow is manually removed to mimic water evaporation. After evaporation, a precipitation operation is implemented in WFA to simulate natural rainfall and explore a wider solution area.

The pseudo-code for the general procedure for implementing the WFA is shown in Figure 2.4.

```
WFA_Algorithm ( )
{
   Generate an initial solution.
   WHILE(stop criterion is false)
   {
          Flow splitting and moving.
          Flow merging.
          Water evaporation.
          IF (rainfall required)
          {
              Precipitation.
              Flow Merging.
          }
          IF (new best solution found)
               Update best solution record.
    }
 }
```

Figure 2.4 Pseudo-code for WFA algorithm

The WFA algorithm consists of four primary operations: (1) flow splitting and moving, (2) flow merging, (3) water evaporation, and (4) precipitation. Before proceeding to the descriptions of these four operations, we introduce some notations.

N_{max} : Iteration limit

W_0 : Initial mass of original flow

W_i : Mass of flow i

V_0 : Initial velocity of original flow

V_i : Velocity of flow i

T_m : Base momentum

$\bar{n}$: Upper limit on number of subflows split from a flow

n_i : Number of subflows forked from flow i

N : Total number of water flows in current iteration

μ_{ik} : Velocity of subflow k split from flow i

δ_{ik} : Attitude drop from flow i to subflow k; equivalently, changes in objective value from solution i to its neighborhood solution k

G : Gravitational acceleration

T : A prescribed number of iteration a flow will be removed by evaporation

1. Flow splitting and moving operation

It is assumed that there is only one water flow in the beginning of the WFA, and that its location is randomly generated. Driven by fluid momentum and potential energy, the flow starts to move to new locations to explore the solution space for new and better solutions. Yang and Wang (2007) used constant-step movement to the best neighborhood solution when solving the object grouping problem. However, various flow-moving strategies can be designed and applied depending on the characteristics of different optimization problems.

In the WFA, flow splitting results from capable momentum, and a flow with higher momentum generates more sub-flows than that with a lower one. The locations of the split sub-flows are derived from the neighboring locations of the original flow. When a flow does not split, it goes on as a single stream to the best feasible neighboring location. Allowing N to be the number of water flows in the current iteration, the number of sub-flows n_i forked from flow i (i = 1, 2, …, N) is determined by its momentum, W_iV_i. A flow with zero momentum stays in its current location and is considered a stagnant solution. A flow can split into sub-flows only when its momentum exceeds a predefined base momentum T_m. The number of sub-flows is determined by dividing its momentum by the base momentum T_m. If the momentum of a flow is between 0 and T_m, it is treated as a single stream moving to a new location without splitting. As WFA proceeds, it is possible that the number of sub-flows grows exponentially and exhausts the computational resource. Yang and Wang (2007) suggests imposing an upper limit $\bar{n}$ on the number of sub-flows forked from a flow at each iteration. The number of sub-flows split from a flow can thus be obtained through:

$$n_i = \min\left\{\max\left\{1,\ \text{int}\left(\frac{W_i V_i}{T_m}\right)\right\},\ \bar{n}\right\} \tag{2.8}$$

When the flow is split into sub-flows, its original mass has to be accordingly distributed to sub-flows based on the rule designed. Yang and Wang (2007) distributed mass based on the ranks of the sub-flows, as shown in Eq. (2.9).

$$w_{ik} = \left(\frac{n_i + 1 - k}{\sum_{r=1}^{n_i} r}\right) W_i,\ k = 1, 2, \ldots, n_i \tag{2.9}$$

For instance, if W_i =5 and n_i =3, then

$$w_{i1} = \left(\frac{3}{1+2+3}\right)5,\ w_{i2} = \left(\frac{2}{1+2+3}\right)5,\ w_{i1} = \left(\frac{1}{1+2+3}\right)5.$$

The velocity of each sub-flow is computed from the equation of energy conservation. μ_{ik}, the velocity of sub-flow k split from flow i, is:

$$\mu_{ik} = \begin{cases} \sqrt{V_i^2 + 2g\delta_{ik}}, & \text{if } V_i^2 + 2g\delta_{ik} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

where g is the gravitational acceleration, and δ_{ik} is the altitude drop from flow i to its sub-flow k; that is, the improvement of objective value moving from current solution i to its neighborhood solution k. When $V_i^2 + 2g\delta_{ik} < 0$, the momentum delivered to sub-flow k has been used up, implying that this sub-flow will stagnate in its current location (e.g. the solution is trapped in local optima) without splitting and movement. Such stagnant flow will gradually evaporate into the atmosphere, returning to the ground by precipitation later on.

At the end of the splitting and moving operation, the original flow becomes discarded because sub-flows have been generated. Information regarding the current number of sub-flows and solutions sets will then be recorded.

2. Flow-merging operation

When more than two flows move to the same location, they will merge into one flow

with a bigger mass and momentum. Whether a flow shares the same location with others in the WFA is thus systematically examined. If a flow does share the same location, the latter flow is then merged into the former one. Assuming that flows *i* and *j* share the same location, then flow *j* will be deleted and the mass and velocity of flow *i* will be updated as follows:

$$W_i = W_i + W_j \tag{2.11}$$

$$V_i = \frac{W_i V_i + W_j V_j}{W_i + W_j} \tag{2.12}$$

Using the flow-merging operation, the WFA reduces the number of solution agents when multiple agents result in the same objective value in order to avoid redundant searches.

3. Water evaporation operation

It is natural for water to evaporate and return to the ground through precipitation after possible movement from its original location. Water evaporation and precipitation coincide with the "escaping from local optima" mechanism that many heuristic algorithms nowadays use to avoid being trapped and to explore more solution spaces.

Each flow in the WFA is subject to water evaporation, where part of the water evaporates into the atmosphere. It is determined that a flow will be completely removed after a prescribed number of iterations *t*; that is, the masses of all flows are decreased by the ratio of *1/t*, as shown in Eq. (2.13), every time evaporation occurs.

$$W_i = \left(1 - \frac{1}{t}\right) W_i, \ i = 1, 2, ..., N \tag{2.13}$$

4. Precipitation operation

When water vapor accumulates to a certain volume, it will return to the ground in some form such as rain. In the original WFA, two types of precipitation are performed to simulate the natural cycle of water: enforced and regular precipitation.

Enforced precipitation is applied when all flows are grounded with zero velocities. Under this circumstance, all flows are forced to evaporate into the atmosphere and then returned to the ground without changing the number of current flows. However, the locations of these returned flows are deviated stochastically from the original ones. Mass of W_0 is proportionally distributed to flows based on their original mass with the same initial velocity. Consequently, the mass assigned to flow i, W'_i, can be determined using Eq. (2.14).

$$W'_i = \left(\frac{W_i}{\sum_{k=1}^{N} W_k} \right) W_0 \tag{2.14}$$

Regular precipitation is applied periodically in balance with water evaporation. The regular precipitation operation is performed every t (same t value as in evaporation) iterations to pour down the evaporated water. Note that the cumulative mass of the evaporated water is $W_0 - \sum_{k=1}^{N} W_k$. Thus, instead of using Eq. (2.14), the mass assigned to flow i, W'_i, is determined using Eq. (2.15) when applying regular precipitation. The newly poured flow joins the current solution set, thus increasing the number of current solutions. In addition, both enforced and regular precipitation might generate several new flows in the same locations. A flow merging operation will be executed to eliminate possible redundant flows.

$$W'_i = \frac{W_i}{\sum_{k=1}^{N} W_k} \left(W_0 - \sum_{k=1}^{N} W_k \right) \tag{2.15}$$

2.3 Performance Measures for CFP

There is a need to develop performance measures or criteria in order to compare the quality of solutions obtained by different methods on an absolute scale. A limited number of performance measures have been proposed. Some commonly known grouping efficiency measures for 0-1 machine-part incidence matrix data are illustrated in Table 2.1. Among

them, two measures frequently used are the grouping efficiency (Chandrashekharan and Rajagopalan, 1987) and the grouping efficacy (Kumar and Chandrasekharan, 1990) because of their ease of implementation.

Although grouping efficiency has been used widely, critics argue that it has weak discriminating power (i.e., the ability to distinguish good quality grouping from bad). For example, a bad solution with large number of exceptional elements will give a value around 0.75. To overcome the low discriminating power of grouping efficiency between well-structured and ill-structured incidence matrices, Kumar and Chandrasekharan (1990) proposed another measure that they called grouping efficacy. Unlike grouping efficiency, grouping efficacy is not affected by the size of the matrix. Today, grouping efficacy is one of the most widely used measures applied to the CFP when a binary machine-part incidence matrix is used. Grouping efficacy can be defined as:

$$\Gamma = \frac{e - e_0}{e + e_v}, \tag{2.3}$$

where e is the total number of 1s in the matrix; e_0 is the total number of exceptional elements; and e_v is the total number of voids. Those 1's outside the diagonal blocks are called ''exceptional elements'', while those 0's inside the diagonal blocks are called ''voids.'' Grouping efficacy ranges from 0 to 1, with 1 being the perfect grouping.

We chose grouping efficacy as the measure of performance for the standard CFP in this thesis for several reasons:

(1) In the literature, it has been considered the standard measure to report the quality of the grouping solutions.

(2) It has a high capability to differentiate between well-structured and ill-structured matrices (e.g. high discriminating power).

(3) It is considered a better measure than grouping efficiency.

(4) It is able to incorporate both within-cell machine utilization and inter-cell movement.

(5) It generates block diagonal matrices that are attractive in practice.

(6) It does not require a weighting factor.

Table 2.1 Commonly known measures for 0-1 machine-part incidence matrix data

Measure	Name	Definition	Reference
1	Grouping efficiency(η)	$\frac{qe_1}{(e_1+e_v)}+\frac{(1-q)(o-e_v)}{(o-e_v)+(e-e_1)}$	Chandrasekharan and Rajagopalan (1986a)
2	Grouping efficacy (Γ)	$\frac{(e-e_0)}{(e+e_v)}$	Kumar and Chandrasekharan (1990)
3	Grouping capability index (GCI)	$1-\frac{e_0}{e}$	Hsu (1990)
4	Grouping measure(η_g)	$\frac{e_1}{(e_1+e_v)}-\frac{e_0}{e}$	Miltenburg and Zhang (1991)
5	Weighted grouping efficacy (γ)	$\frac{q(e-e_0)}{q(e+e_v-e_0)+(1-q)e_0}$	Ng (1993)
6	Grouping index(τ_3)	$\frac{B-qe_v-(1-q)(e_0-A)}{B+qe_v+(1-q)(e_0-A)}$, $A=\begin{cases}0, & e_0\le B\\ e_0-B, & e_0>B\end{cases}$	Nair and Narendran (1996)
7	Alternative routing grouping efficiency(η_{ARG})	$\left(\frac{e-e_0}{e+e_0}\right)\left(\frac{o-e_v}{o+e_v}\right)$	Sarker and Li (1998)
8	Double weight grouping efficacy(η_Q)	$\left(\frac{qe_1+(1-q)e_v}{e_1+e_v}\right)\left(\frac{qe_1+(1-q)e_0}{e_1+e_0}\right)$	Sarker (2001)

e: total number of ones in the machine-part incidence matrix; *o*: total number of zeros in the machine-part incidence matrix; e_0: total number of exceptional elements; e_v: total number of voids; e_1: total number of ones within the diagonal blocks; q: weighting factor.

Although grouping efficiency and grouping efficacy have been used widely, they do not consider production factors, such as process sequence of operations, production volumes processing times of operations, and were designed for 0–1 matrices only. Hence, Harhalakis *et al*. (1990) proposed another measure called the group technology efficiency (GTE) that takes into account the sequence of operation, which can be defined as:

$$GTE=1-\frac{U}{I}, \tag{2.4}$$

where I is the maximum number of inter-cell travels possible and U is the number of inter-cell travels actually required by the system.

Seifoddini and Djassimi (1995) developed a new grouping measure called Quality Index (QI) that takes into account the sequence of operation, production volume, and processing times of operation. This can be defined as:

$$\text{Quality Index } (QI) = 1 - \frac{ICW}{PW}, \tag{2.5}$$

where ICW is the intercellular workload and PW is the total plant's workload.

Nair and Narendran (1998) observed that the GTE is inadequate because it is poor in pattern recognition. Hence, they proposed another measure called bond efficiency that takes into account inter-cell moves within cells and *compactness*, which can be defined as:

$$\text{Bond efficiency}(BE) = q \times GTE + (1-q) \times Compactness, \tag{2.6}$$

where $q(0 \le q \le 1)$ is a weighting factor; and *Compactness* is the ratio of the number of operations within it to the maximum number of operations possible in it, and is given by:

$$Compactness = \frac{\sum_{k=1}^{NC} TOTOP_k}{\sum_{k=1}^{NC} (TOTOP_k + NOP_k)}, \tag{2.7}$$

where NC is the maximum number of machine-cells; $TOTOP_k$ is the total number of operations in the kth cell；and NOP_k is the total number of non-operations (voids) in the kth cell.

Although the abovementioned performance measures have taken into account the production sequence, production volume, and processing times of operation, many realistic factors such as alternative process routings, cellular layout, and machine reliability are still not considered simultaneously. If incorporated, these factors can enhance the quality of solutions. Hence, a performance measure for cell formation, cell layout, and intracellular machine layout with considerations of alternative process routings, operation sequences, production volume, machine reliability, and different cellular layout type is developed in Section 3.4.

2.4 Previous Work on Resolving CFP

Numerous models and solution approaches have been developed to deal with CFP since the 1970s. Some focus on developing effective heuristics or algorithms for solving standard CFP in which machine cells and part families are obtained sequentially or simultaneously. For instance, McAuley (1972) and Carrie (1973) developed the first algorithms using SCM on CFP. King and Nakornchai (1982) developed the earliest array-based methods to solve CFP. Cheng *et al*. (1998) formulated the CFP as a traveling salesman problem and solved the model using GA. Gonçalves and Resende (2004) presented an evolutionary algorithm (EA) for obtaining machine cells and product families. Yang and Yang (2008) proposed a modified ART1 neural learning algorithm for CFP. Unler and Gungor (2009) effectively applied the K-harmonic means clustering technique to form machine cells and part families simultaneously. Meanwhile, Tariq *et al*. (2009) combined a local search heuristic with GA and developed a hybrid GA for machine-part grouping. Mahdavi *et al*. (2009) designed an efficient algorithm based on GA to solve the CFP.

On the other hand, some focus on considering more factors and system constraints for forming machine cells and part families. For instance, Gupta *et al*. (1996) presented a bi-criteria model simultaneously considering the minimization of the weighted sum of inter-cell and intra-cell moves and the minimization of the total cell load variation. Lee *et al.* (1997) developed a GA to deal with the CFP considering production volumes, alternate routings, and process sequences. Su and Hsu (1998) introduced a parallel SA to minimize (1) the total cost of machine investment, as well as inter-cell and intra-cell transportation cost; (2) intra-cell machine loading unbalance; and (3) inter-cell machine loading unbalance. A similar study was made by Lei and Wu (2005). They presented a Pareto-optimality-based multi-objective TS algorithm for machine-part grouping problems with multiple objectives. They were able to minimize total cost, which includes intra- and inter-cell transportation cost and machine investment cost, thus minimizing intra-cell loading unbalance and

inter-cell loading unbalance. Sofianopoulou (1999) developed an SA to address the CFP with alternate routings and process sequences considerations. Akturk and Turkcan (2000) proposed a local search algorithm to solve cell formation and intra-cell layout problem simultaneously. Bazargan-Lari *et al*. (2000) presents the application of an integrated approach to the three phases of CM design to a white-goods manufacturing company in Australia. Chiang and Lee (2004) proposed a GA-based algorithm augmented with the optimal partition approach to deal with both cell formation and inter-cell layout simultaneously. Hu and Yasuda (2006) presented a GA to minimize the total material handling costs for CFP with alternative processing routes. Boulif and Atif (2006) developed a new branch and bound enhanced GA to the CFP with considerations of process sequences, maximum NC, maximum cell size, and machine cohabitation and non-cohabitation. Chan *et al.* (2006) proposed a two-stage method that solved CFP and cell layout problems simultaneously by GA. Wu *et al*. (2006) developed a hierarchical GA to concurrently solve cell formation and inter-cell and intra-cell layouts in CMS design. Based on a new concept of similarity coefficients and the use of SA, Arkat *et al.* (2007) proposed an effective methodology to solve the CFP with alternative routings and production volume considerations. Wu *et al*. (2007a, b) developed a hierarchical GA to concurrently integrate cell formation and intracellular machine layout decisions in CMS design. Jabal Ameli and Arkat (2008) presented a mathematical approach to cell formation with alternative process routings and machine reliability consideration. Meanwhile, Jabal Ameli *et al*. (2008) proposed a multi-objective pure integer linear programming approach for the CFP with alternative process routings and machine reliability consideration. The model minimizes total cost and maximizes system reliability simultaneously. Mahdavi and Mahadevan (2008) used sequence data to develop a construction heuristic algorithm to identify intra-cell problems. Meanwhile, Chan *et al.* (2008) proposed a two-phase GA approach to solve the CFP and intra-cell and inter-cell layout problems. A similar study was made by Ahi *et al.*

(2009). Applying multiple attribute decision making (MADM) concepts, they proposed a novel approach to determine CFP and intra-cell and inter-cell layout problems.

Table 2.2 shows a summary of previous literature. Though there have been a number of studies done on CFP, very little has been devoted to integrating cell formation, cell layout, and intracellular machine layout, the three basic steps in CMS design, simultaneously with the considerations of some real-life production factors, such as alternative process routings, operation sequences, production volume, machine reliability, and cellular layout; thereby limiting the practical nature of their approaches in a real CMS environment. Moreover, most methods in the literature assume that the NC is prescribed beforehand. However, it is difficult to determine the proper NC in the cell formation stage because the layout designer does not have any knowledge about it at the beginning. Hence, it is important and more practical to integrate the above mentioned factors simultaneously in the design of CMS.

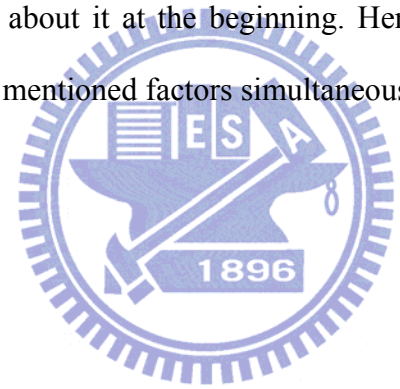


Table 2.2 Summary of literature review

Authors	Decisions			Production data				Number of cells		Method
	CF	Inter CL	Intra CL	BD	OS	APR	MR	PS	AD	
McAuley (1972)	✓			✓				✓		SCM
Carrie (1973)	✓			✓				✓		SCM
King and Nakornchai (1982)	✓			✓				✓		Array-based
Gupta *el al*. (1996)	✓	✓	✓					✓		GA
Lee *et al*. (1997)	✓				✓	✓		✓		GA
Cheng *et al*. (1998)	✓			✓				✓		GATSP
Su and Hsu (1998)	✓	✓	✓		✓			✓		SA
Sofianopoulou (1999)	✓				✓			✓		SA
Bazargan-Lari *et al*. (2000)	✓	✓	✓		✓			✓		SA
Akturk and Turkcan (2000)	✓	✓	✓		✓	✓		✓		LS
Chiang and Lee (2004)	✓		✓		✓				✓	GA
Gonçalves and Resende (2004)	✓			✓				✓		EA
Hu and Yasuda (2006)	✓				✓				✓	GA
Lei and Wu (2005)	✓	✓	✓		✓			✓		TS
Boulif and Atif (2006)	✓				✓			✓		GA
Chan *et al*. (2006)	✓	✓			✓			✓		GA
Wu *et al*. (2006)	✓	✓	✓		✓			✓		GA
Arkat *et al*. (2007)	✓				✓	✓		✓		SA
Wu *et al*. (2007a)	✓		✓		✓			✓		GA
Wu *et al*. (2007b)	✓		✓		✓			✓		GA
Yang and Yang (2008)	✓			✓				✓		ANN
Jabal Ameli and Arkat (2008)	✓				✓	✓	✓	✓		MP
Jabal Ameli *et al*. (2008)	✓				✓	✓	✓	✓		MP
Chan *et al*. (2008)	✓	✓	✓		✓	✓		✓		GA
Mahdavi and Mahadevan (2008)	✓		✓		✓				✓	Heuristic
Ahi *et al*. (2009)	✓	✓	✓		✓				✓	Novel
Unler and Gungor (2009)	✓			✓				✓		KHM
Tariq *et al*. (2009)	✓			✓				✓		GA
Mahdavi *et al*. (2009)	✓			✓				✓		GA

CF: cell formation; Inter CL: Inter-cell layout; Intra CL: intra-cell layout; BD: binary data; OS: operation sequences; APR: alternative process routings; MR: machine reliability; PS: prescribed; AD: auto-determining; LS: local search; KHM: K-harmonic means clustering algorithm.

CHAPTER 3

PROBLEM FORMULATION

As mentioned in previous chapter, two types of CFP are addressed in this thesis. One is the standard CFP with a binary machine-part incidence matrix consideration, and the other is the generalized CFP with layout design and machine reliability considerations. In this chapter, the problem formulation for these types of CFP, including problem descriptions and mathematical models, are presented.

Notations:

(1) Indices:

a : Index for operations which belongs to part I along route j (a=1,..., K_{ij})

b : Index for position number (or index for sequence of machine)

i : Index for parts (i=1,..., p)

j : Index for routings which belongs to part i (j=1,..., Q_i)

k : Index for machines (k=1,..., m)

l : Index for manufacturing cells (l=1,..., NC)

(2) Input parameters:

A_i : Unit cost of intercellular movement for part i

a_{ki} : 1, if part i is processed on machine k; 0, otherwise

B_k : Breakdown cost for machine k

$D_{l,l'}$: Distance between cell l and l'

e : The total operations in the machine-part incidence matrix

$f_{kk'}$: Flow coefficient between machines k and k'

K_{ij} : Number of operations in routing j of part i

L_m : Minimum number of machines in each cell

M_l : Set of machines in the lth cell

$MTBF_k$: Mean time between failures for machine k

m : Number of machines

m_l : Number of machines in cell l

NC : Number of cells

N_{cff} : Total number of consecutive forward flows in all the cell

N_{tf} : Total number of flows

p : Number of parts

Q_i : Number of routings for part i

r_i : Best routing selection for part i

$S_{kk'i}$: Unit flow coefficient for a part i between machines k and k'. $S_{kk'i}$=1, if part i visits machines k and k' in immediate succession; otherwise $S_{kk'i}$=0

$T_{ij}^{(a)}$: Processing time for the a-th operation of part i along route j

U_m : Maximum number of machines in each cell

$u_{ij}^{(a)}$: Index for machines which belongs to the a-th operation of part i along route j

V_i : Production volume for part i

(3) Decision variables:

Γ : Grouping efficacy

e_0 : The total number of exceptional elements

e_v : The total number of voids

X_{il} : 1, if part i locates in cell l; 0, otherwise

$X_{ijklk'l'}$: 1, if routing j of part i is selected; machine k locates in cell l and machine k' locate in cell l'; 0, otherwise

X_{lbk} : 1, if machine k locates in the b-th position of cell l; 0, otherwise

Y_{kl} : 1, if machine k locates in cell l; 0, otherwise

Z_{ij} : 1, if routing j of part i selected; 0, otherwise

3.1 Problem Description for Standard CFP

In a standard CFP, production data are given in a binary machine-part incidence matrix—a binary matrix used to indicate whether a machine is used to process a part or not—of $n \times m$ dimension. The n rows represent n machines and the m columns indicate m parts. In the $n \times m$ matrix, each binary element (a_{ki}) denotes a relationship between parts and machines where a_{ki}=1 if part i should be processed on machine k, and a_{ki}=0 otherwise. Studies usually attempt a rearrangement of rows and columns to create part families and machine cells. After the rearrangement, blocks can be observed along the diagonal of the matrix in which inter-cell movement can be minimized (i.e., the number of exceptional elements outside the diagonal block are minimized) and within-cell machine utilization maximized (i.e., the number of voids inside the diagonal block are minimized).

Figure 3.1 presents an example of the block diagonalization process of a 5 × 5 matrix. The objective is to group parts and machines of the initial matrix (Figure 3.1(a)) together into cells based on their similarities in characteristics and operating requirements to maximize grouping efficacy. The shading in Figure 3.1(b) indicates that there are two cells being formed, two '1', named 'exceptional elements', outside the diagonal block. That is, P1 and P3 will be processed on more than one machine group. Meanwhile, there is also a '0', called a 'void', inside the diagonal block. A solution without exceptional elements and voids is called a 'perfect solution'; that is, the different cells are completely independent, indicating that each part family will be processed only within a single machine group. The grouping efficacy for the matrices in Figure 3.1(a) is calculated as follows:

$$\Gamma = \frac{e - e_0}{e + e_v} = \frac{13 - 2}{13 + 1} = 78.57\% .$$

M\P	P1	P2	P3	P4	P5
M1	0	0	1	1	0
M2	0	1	1	0	1
M3	1	0	0	1	0
M4	1	1	1	0	1
M5	1	0	0	1	0

(a) Initial matrix

M\P	P2	P3	P5	P1	P4
M2	1	1	1	0	0
M4	1	1	1	1	0
M1	0	1	0	0	1
M3	0	0	0	1	1
M5	0	0	0	1	1

(b) Matrix after rearrangement

Figure 3.1 Rearrangement of rows and columns of matrix to create cells

3.2 Mathematical Model for Standard CFP

3.2.1 Assumptions

The mathematical model for standard CFP in this thesis is formulated on the basis of the following assumptions:

(1) All parts are assigned to part families.

(2) All machines are assigned to machine cells.

(3) All machines are non-identical.

(4) Each part family has at least one part, but does not have total number of parts at most.

(5) Each machine cell has at least one machine, but does not have total number of machines at most.

(6) The binary machine-part incidence matrix is the main input information. Other production information, such as alternative process routings, operation sequences, production volume, production times, machine reliability, and different cellular layout type are not considered in the standard CFP.

3.2.2 Mathematical formulation

By using the above notations and assumptions, the proposed mathematical model maximizing grouping efficacy for standard CFP can be formulated as follows:

$$Max \quad \Gamma = \frac{e - e_0}{e + e_v} \tag{3.1}$$

Subject to:

$$e = \sum_{i=1}^{p} \sum_{k=1}^{m} a_{ik} \tag{3.2}$$

$$e_v = \sum_{l=1}^{NC} \left(\sum_{i=1}^{p} X_{il} \times \sum_{k=1}^{m} Y_{kl} \right) - \sum_{l=1}^{NC} \sum_{i=1}^{p} \sum_{k=1}^{m} a_{ik} X_{il} Y_{kl} \tag{3.3}$$

$$e_0 = e - \sum_{l=1}^{NC} \sum_{i=1}^{p} \sum_{k=1}^{m} a_{ik} X_{il} Y_{kl} \tag{3.4}$$

$$\sum_{l=1}^{NC} X_{il} = 1 \qquad \forall i \tag{3.5}$$

$$\sum_{l=1}^{NC} Y_{kl} = 1 \qquad \forall k \tag{3.6}$$

$$\sum_{k=1}^{m} Y_{kl} \geq L_m \qquad \forall l \tag{3.7}$$

$$X_{il},\ Y_{kl} \in \{0,1\} \quad \forall i,k,l \tag{3.8}$$

In the above model, Eq. (3.1) is the objective function that seeks maximization of grouping efficacy. Eqs. (3.2), (3.3), and (3.4) show the calculation of the total operations in the machine-part incidence matrix, the total number of voids, and the total number of exceptional elements, respectively. Eq. (3.5) provides a restriction that each part will be assigned to exactly one cell, while Eq. (3.6) provides a restriction that each machine will be assigned to exactly one cell. Eq. (3.7) assigns the lower limit of the cell size and Eq. (3.8) indicates that X_{il} and Y_{kl} are 0–1 binary decision variables.

Some studies allow the existence of singletons (cells having less than two machines) in the solutions and some don't. In this thesis, we use Eq. (3.7) to integrate both situations. When L_m equal to 1, the existence of singletons are allowed, while singletons are not allowed, when L_m equal to 2.

3.3 Problem Description for Generalized CFP

In the standard CFP, the binary machine-part incidence matrix is the main production data. Some real-life production factors, such as alternative process routings, operation sequences, production volume, machine reliability, and cellular layout, are not addressed in the design of CMS, thereby limiting the practical applied in a real CMS environment. Hence,

a generalized CFP that incorporates the abovementioned factors in the design of CMS is introduced. These factors are described in detail next.

(1) Alternative process routings

A process routing for a given part is the set of machines passed by this specific part. In most CF methods, parts are assumed to have a unique part process plan. However, it is well known that alternatives may exist in any level of a process plan. In some cases, there may be many alternative process plans for making a specific part, especially when the part is complex (Qiao *et al*., 1994). In the case shown in Figure 3.2(a), part #1 has three process routings: R1, R2, and R3. When introducing alternative process routings to CFP, the grouping of parts can be more effective due to the flexibility of the routes. However, it leads to a more complex problem than the standard CFP. Under this circumstance, not only the formation of part families and machine cells must be determined but also the selection of routings for each part to achieve decision objectives, such as the minimization of intercellular movement. For instance, Figure 3.2 (b) provides a feasible solution to the sample problem of Figure 3.2 (a) where routing #2 is selected by all parts.

PN	P1			P2		P3		P4		P5	
PV	50			30		20		30		20	
RN	R1	R2	R3	R1	R2	R1	R2	R1	R2	R1	R2
M1			2		2	2		1	1		1
M2		1	1	1			2				
M3	2			2	1				2	1	
M4	1	2				1	1	2		2	

(a) Problem data

PN	P1	P3	P2	P4	P5
PV	50	20	30	30	20
RN	R2	R2	R2	R2	R2
M2	1	2			
M4	2	1			
M1			2	1	1
M3			1	2	

(b) Final solution

Figure 3.2 Cell formation with alternative process routings

(2) Cellular layout

In CMS, different cellular layout type and cellular layout sequence will affect the inter-cell move distance (ICMD). They are described as follows.

(a) Determination of cellular layout type

Cellular layout is represented in Figure 3.3, where r is the number of rows, c is the number of columns, and (X_r,Y_c) is the coordinate of cell cr. The cellular layout type can be determined, while the value of r is set by the layout designer. For example, when r=1, the cellular layout type is linear single-row layout (Figure 3.4 (a)); and when r=2, the cellular layout type is linear double-row layout (Figure 3.4 (b)), where NC is the number of cells.

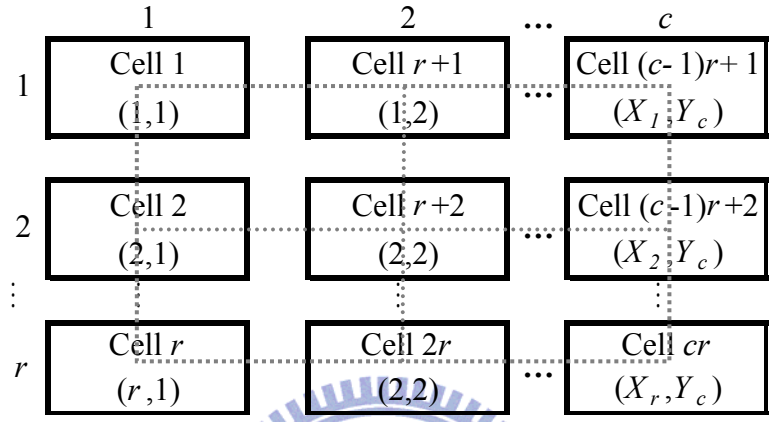


Figure 3.3 Cellular layout

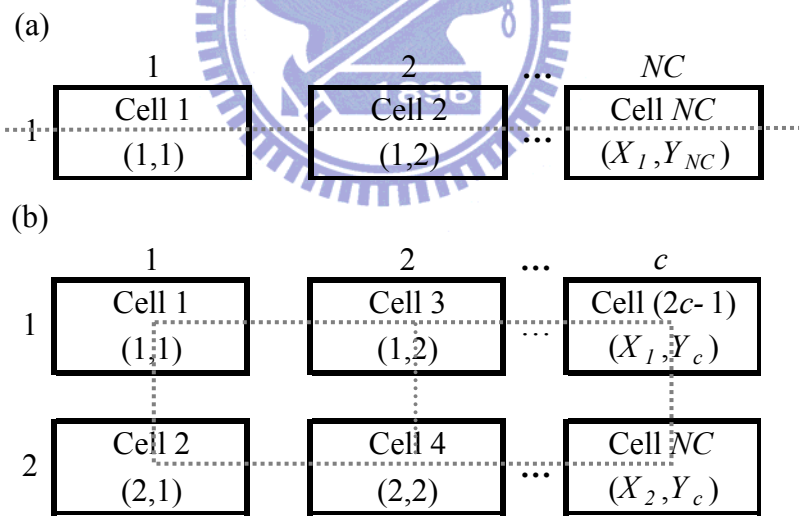


Figure 3.4 Two typical cellular layouts: (a) linear single-row layout (r=1) (b) linear double-row layout (r=2)

(b) Inter-cell move distance

There are two popular methods for measuring ICMD between a pair of cells l and l' (Tam and Li, 1991). They are:

Cartesian method: $D_{l,l'} = \left[\left(X_{l'} - X_l \right)^2 + \left(Y_{l'} - Y_l \right)^2 \right]^{1/2}$, (3.9)

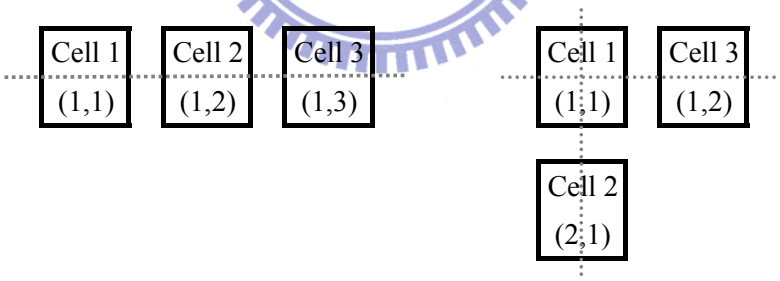
Manhattan method: $D_{l,l'} = \left| X_{l'} - X_l \right| + \left| Y_{l'} - Y_l \right|$, (3.10)

where $\left(X_l, Y_l \right)$ and $\left(X_{l'}, Y_{l'} \right)$ are the coordinates of the measuring points of cells l and l'.

In terms of measuring points, we can use either: (a) the centroid of a cell site or (b) the nearest point between adjacent cells. In this thesis, the Cartesian method was chosen and the centroid of a cell site will be used for calculating ICMD. Thus, the ICMD between cells 1 and 2 in Figure 3.4(a) is equal to $1 = \left[\left(1-1 \right)^2 + \left(2-1 \right)^2 \right]^{1/2}$.

(c) Effects of cellular layout type

Different cellular layout types will result in different ICMD. Figure 3.5 shows that the ICMD between cells 1 and 3 in Figure 3.5(a) will be twice the distance moved between cells 1 and 3 in Figure 3.5(b). Hence, the cellular layout type is an important issue in CMS design.



(a) linear single-row layout (r=1) (b) linear double-row layout (r=2)

Figure 3.5 Two typical cellular layouts (NC=3)

(d) Effects of cellular layout sequence

We present an example to illustrate the effects of cellular layout sequence. If the NC is equal to three and a linear single-row layout (r=1) is considered as shown in Figure 3.5(a), then ICMD between cells 1 and 3 will be twice the distance moved between cells 1 and 2 or between cells 2 and 3. When a linear double-row layout (r=2) is considered as shown in

Figure 3.5(b), the corresponding ICMD between cells 2 and 3 will be $\sqrt{2}$ times the distance moved between cells 1 and 2 or between cells 1 and 3. Hence, the cellular layout is an important issue in CMS design.

(3) Operation sequence and production volume

The operation sequence and production volume of each part affects the machine cell formation significantly. Therefore, both operation sequence and production volume of each part should be incorporated in the analysis of CM systems. For example, a simple CFP consists of four machines (M1, M2, M3, M4) and two parts (P1, P2) with part routes (M2, M1) for P1 and (M4, M1, M3) for P2. Suppose that the annual demands of P1 and P2 are 40 units and 60 units, respectively. Two cell formation results are shown in Figure 3.6, where the number in each entry indicates the visiting order of part to machine. If we do not consider machine sequence in calculating inter-cell movement, the solution in Figure 3.6 (a) is better than that in Figure 3.6(b) because there are 60 inter-cell movements in Figure 3.6 (a) and 100 inter-cell movements in Figure 3.6 (b). However, if the machine sequence is considered, the solution in Figure 3.6(b) is better, because the sum of inter-cell movements in Figure 3.6(a) is 120 (60 × 2) compared with 100 (40 + 60) in Figure 3.6(b). If we do not consider production volume in calculating inter-cell movement, the solutions of Figure 3.6(a) and (b) are the same with 2. However, if the manufacturing volumes are considered, the solution in Figure 3.6(b) is better, because the sum of inter-cell movements in Figure 3.6(a) is 120 (60 × 2) compared with 100 (40 + 60) in Figure 3.6(b).

(a)

	P1	P2
PV	40	60
M1	2	2
M2	1	
M3		3
M4		1

(b)

	P1	P2
PV	40	60
M2	1	
M3		3
M1	2	2
M4		1

Figure 3.6 An example for the affect of operation sequence

(4) Machine reliability

A number of previous works assumed that all machines are 100% reliable. However, this is not always the case. Machines are key elements in manufacturing systems and oftentimes it is not possible to handle their collapse as quickly as production requirements dictate. Their collapse can dramatically affect system performance measures and bring about detrimental effects on due date performance. Hence, machine reliability should be taken into account during the design of CMS to improve the overall performance of the system (Jeon *et al*., 1998).

A common way of dealing with machine reliability in the design phase of a manufacturing system is by the evaluation of the quantities of the mean time between failures (MTBF). MTBF can be obtained by taking the reciprocal of λ, where λ is the machine failure rate. As long as the breakdown cost for each machine is known in advance, the cost caused by machine unreliability can be acquired after simple calculation. Jabal Ameli and Arkat (2008) have presented a mathematical approach to calculate machine breakdown cost (MBC) that involves dividing production time by MTBF and then multiplying this quantity by the unit MBC (Eq. (3.12)).

3.4 Mathematical Model for Generalized CFP

As mentioned in the previous section, it is important and more practical to integrate the abovementioned factors simultaneously in the design of CMS. Cell formation, cell layout, and intracellular machine layout are three major steps in the design of CMS. Ideally, these steps should be addressed simultaneously in order to obtain the best results. However, this is not easy to do due to the NP-complete nature of each step and the limitations of traditional approaches. Moreover, intracellular machine layout is a detailed layout planning. It usually starts after the cell formation and cell layout decisions have been determined. Hence, a two-stage multi-objective mathematical programming model is formulated in this section to

integrate cell formation, inter-cell layout, and intracellular machine layout problem with considerations of alternative process routings, operation sequences, production volume, machine reliability, and different cellular layout type. The framework of the proposed two-stage model is given in Figure 3.7. The aim of stage I is to solve cell formation and inter-cell layout simultaneously and the primary work of stage II is to determine machine layout (sequence) in each cell based on the given cell formation determined in stage I.

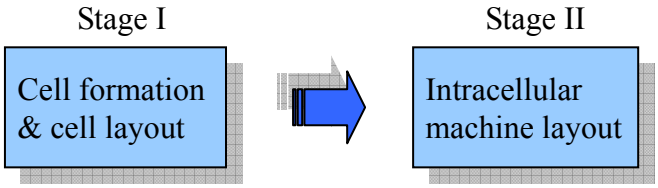


Figure 3.7 The framework of the proposed two-stage model for generalized CFP

3.4.1 Assumptions

The mathematical model for generalized CFP in this research is formulated on the basis of the following assumptions:

(1) All parts are assigned to part families.

(2) All machines are assigned to machine cells.

(3) All machines are non-identical.

(4) The type of cellular layout and the distance moves between cells are known a priori.

(5) Operation requirements, including operation sequence, operation time, and production volume, are known.

(6) Inter-cell part transportation unit cost for each part, breakdown cost, and MTBF for each machine are known.

(7) The limitation of total number of machines in each cell is user-defined.

(8) The intra-cell move distances for each part are not considered.

3.4.2 Mathematical formulation

By using the above notations and assumptions, the proposed two-stage multi-objective mathematical programming models are formulated, one for each stage, and are presented here.

3.4.2.1 Stage I: Cell formation and inter-cell layout

The aim of this stage is to solve cell formation and inter-cell layout simultaneously in terms of minimization of total inter-cell move cost (ICMC) and MBC. The multi-objective 0-1 integer programming model is given below.

Total ICMC:

$$ICMC = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{a=1}^{K_{ij}-1}\sum_{l=1}^{NC}\sum_{l'=1}^{NC} Z_{ij} Y_{(u_{ij}^{(a)})l} Y_{(u_{ij}^{(a+1)})l'} V_i D_{l,l'} A_i \tag{3.11}$$

Total machine breakdown cost:

$$MBC = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{a=1}^{K_{ij}} Z_{ij} \frac{V_i T_{ij}^{(a)} B_{(u_{ij}^{(a)})}}{MTBF_{(u_{ij}^{(a)})}} \tag{3.12}$$

The multi-objective function is as follows:

$$Min\ TC = ICMC + MBC \tag{3.13}$$

Subject to:

$$\sum_{j=1}^{Q_i} Z_{ij} = 1, \forall i \tag{3.14}$$

$$L_m \le \sum_{k=1}^{m} Y_{kl} \le U_m \quad \forall l \tag{3.15}$$

$$\sum_{l=1}^{NC} Y_{kl} = 1 \quad \forall k \tag{3.16}$$

$$Y_{kl}, Z_{ij} \in \{0,1\} \quad \forall i, j, k, l \tag{3.17}$$

In the above model, Eqs. (3.11) and (3.12) show the calculation of the total inter-cell part transportation cost and MBC, respectively. Eq. (3.13) is the objective function that seeks the minimization of total cost of inter-cell part transportation cost and machine

breakdown. Eq. (3.14) indicates that only one process routing will be assigned to each part, while Eq. (3.15) assigns the upper and lower limits of the cell size. Eq. (3.16) provides a restriction that each machine will be assigned to exactly one cell and Eq. (3.17) indicates that Y_{kl} and Z_{ij} are 0–1 binary decision variables.

Obviously, the objective function is in a non-linear form and thus may require extensive computational efforts for current commercial solvers to obtain possibly local optimal solutions. A linearization approach (Jabal Ameli *et al*., 2008) for converting a non-linear model into linear form is adopted. The transformation equation is as follows.

$$X_{ijklk'l'} = Z_{ij} Y_{kl} Y_{k'l'} \tag{3.18}$$

Where:

$$X_{ijklk'l'} = \begin{cases} 1 & \text{if routing } j \text{ of part } i \text{ is selected, machine } k \text{ locates in cell } l \text{ and machine } k' \text{ locate in cell } l'. \\ 0 & \text{otherwise.} \end{cases}$$

Linearization Constraints:

$$X_{ijklk'l'} \le Z_{ij}, \forall i, j, k, k', l, l' \tag{3.19}$$

$$X_{ijklk'l'} \le Y_{kl}, \forall i, j, k, k', l, l' \tag{3.20}$$

$$X_{ijklk'l'} \le Y_{k'l'}, \forall i, j, k, k', l, l' \tag{3.21}$$

$$Z_{ij} + Y_{kl} + Y_{k'l'} - X_{ijklk'l'} \le 2, \forall i, j, k, k', l, l' \tag{3.22}$$

$$Y_{kl}, Y_{k'l'}, Z_{ij}, X_{ijklk'l'} \in \{0,1\} \quad \forall i, j, k, k', l, l' \tag{3.23}$$

The first three linearization constraints (Eqs. 3.19–3.21) ensure that if one of the primary binary variables has a zero value, then their corresponding new variables will take a zero value as well. The last constraint (Eq. 3.22) ensures that if all primary variables take unit values, then their corresponding new variables take unit values as well. We rewrite the objective function as follows:

$$Min\ TC = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{k=1}^{K_{ij}-1}\sum_{l=1}^{NC}\sum_{l'=1}^{NC} X_{ij u_{ij}^{(k)} l u_{ij}^{(k+1)} l'} V_i D_{l,l'} A_i + \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{k=1}^{K_{ij}} Z_{ij} \frac{V_i T_{ij}^{(k)} B_{(u_{ij}^{(k)})}}{MTBF_{(u_{ij}^{(k)})}} \quad (3.24)$$

Subject to:

Eqs. 3.14 - 3.16 and Eqs. 3.19 - 3.23

This new form of the objective function is in a linear form. Thus, linear programming software, such as Lingo 8.0, can solve this model.

3.4.2.2 Stage II: Intracellular machine layout

The parts being transported from one machine to another within a cell are called intra-cellular flow. Intra-cellular part flows are usually rushed and short in distances. In CMS, these movements are very frequent, and the frequency directly affects the intracellular machine layout design. Based on the classification scheme of Aneke and Carrie (1986), intracellular flow can be classified into four categories (Figure 3.7): (1) repeat operation, R; (2) forward flows, FF; (3) by-pass movement, BP; and (4) reverse flows, RF. The ideal material flow in a good layout design should be mostly consecutive forward flows (CFF). The CFF usually has the benefits of smaller flow distance, easier control of the production process, and easier material handling (Ho *et al*., 1993).

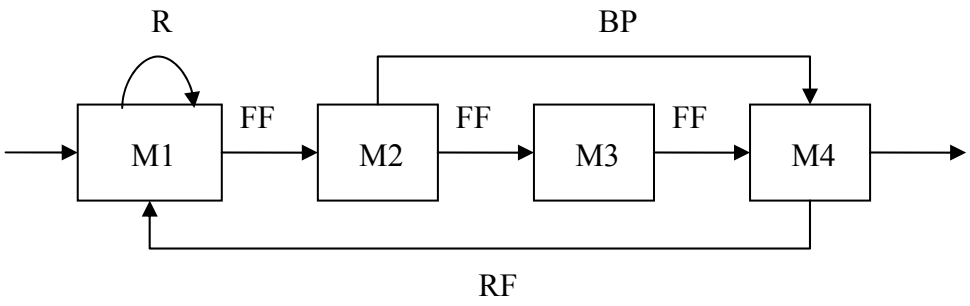


Figure 3.8 Intracellular part flows

Since the CFF is a good indicator of the goodness of the solution, Mahdavi and Mahadevan (2008) developed a flow matrix on the basis of the number of CFF between a pair of machines and used it as the basic input to the grouping and layout problem. However,

their method did not consider the effect of manufacturing volumes. As mentioned in Section 3.3, taking the effect of manufacturing volumes into account is more realistic when designing a performance measure for intracellular machine layout. A flow matrix with manufacturing volumes consideration is thus proposed here. The flow matrix (F) is re-defined as follows:

$$F = [f_{kk'}] = \sum_{i=1}^{p} V_i S_{kk'i} \quad \forall k,k' \text{ and } k \neq k' \tag{3.25}$$

Based on the flow matrix, a CFF index (CFFI) for measuring intracellular machine layout is proposed in this section. The CFFI is defined as the ratio of total number of CFFs in all cells (N_{cff}) to the total number of flows (N_{tf}).

$$\text{CFFI} = \frac{N_{cff}}{N_{tf}} \tag{3.26}$$

where

$$N_{cff} = \sum_{l=1}^{NC} \sum_{b=1}^{m_l-1} \sum_{k \in M_l} \sum_{k' \in M_l} f_{kk'} X_{lbk} X_{l(b+1)k'} \quad k \neq k' \tag{3.27}$$

$$N_{tf} = \sum_{i=1}^{p} (K_{i(r_i)} - 1) V_i \tag{3.28}$$

The primary goal of the second stage is to determine the machine layout (sequence) in each cell in terms of maximizing the CFFI based on the cell formation determined in stage one. The model is given below.

$$Max \ \text{CFFI} \tag{3.29}$$

Subject to:

$$\sum_{k \in M_l} X_{lbk} = 1 \quad \forall l,b \tag{3.30}$$

$$\sum_{b=1}^{m_l} X_{lbk} = 1 \quad \forall l, k \in M_l \tag{3.31}$$

$$X_{lbk} \in \{0,1\} \quad \forall l,b,k \tag{3.32}$$

In the above model, Eq. (3.29) is the objective function that seeks the maximization of CFFI. Eqs. (3.30) and (3.31) ensure that each position is assigned to one machine and each machine is assigned to exactly one position. Eq. (3.32) indicates that X_{lbk} is a 0–1 binary decision variable.

Due to the combinatorial nature of the above models, good heuristic approaches should be more appropriate than the exact method in terms of solution efficiency, especially for large-sized problems. Thus, in the next chapter, we develop two fast and effective two-stage approaches to solve these complex problems.

CHAPTER 4

PROPOSED ALGORITHMS

In the previous chapter, two mathematical models representing standard CFP and generalized CFP have been formulated. Due to the NP-hard nature of the presented mathematical formulations, solving these problems through a traditional optimization technique is difficult and impractical. Furthermore, as mentioned in chapter two, meta-heuristic algorithms such as SA, TS, and WFA, have been the most successful solution approaches to provide global or near-global optimal solutions within a reasonable computation time, and SCM-based methods are more flexible in incorporating various production data into the machine-part clustering process. Thus, two hybrid meta-heuristic algorithms based on SCM-based clustering algorithm and SA/TS/WFA are proposed to solve the complex problems.

Before proposed algorithms are described, some notations used in this chapter are introduced first.

α	:	Cooling rate
counter_iter	:	Number of iterations
counter_stag	:	Number of times the incumbent solution did not improve
counter_mut	:	Number of times the mutation strategy has been implemented
C^*	:	Optimal number of cells
f(S)	:	Value of object function in solution *S*
L	:	Markov chain length
N_{max}	:	Maximum number of iterations
NC	:	Number of cells
N^F	:	Set of feasible solutions
N^C	:	Set of solutions without violating cell cardinality constraints
N^T	:	Set of solutions in tabu status
N^A	:	Set of solutions satisfying aspiration criterion

Stag_check	:	Maximum number of solution has not been improved
S^0	:	Initial solution
S	:	Current solution
S^N	:	Neighborhood solution
S^*	:	Incumbent solution of current cell size
S^{**}	:	Best solution found so far
T_0	:	Initial temperature
T_f	:	Final temperature

4.1 Proposed Algorithms for Standard CFP

Most algorithms designed to solve CFP attempt to obtain the machine-part groupings so that some decision objectives, such as grouping efficiency or grouping efficacy, can be maximized. However, without prior determination of the NC, the abovementioned objectives can hardly be achieved. It is given beforehand in a few cases, but is left to be determined as part of the decision in most. Usually, in the iterative solution process, the initial NC is set at two and is gradually increased by one unit. These algorithms are then repeatedly applied until the NC resulting in the best grouping efficiency/efficacy value becomes established. Thus, many computational efforts have to be exerted in order to obtain the optimal NC. Instead of using a beginning number as the starting point, identifying a good intermediate point for the NC at the very beginning should save plenty of run time when designing an algorithm to search for the optimal NC.

We present a test problem from literature (Carrie, 1973) as an example. The relationship between the NC and the resulting grouping efficacy is shown in Figure 4.1. Grouping efficacy value increases as the NC increases, and the optimal/near-optimal value is achieved when cell size is nine. Afterwards, efficacy starts to decrease as the NC increases. Similar observations can be found in other test problems. Based on this, the NC can be automatically calculated and determined such that the best grouping efficacy may result in.

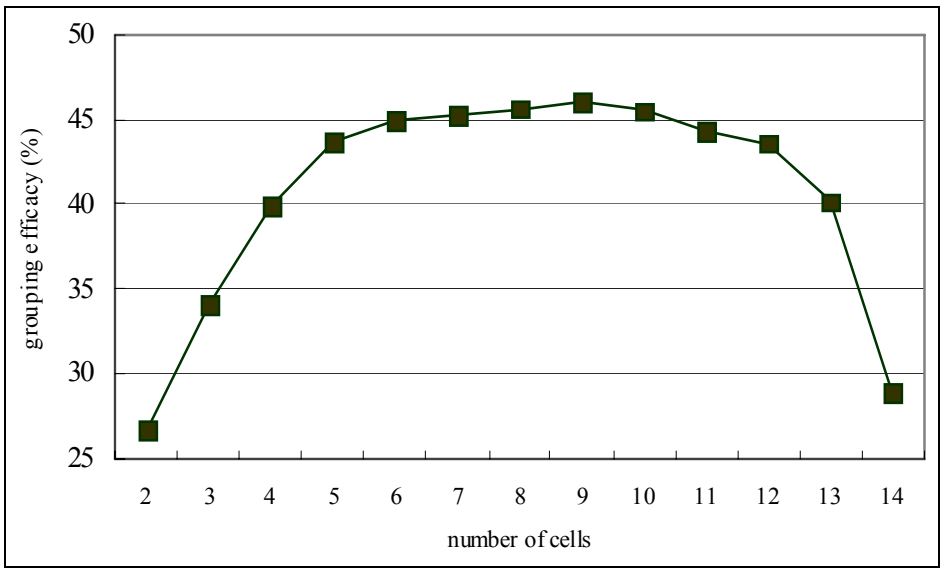


Figure 4.1 Relationship between grouping efficacy and number of cells

Based on the above discussion, we propose a two-stage hybrid algorithm HCFA to solve the standard CFP. The framework of the proposed two-stage HCFA is given in Figure 4.2. In the first stage, the SCM-based clustering algorithm is adapted to derive NC quickly. NC value is then used as input to the second stage to search for the optimal/near-optimal solution through the proposed SA/TS/WFA algorithm. We anticipate that NC obtained in stage one can serve as a good lower boundary to start the solution process in stage two. Hence, a considerable amount of computational efforts can be saved, especially when large-sized problems are solved. The procedures for both stages are described below.

Stage I of HCFA:

Step 1. Set $NC = 2$, $f(S^0) = f(S^*) = 0$.

Step 2. Apply the SCM-based clustering algorithm to generate an initial solution S^0.

Step 3. If $f(S^0) > f(S^*)$, then set $S^* \leftarrow S^0$, $C^* = NC$, $NC = NC + 1$, go to Step 2; otherwise, report incumbent cell configuration found: S^*, C^*, and terminate stage one.

The solution obtained at the end of stage one, including the suggested NC (C^*) and cell configurations (S^*), is then used as the input in stage two to search for the

optimal/near-optimal solution through the proposed SA/TS/WFA procedure.

Stage II of HCFA:

Step 1. Read solutions from stage one, including C^* and S^*.

Step 2. Set $NC = C^*$, $f(S^0) = f(S^*), f(S^{**}) = 0$, go to Step 4.

Step 3. Apply the SCM-based clustering algorithm to generate an initial solution S^0.

Step 4. Apply SA/TS/WFA procedure to improve S^0 and generate an incumbent solution S^*.

Step 5. If $f(S^*) > f(S^{**})$, then set $S^{**} \leftarrow S^*$, $C^* = NC$, NC = NC+1, go to Step 3; otherwise, report the current best cell configuration (S^{**}) and NC (C^*), and terminate stage two.

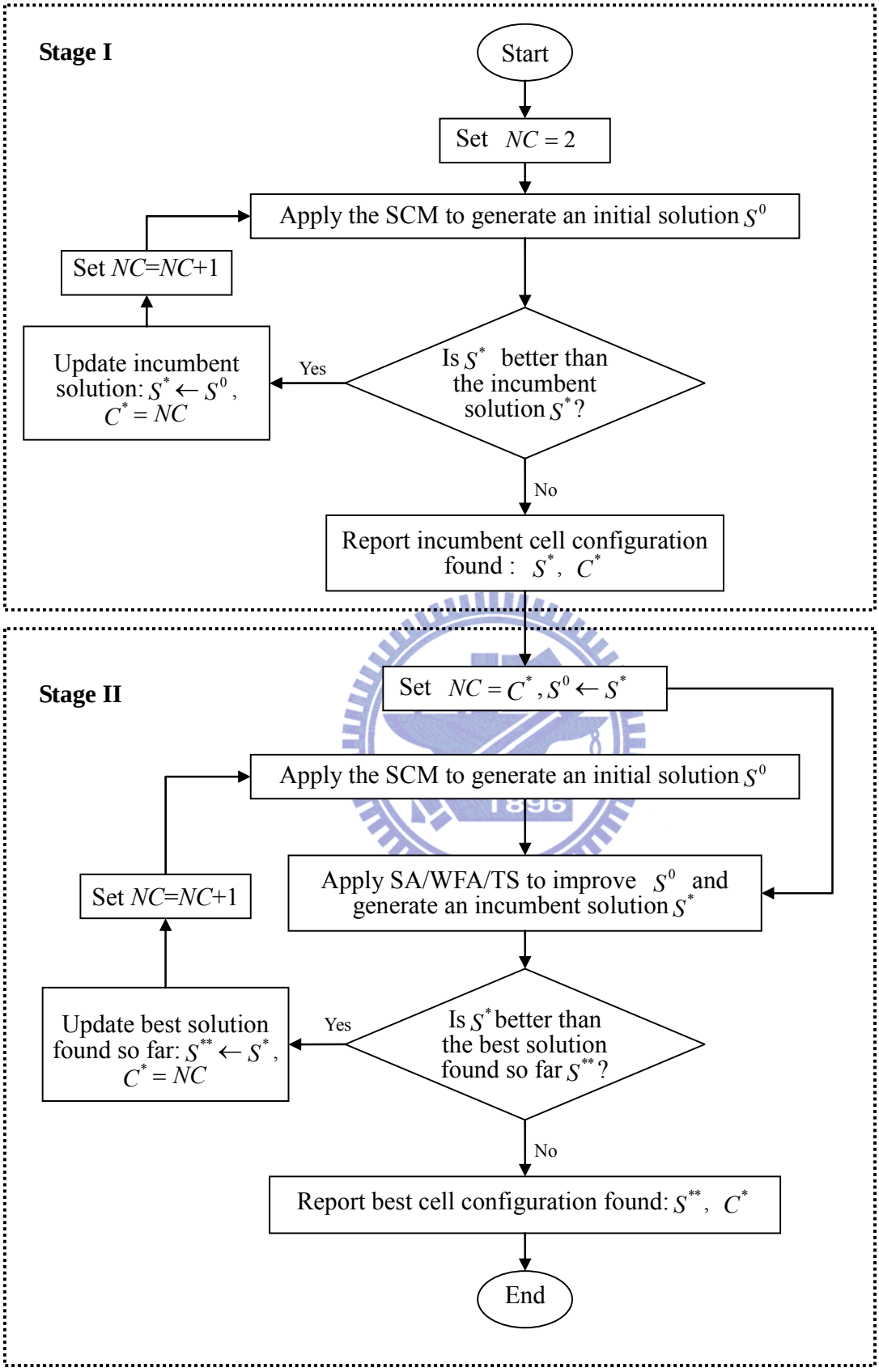


Figure 4.2 Two-stage approach: Hybrid Cell Formation Algorithm (HCFA)

The SCM-based clustering algorithm and SA/TS/WFA are primary algorithms that consist of HCFA. The details of them are described as follows.

4.1.1 SCM-based clustering algorithm

As mentioned in Section 2.3, SCMs are more flexible in incorporating various production data into the machine-part clustering process. Hence, this study proposes the use of an SCM-based clustering algorithm to generate quick initial solutions, which will then be later improved by SA/TS/WFA method. It is well known that decomposing an originally difficult problem into several sub-problems usually increases problem-solving efficiency. Since the CFP considers the grouping of machines and parts, an intuitive solution approach is to decompose the entire problem into two sub-problems dealing with the assignment of machines and parts, respectively. In our construction of the initial solution, machine assignment is determined in the first stage, while the assignment of parts is achieved in the second stage.

Our approach for generating initial solutions consists of three steps: (1) computing similarity values between machine pairs and constructing a similarity matrix, (2) using a clustering rule to process the values in the similarity matrix and forming machine cells, and (3) assigning parts to machine cells using a parts assignment procedure. Details of them are described here.

(1) Machines assignment

As mentioned in Section 2.2, the Jaccard similarity coefficient is the most stable similarity coefficient. Hence, Jaccard's similarity measure is used to evaluate similarity between machines as an important index for assigning machines to cells in this sub-problem. The similarity measure, denoted by S_{ij}, is defined as $S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij} + c_{ij}}$, where a_{ij} represents the number of parts processed by both machines i and j; b_{ij} is the number of parts processed

by machine i but not by machine j; and c_{ij} is the number of parts processed by machine j but not by machine i. After calculating the similarity matrix for each pair of machines, we are able to generate the initial machines assignment by using the following greedy rule: the higher the similarity measure of a pair of machines, the higher priority they have for placement in the same cell. This process is repeated until all machines have been assigned to cells. For the sample machine-part matrix in Figure 4.3(a), the corresponding similarity matrix for machines is displayed in Figure 4.3(b). Assuming that two cells are to be formed, the largest coefficient in the matrix of Figure 4.3(b) is 0.67, indicating that machines 2 and 4 must be assigned to the same cell, e.g. cell 1. We proceed to the second largest coefficient in the matrix, 0.5, appearing in pairs (1, 3) and (1, 5). Since these three machines do not have any relationship with any machines in cell 1, they should be assigned together to the next cell, cell 2. Figure 4.4 shows the machines assignment using the proposed greedy rule.

	P1	P2	P3	P4	P5
M1	1	0	0	1	0
M2	0	1	1	0	1
M3	1	0	0	0	0
M4	0	1	1	0	0
M5	0	0	0	1	0

(a) Machine-part matrix

	M1	M2	M3	M4	M5
M1	-	0	0.50	0	0.50
M2		-	0	0.67	0
M3			-	0	0
M4				-	0
M5					-

(b) Similarity matrix for machines

Figure 4.3 Machine-part matrix and corresponding similarity matrix for machines

		P1	P2	P3	P4	P5
Cell 1	M2	0	1	1	0	1
	M4	0	1	1	0	0
Cell 2	M1	1	0	0	1	0
	M3	1	0	0	0	0
	M5	0	0	0	1	0

Figure 4.4 Assignment of machines

(2) Parts assignment

In this procedure, the parts are assigned to cells so that the number of voids and exceptional elements—major components comprising the formula of grouping

efficacy—are explicitly considered. It can be summarized as follows:

Step 1. Read the results of machines assignment.

Step 2. For each part, find the cell to which a part assignment will result in the least sum of number of exceptional elements and number of voids. If a tie happens, assign the part to a cell with the least number of voids.

Step 3. Repeat Step 2 until all parts have been assigned to cells.

Results of parts assignment shown in Figure 4.5 demonstrate this procedure. After calculating the sum of numbers of voids and exceptional elements for each part-cell combination, parts 2, 3, and 5 are assigned to cell 1, while parts 1 and 4 are assigned to cell 2. The initial solution matrix for this CFP can thus be obtained and the configuration for this initial solution can be represented by Figure 4.5.

		P2	P3	P5	P1	P4
Cell 1	M2	1	1	1	0	0
	M4	1	1	0	0	0
Cell 2	M1	0	0	0	1	1
	M3	0	0	0	1	0
	M5	0	0	0	0	1

Figure 4.5 Initial solution matrix obtained

4.1.2 SA/TS/WFA algorithms

When designing a heuristic search algorithm, several important considerations should be kept in mind. The first is to develop a mechanism for searching the neighborhood solutions for improvement. Since the neighborhood will be searched next, the choice of neighborhood function will strongly influence the direction of the search. Another consideration is the mechanism for allowing escape from local optima and for settling only in a global optimum. Based on these concepts, three algorithms, namely HSAM, HWFAM, and HTSM, are developed in this section.

(1) Configuration

An easy way to represent the configuration of a feasible solution to CFP is through a

string, whose size is equal to the number of machines/parts. The *j*th bit of the string stores the identifier of the cell to which the machine/part is assigned. For example, Figure 4.6 is the configurations for machine cells and part families. In such a configuration, the string (2, 1, 2, 1, 2) in Figure 4.6(a) indicates that machines 2 and 4 are assigned to cell 1, while machines 1, 3, and 5 are assigned to cell 2; the string (2, 1, 1, 2, 1) in Figure 4.6(b) represent that machines 2, 3, and 5 are assigned to cell 1, while parts 1 and 4 are assigned to cell 2.

Machine #	1	2	3	4	5
Cell #	2	1	2	1	2

(a) Configuration for machine cells

Part #	1	2	3	4	5
Cell #	2	1	1	2	1

(b) Configuration for part families

Figure 4.6 Configuration of a feasible solution to the CFP

(2) Insertion-move operation

In this study, the insertion-move operation is applied as a mechanism for searching the neighborhood solutions for improvement. It moves a machine k from its current cell l (source cell) to a new cell l' (destination cell). The new move is denoted as (l', k). A move that results in the greatest improvement of the objective function value from the current solution is selected. That is,

$$Z(l',\ k) = Max\{obj^{(l',k)} - obj^{(l,k)}, \forall l, l' \in N^F, l' \neq l, \forall k \in M\} \tag{4.1}$$

where $obj^{(l,k)}$ is the objective function value; N^F is the set of feasible solutions; and M is the set for machines.

(3) Mutation strategy

The mutation strategy of GA aims to increase the probability of finding more "diversified" solutions in order to bring the searching process to a new and unexplored

solution space, thus ensuring that large areas of the space are searched. In this study, the mutation strategy *mut_check* is implemented when the number of moves has not been improved within a certain number of iterations. This performs an exchange of a machine to any cell other than the current one based on a prescribed probability β. That is, all machines have the probability of changing cell when machine mutation is applied. For each machine in the incumbent solution, a random number from (0, 1) is first drawn. If the value is greater than β, then the machine is exchanged with another randomly determined cell; otherwise, it stays in the current cell. Through this strategy, the search is able to explore a large solution space, thereby enhancing the possibility of finding the optimum solution in a very short time. The procedure of the mutation strategy in the pseudo-code format is shown in Figure 4.7.

```
Mutation_strategy ( β )
{
    Let the current solution (S) equal to the current best solution (S*).
     FOR each machine DO
    {
        Generate a random number  r ∈ U(0,1) .
        IF ( r > β )
            Exchange machine with any cells other than the current one.
        ELSE
            Stay machine in the current cell.
    }
}
```

Figure 4.7 Pseudo code of mutation strategy

4.1.2.1 SA-based algorithm (HSAM)

As mentioned in Section 2.6, the main disadvantages of SA are as follows: (1) high execution time, (2) ease of being trapped to local minima if the cooling speed is too fast or the initial temperature is not high enough, and (3) difficulty of obtaining a globally optimum

solution if the search cannot reach the equilibrium state at each temperature. In this study, two types of mechanisms, the insertion-move and the mutation strategy of GA, are utilized to construct a hybrid SA method called HSAM to address these issues. Both mechanisms play different roles in the process of solution improvement. We use insertion-move as a primary tool for finding better neighborhood solution, while employing mutation strategy to increase the probability of finding more "diversified" solutions to bring the searching process to a new and unexplored solution space. The pseudo-code format of the proposed procedure HSAM is diagrammed in Figure 4.8 and described in detail below.

Algorithm HSAM

Step 1. Read initial solution S^0.

Step 2. Initialization: Let *counter_MC* = 0, $T = T_0$, $S \leftarrow S^0$, $S^* \leftarrow S^0$.

Step 3. If *counter_MC* < *L*, then repeat Steps 3.1 to 3.5; otherwise, go to Step 4.

Step 3.1. If *counter_mut* ≥ *mut_check*, then apply the **mutation strategy** to generate a new current solution *S* and let *counter_mut* = 0.

Step 3.2. Generate a best solution $S^N \left(S^N \in N^C\right)$ in the neighborhood of S by performing the **insertion-move operation**.

Step 3.3. Compute $\Delta = f\left(S^N\right) - f\left(S\right)$. If $((\Delta > 0)$ or $(e^{\Delta/T} > r \in U(0,1)))$, then let $S \leftarrow S^N$, *counter_mut* = 0; otherwise, *counter_mut* = *counter_mut* + 1.

Step 3.4. If $(f(S^N) > f(S^*))$, then let $S^* \leftarrow S^N$.

Step 3.5. Let *counter_MC*= *counter_MC* + 1, go to Step 3.

Step 4. If $T \le T_f$, then report the best solutions so far, and stop the algorithm; otherwise, $T = T \times \alpha$, *counter_MC* = 0, go to Step 3.

Note that the algorithm starts from an initial solution in Step 1, after which all algorithmic parameters and counters are initialized in Step 2. As long as the value of *counter_mut* is smaller than *mut_check*, a new neighborhood solution is generated through

the insertion-move in Step 3.2; otherwise, mutation strategy is applied to generate a new solution with higher degree of diversification in Step 3.1. If the newly generated neighborhood solution is better than the current solution or the probability function ($e^{\Delta/T}$ is great than a random number r), a replacement is made and the *counter_mut* will be set to 0 in Step 3.3; otherwise, the *counter_mut* is increased by 1. The incumbent solution will be updated in Step 3.4 if the newly generated neighborhood solution results in a better objective function value. Step 3 will be repeated L times at each temperature to reach the thermal equilibrium. Parameter T is gradually decreased by a cooling function and the solution process repeats until the stopping criteria in Step 4 is met.

```
HSAM_Algorithm ( )
{
   Read initial solution S^0.
   Let counter_MC = 0, counter_mut = 0, T=T_0, S ← S^0, S* ← S^0.
   WHILE( T > T_f )
   {
      WHILE(counter_MC < L)
      {
            IF(counter_mut ≥ mut_check)
            {
                Apply the mutation operator to generate a new current solution S.
                Let counter_mut = 0.
            }
            Generate a best solution S^N (S^N ∈ N^C) in the neighborhood of S by
            performing the insertion-move operation.
            Compute Δ = f(S^N) − f(S).
            IF ((Δ > 0) or (e^(Δ/T) > r ∈ U(0,1)))
                Let counter_mut = 0, S ← S^N.
            ELSE
                Let counter_mut = counter_mut + 1.
            IF ( f(S^N) > f(S*) )
                Let S* ← S^N.
            Let counter_MC= counter_MC + 1.
      }
      Let T =T×α, counter_MC = 0.
   }
}
```

Figure 4.8 Pseudo code of proposed HSAM procedure

4.1.2.2 TS-based algorithm (HTSM)

Most tabu-based algorithms adopt short-term memory as the primary design for ease of implementation. The solution searching process of short-term TS usually gets trapped in local solutions. Hence, some strategies are developed to guide the search and obtain a limited level of diversified solutions in order to increase the probability of finding the optimal/near-optimal solutions. In this section, the mutation strategy is utilized as a

diversification strategy in the design of our proposed hybrid TS algorithm called HTSM.

(1) Moves

In this study, the insertion-move operation is applied as a mechanism for searching the neighborhood solutions. The neighborhood solutions (N^F) are defined as: $N^F = N^C - N^T + N^A$, where N^C is the set of solutions without violating cell cardinality constraints; N^T is the set of solutions in tabu status; and N^A is the set of solutions satisfying the aspiration criterion.

(2) Tabu list

In the process of tabu search, certain moves are characterized as tabu for some iterations (tabu tenure/tabu list size) to avoid repetition of previously visited solutions. In this paper, a tabu list TL[m][NC][NC] with a three-dimensional array (m×NC×NC) is used to check if a move from a solution to its neighborhood is forbidden or allowed (where m is the number of machines and NC is the number of cells). If machine k moves from its current cell l to a new cell l', then moving machine k from cell l' to cell l will be forbidden for a certain number of iterations, which is equal to the tabu list size (e.g. $\mathrm{TL}[k][l'][l] = tls$).

(3) Aspiration criterion

The tabu restriction may be overridden if the move will result in a solution that is better than the best solution found thus far. This aspiration criterion is applied in the proposed algorithm.

(4) Stopping criterion

The proposed solution procedure will be terminated if a maximum number of iterations $N_{\max}$ have been reached or the solution has not been improved within a certain number of iterations *stag_check*.

The pseudo-code format of the proposed procedure HTSM is diagrammed in Figure 4.9 and is described in detail below:

Algorithm HTSM

Step 1. Read initial solution S^0.

Step 2. Initialization: Let *counter_iter* = 0, *counter_stag* = 0, $S \leftarrow S^0$, $S^* \leftarrow S^0$, $N^T = \varnothing$.

Step 3. If *counter_iter* $\leq N_{\max}$ and *counter_stag* $\leq$ *stag_check*, repeat Steps 4 to 8; otherwise, go to Step 9.

Step 4. If *counter_mut* $\geq$ *mut_check*, then apply the ***mutation strategy*** to generate a new current solution *S* and let *counter_mut* = 0.

Step 5. Generate a best solution $S^N \left(S^N \in N^F\right)$ in the neighborhood of S by performing the ***insertion-move operation***.

Step 6. Update tabu list N^T.

Step 7. If $f(S^N) > f(S^*)$ then $S^* \leftarrow S^N$, *counter_stag* = 0, *counter_mut* = 0; otherwise, *counter_stag* = *counter_stag* + 1, *counter_mut* = *counter_mut* + 1.

Step 8. Let $S \leftarrow S^N$, *counter_iter* = *counter_iter* + 1, go to Step 3.

Step 9. Report the best solutions so far, and stop the algorithm.

Note that the algorithm starts from an initial solution. All parameters and counters are initialized in Step 2. As long as the value of *counter_mut* is smaller than *mut_check*, a new neighborhood solution is generated through the insertion-move in Step 5; otherwise, mutation strategy is applied to generate a new solution with higher degree of diversification in Step 4. If the newly generated neighborhood solution results in a better objective function value, the incumbent solution will be updated, and *counter_stag* and *counter_mut* will be set to 0 in Step 7; otherwise, *counter_stag* and *counter_mut* are increased by 1. The solution process repeats until any of the two stopping criteria in Step 3 is met.

```
HTSM_Algorithm ( )
{
   Read initial solution  S^0 .
   Let counter_iter = 0, counter_stag = 0, S ← S^0 ,  S* ← S^0 ,  N^T = ∅ .
   WHILE(counter_iter ≤ N_max and counter_stag ≤ stag_check)
   {
         IF(counter_mut  ≥  mut_check)
         {
             Apply the mutation strategy to generate a new current solution S.
             Let counter_mut = 0.
         }
         Generate a best solution  S^N (S^N ∈ N^F)  in the neighborhood of  S  by
            performing the insertion-move operation.
         Update tabu list N^T .
         IF ( f(S^N) < f(S*) )
             Let counter_stag = 0, counter_mut = 0, S* ← S^N .
         ELSE
             Let counter_stag = counter_stag + 1, counter_mut = counter_mut + 1.
         Let  S ← S^N , counter_iter = counter_iter + 1.
    }
}
```

Figure 4.9 Pseudo code of proposed HTSM procedure

4.1.2.3 WFA-based algorithm (HWFAM)

As mentioned in Section 2.7, the main operations of WFA include (1) flow splitting and moving, (2) flow merging, (3) water evaporation, and (4) precipitation. We made several changes based on our trials, experiences, and observations.

First, the splitting and moving operation is endowed with the mission of searching for better neighborhood solutions and ultimately the optimal/near-optimal solution. We hence applied two mechanisms, the insertion-move and the mutation strategy, to find the best neighborhood solution of the current solution. The mutation strategy is applied to find a rough direction for the neighborhood solutions in the first stage promptly; the exact location for the best neighborhood solution is then obtained through the “insertion-move” strategy in

the second stage. Figure 4.10 demonstrates the splitting and moving operation for searching neighborhood solutions. As flow i splits into subflows, the number of subflows n_i is determined by its momentum, e.g. n_i equals k. The machine mutation strategy is implemented to determine the rough directions for k subflows; that is, the locations of X_{i1}, X_{i2}, …, X_{ik} can be identified. The insertion-move is then performed to find the best neighborhood solution around X_{i1}; that is, the X^*_{i1}. This is repeated until the best neighborhood solution for each of the subflows has been found. For each iteration, these newly generated subflows may merge with others sharing the same location, proceed in a single stream, split further into more subflows at later iterations, or stagnate in the current location until the stopping criteria of the algorithm is met.

Second, the mass of the subflows is determined based solely on their ranks (Section 2.7.1) without considering their respective performances in the original WFA. Subflows with better objective values should possess greater masses and should persist longer in the water-flowing process. Based on this concept, a new formula for assigning mass to each subflow is designed (Eq. (4.2)).

$$w_{ik} = \left(\frac{f(X_{ik})}{\sum_{k=1}^{n_i} f(X_{ik})} \right) W_i, \tag{4.2}$$

where $f(X_{ik})$ is the objective value of solution X_{ik}.

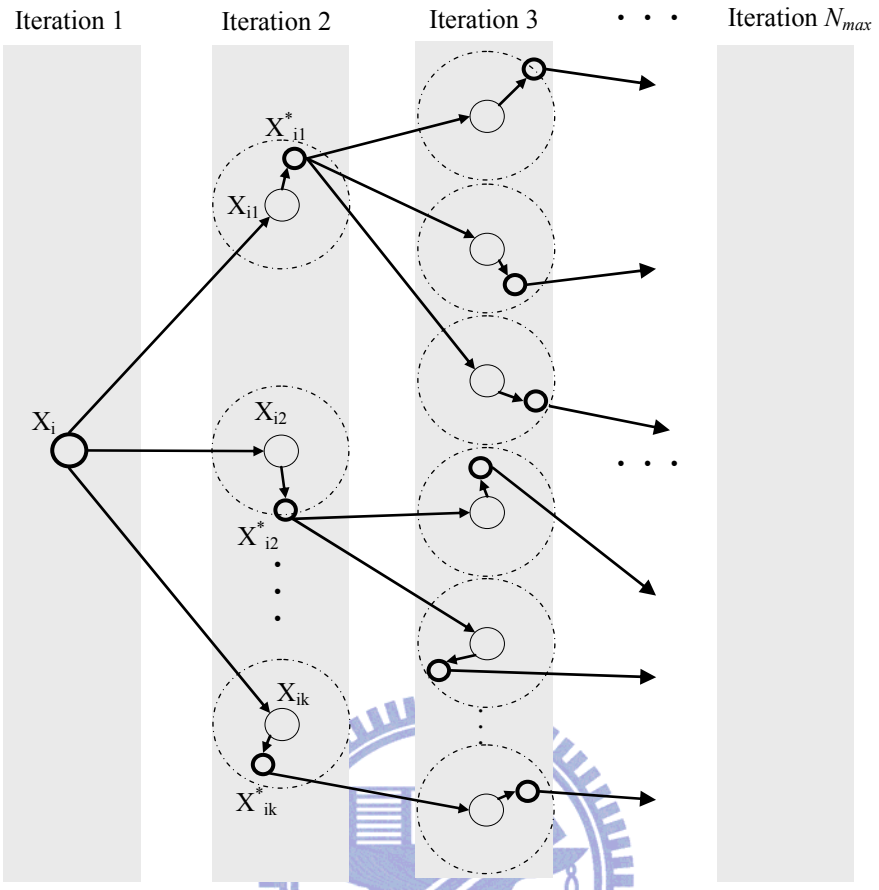


Figure 4.10 Proposed flow splitting and moving operation for searching neighborhood solutions

Third, in addition to the fixed-ratio evaporation presented in the original WFA, another way of evaporation—velocity-based evaporation—is presented and added to the procedure. Eq. (2.10) shows that the higher the altitude drop (i.e., the larger improvement in objective value) of a subflow, the larger the velocity it will be subjected to. We define an evaporation ratio that is conversely related to improvement in velocity, such that flows with smaller velocities should evaporate more quickly than those with larger velocities. The formula is presented below:

$$W_i = (1-\rho_i)W_i\,, \tag{4.3}$$

$$\text{where } \rho_i = \begin{cases} 1, & if\ \mu_{ik}=0 \\ 0, & if\ \dfrac{\mu_{ik}}{V_i} \geq 1 \\ 1-\dfrac{\mu_{ik}}{V_i}, & if\ 0<\dfrac{\mu_{ik}}{V_i}<1 \end{cases}$$

Finally, regular precipitation is performed based solely on a fix iteration (t) to pour down the evaporated water without considering the mass of the evaporated water flow in the original WFA. As mentioned in Section 2.7.3, water evaporation and precipitation are used to avoid being trapped and to explore more solution spaces. Hence, when water vapor accumulates to a certain volume (i.e., solution is trapped), it should return to the ground through precipitation (i.e., escaping from local optima). Based on this concept, another precipitation, the "moist precipitation" is added to the procedure. Moist precipitation is used when the mass of the evaporated water flow reaches half of its original total mass, W_0.

The proposed WFA procedure, namely HWFAM, is presented in pseudo-code format in Figure 4.11 and is described in detail below.

Algorithm HWFAM

Step 1. Read initial solution.

Step 2. Initial HWFAM parameter settings: N_{max}, N, W_0, V_0, T_m.

Step 3. If *counter_iter* $\leq N_{max}$, repeat Steps 4 to 16; otherwise, go to Step 17.

Step 4. For each flow, execute Steps 5 to 16.

Step 5. Calculate the number of subflows based on Eq. (2.8).

Step 6. Flow splitting and moving through ***mutation strategy*** and ***insertion-move operation***.

Step 7. Check whether the new best solution is found. If yes, update best solution.

Step 8. Calculate mass and velocity based on Eqs. (4.2) and (2.10).

Step 9. Merge flows with the same objective values and update the resulting mass and velocity based on Eqs. (2.11) and (2.12).

Step 10. Update the total number of water flow: $N \leftarrow \sum_{i=1}^{N} n_i$.

Step 11. Perform evaporation operation and update the resulting mass for each water flow based on Eq. (4.3).

Step 12. Check whether precipitation condition is met. If yes, perform Steps 13, 14, and 15; otherwise, go to Step 16.

Step 13. Perform ***mutation strategy*** to the current best solution to generate new solutions deviated from the current ones.

Step 14. Distribute mass to flows poured based on Eq. (2.14) or (2.15) depending on the type of precipitation.

Step 15. Check whether the new solution has the same objective value. If yes, merge it and update the resulting mass and velocity based on Eqs. (2.11) and (2.12), then update the total number of water flow *N*.

Step 16. Let *counter_iter* = *counter_iter* + 1, go to Step 3.

Step 17. Report the best solutions so far, and stop the algorithm.

Note that in the mutation strategy, a threshold probability value set at 0.8 implies that each machine has a 20% probability of being assigned to other cells. In the HWFAM procedure, the mutation strategy is used in Steps 6 and 13 with different threshold probability values (β): 0.8 in Step 6 and 0.5 in Step 13. The main purpose of Step 6 is to find some neighborhoods of the current solution, thus the probability of being assigned to other cells is set at a comparatively low value. On the other hand, the purpose of Step 13 is to explore solutions of unvisited regions through the precipitation operation; thus, it becomes necessary to increase the probability of being assigned to other cells to find solutions more deviated from the current best.

```
HWFAM_Algorithm ( )
{
    Read initial solution.
    Initial HWFAM parameter settings.
    Let counter_iter = 0, N=1.
    WHILE (counter_iter < N_max)
    {
       FOR each flow
       {
            Calculate the number of subflows based on Eq. (2.8).
            Flow splitting and moving through the mutation strategy and insertion-move operation.
            IF the new best solution found.
            THEN Update best Solution.
            Calculate the mass and velocity based on Eqs. (4.2) and (2.10).
            IF flows have the same solutions
            THEN run flow merging operation and update W_i and V_i using equations (2.11) and (2.12).
            Update the total number of flow N.
            Run water evaporation and update the mass of flow W_i by equation (4.3).
            IF precipitation condition is met
            {
                 Perform mutation strategy to generate new solutions.
                 Calculate the masses of the pour-downed flows W_i' using equation (2.15) and let V_i'= V_0.
                 IF flows have the same solutions
                 THEN run flow merging operation and update W_i and V_i using equations (2.11) and (2.12).
                 Update the total number of flow N.
            }
       }
       Let counter_iter = counter_iter + 1.
    }
}
```

Figure 4.11 Pseudo code of proposed HWFAM procedure

4.2 Proposed Algorithms for Generalized CFP

In this section, a fast and effective two-stage HGCFA merging a generalized SCM-based clustering algorithm and SA/TS/WFA method is proposed to solve generalized CFP. The framework of the proposed HGCFA is illustrated in Figure 4.12.

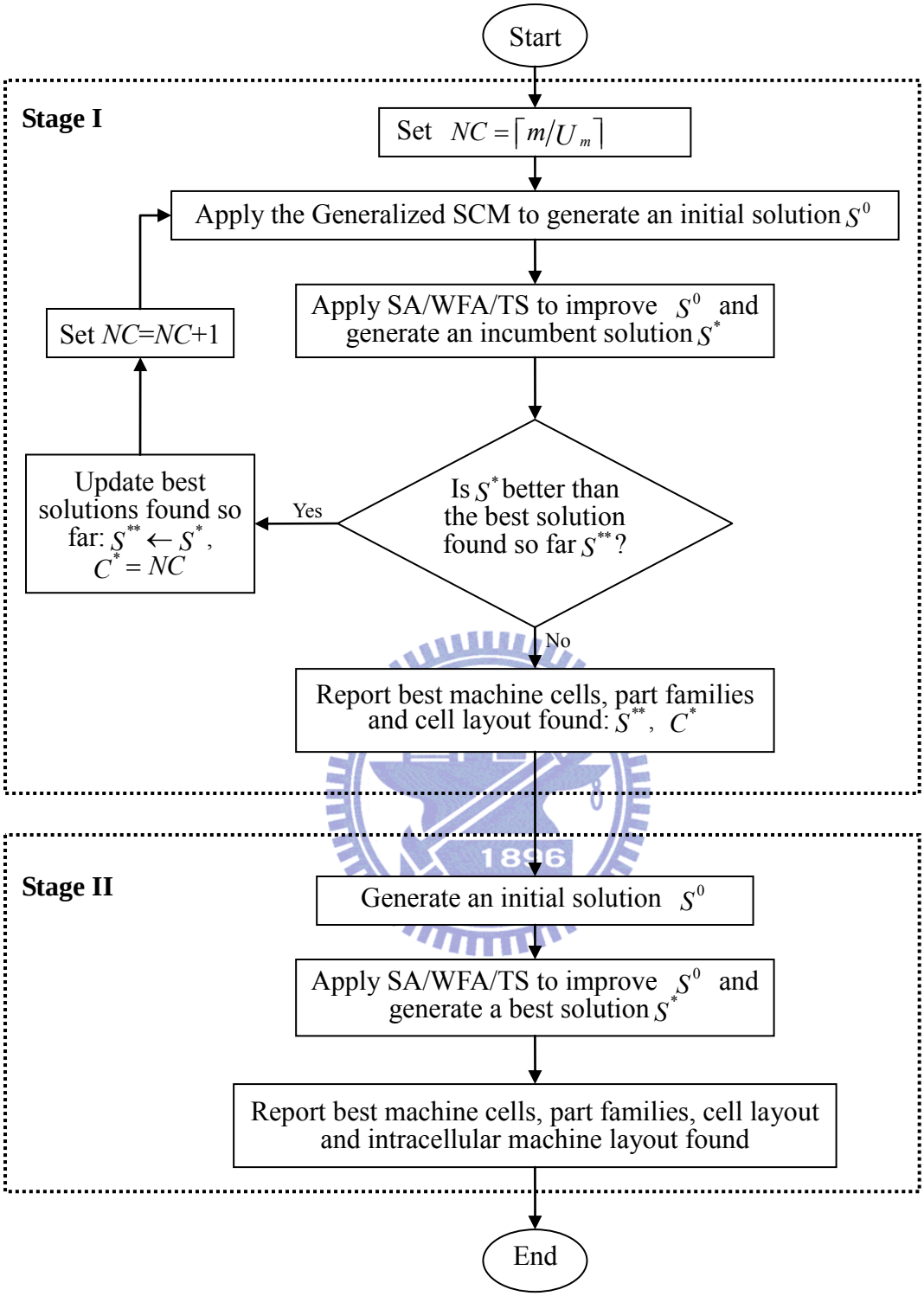


Figure 4.12 Framework of the proposed hybrid generalized CF algorithm (HGCFA)

The first stage mainly solves the CF and inter-cell layout (Inter CL) problem simultaneously in terms of minimizing the sum of total inter-cell move cost (ICMC) and

MBC. In the second stage, the final solution obtained from the first stage is used to construct an initial solution to be improved by the proposed algorithms to determine intra-cell layout (intra CL) in terms of maximizing the CFFI.

The detailed procedures of both stages are described below.

Stage I of HGCFA:

Step 1. Set $NC = \lceil m/U_m \rceil$.

Step 2. Apply the generalized SCM-based clustering algorithm, as mentioned in Section 4.4.1, to generate an initial solution S^0.

Step 3. Let $S^{**} \leftarrow S^0$.

Step 4. Apply SA/TS/WFA procedure, as mentioned in Sections 4.1.2, to improve S^0 and generate an incumbent solution S^*.

Step 5. If $f(S^*) < f(S^{**})$, then set $S^{**} \leftarrow S^*$, $C^* = NC$, $NC = NC+1$, go to Step 2; otherwise, report the best cell formation and inter-cell layout found, and terminate stage I.

Note that the algorithm in this stage consists of an initial solution and an improvement procedure that will be repeatedly applied until a cell formation resulting in the minimum of the total inter-cell move cost (ICMC) and the machine breakdown cost (MBC) have been found. In Step 1, the initial number of cells, NC, can be easily approximated by the nearest integer that is greater than m/U_m; it gradually increases by increments of 1 as long as solution improvement is observed in Step 5. Every time the number of cells is increased, another initial solutions and SA/TS/WFA improvement procedure will be begun in Steps 2 and 4, respectively. For a specific cell size, the best routing selection and grouping plan for parts and machines will be calculated iteratively and obtained in Step 4. Initial solutions of machine cells, routing selections, and part families are generated in Step 2. If larger cell sizes are considered, it is possible that better solutions may be obtained. The incumbent

solution (S^*) of the current cell size (NC) is thus compared with the best cell formation solution (S^{**}) found thus far in Step 5 to determine whether to increase the cell size by 1 and restart another SA/TS/WFA procedure to continue the search or to report the best cell formation solution found and terminate the solution.

Determining the proper number of cells is a difficult decision in the cell formation stage because the layout designer does not have any knowledge regarding the cell size at the beginning. Unlike most of the study in the literature where the number of cells to be formed is prescribed beforehand, the number of cells resulting in the least total cost is automatically calculated and used in the proposed approach. However, to preserve flexibility, users are allowed to specify the preferred number of cells when implementing the algorithm. For users having specific preferences in cell size, the proposed algorithm can save considerable amount of run time because it will skip the process of iteratively searching for the cell size that will result in the best objective function values. The savings in run time become even more significant as the cell size increases.

Stage II of HGCFA:

Step 1. Read solutions from stage I, including number of cells, C^* and cell formation with inter-cell layout S^{**}.

Step 2. Apply the initial solution construction, as mentioned in Section 4.4.2, to generate an initial solution S^0.

Step 3. Apply SA/TS/WFA procedure, as mentioned in Section 4.1.2, to improve S^0 and generate a best layout of machines within each cell (S^*).

Note that the final solutions (C^* and S^{**}) obtained from the first stage will be read in Step 1 and will be used to construct an initial solutions of machines sequence configuration (S^0) in Step 2. In Step 3, the initial solution (S^0) will be improved through SA/TS/WFA procedure, as mentioned in Section 4.1.2, to generate a best solution (S^*) in terms of maximizing the CFFI.

4.2.1 Proposed algorithms for stage I

Proposed algorithms for stage one consist of two procedures: the initial solution construction (i.e., generalized SCM-based clustering algorithm) and the solution improvement (i.e., SA, TS, and WFA). The SA/TS/WFA procedure is as mentioned in Section 4.1.2, while the initial solution construction will be described later.

The initial solution is generated through generalized SCM-based clustering algorithm. It is composed of four parts: (1) determination of layout type, (2) formation of machine cells, (3) selection of routings for each part, and (4) formation of part families. The details of these procedures are given below.

(1) Determination of layout type

As mentioned in Section 3.3, two basic cellular layout types (e.g., max number of row r equal to 1 or 2) are considered in this study. When the layout designer has chosen a specific cellular layout type, this type is assumed and used in the subsequent design.

(2) Formation of machine cells

According to Seifoddini and Djassemi (1995), incorporation of production volume into the similarity measures may increase the probability of components with high production volumes being processed within a single cell. As a result, there will be fewer intercellular movements and lower material handling costs. The generalized SCM of Won and Kim (1997) is modified to incorporate product volume information. Taking into account a specific machine-part incidence matrix and product volume information, the corresponding similarity matrix for machines can be obtained using the following formula:

$$S_{ij} = \frac{N_{ij}}{N_i + N_j - N_{ij}}, \tag{4.4}$$

where

S_{ij} = similarity coefficient between machines i and j

$$N_i = \sum_{k=1}^{p} V_k \, a_i^k, \; N_j = \sum_{k=1}^{p} V_k \, a_j^k, \; N_{ij} = \sum_{k=1}^{p} V_k \, a_{ij}^k$$

p = number of parts

V_k = production volume of part k

$$a_i^k = \begin{cases} 1 & \text{if } i \in \text{ some routing of part } k \\ 0 & \text{otherwise} \end{cases}$$

$$a_j^k = \begin{cases} 1 & \text{if } j \in \text{ some routing of part } k \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij}^k = \begin{cases} 1 & \text{if } i, j \in \text{ the same routing of part } k \text{ synchronously} \\ 0 & \text{otherwise} \end{cases}$$

After calculating the similarity matrix for each pair of machines, the initial machine assignment is generated using the single linkage clustering (SLC) algorithm. The SLC algorithm works as follows:

Step 1. Join the two most similar objects (two machines, a machine and a machine group, or two machine groups) to form a new machine group.

Step 2. Evaluate the similarity coefficient between the new machine group and other remaining machine/machine groups as follows: $S_{tv} = \text{Max}\{S_{ij}\} \;\; i \in t \;\; j \in v$, where i is the machine in the machine group t and j is the machine in the machine group v.

Step 3. Repeat Steps 1 to 2 until a predetermined number of machine groups has been obtained.

(3) Selection of routings for each part

After the formation of machine cells have been obtained, the routing for each part can be determined by the procedure detailed below.

Step 1. Read the results of the machine cells formed by the machine-based similarity matrix.

Step 2. For each part with alternative routings, find the routing that will result in the least sum of objective value. If a tie occurs, make a random selection.

Step 3. Repeat Step 2 until the process routing has been determined for each part.

(4) Formation of part families

Part families are formed after the formation of machine cells and determination of the routing for each part. The procedure is summarized as follows:

Step 1. Read the results of machine assignment and routing selection for each part.

Step 2. For each part, find the cell to which a part assignment will result in the least sum of exceptional elements and voids. If a tie occurs, assign the part to a cell with the least number of voids.

Step 3. Repeat Step *2* until all parts have been assigned to cells.

4.2.2 Proposal algorithms for stage II

This stage consists of two procedures: initial solution construction and solution improvement (i.e., SA, TS, and WFA). The SA/TS/WFA procedure is the same as discussed in Section 4.1.2. Initial solution construction and some elements comprising the proposed algorithms are described below.

(1) Initial solution construction

The initial solution of the sequence of machines in each cell can be generated by the following procedure:

Step 1. Read the machine cells determined in stage one.

Step 2. Arrange machine cells by cell number in an ascending order.

Step 3. Arrange the sequences of machines in each cell in an ascending order.

(2) Configuration

A three-dimensional array is used to represent the configuration of a feasible solution of the sequences of machines within each cell. Figure 4.13 shows an example where machine #3 was assigned to the first sequence of cell #1.

Cell #	1			2			3			
Sequence #	1	2	3	1	2	3	1	2	3	4
Machine #	3	7	8	2	4	6	1	10	9	5

Figure 4.13 Configuration of an initial solution to sequence of machines

(3) Neighborhood solution searching

In this stage, the neighborhood of a given solution is defined as the set of all feasible solutions reachable by an exchange-move. The exchange-move is an operation that exchanges any pair of machines within the same cell. If we exchange machine k with machine k', then the new move is denoted as (k, k'). The move that results in the most improvement in CFFI value from the current solution is selected; that is,

$$Z(k,\ k') = Max\{obj^{(k,k')} - obj^{(k',k)}, \forall k, k' \in M \text{ and } \in N^F \text{ and } k \neq k'\} \quad (4.5)$$

where $obj^{(k,k')}$ is the objective function value; M is the set for machines; and N^F is the set of feasible solutions.

(4) Mutation strategy

When the number of moves has not been improved within a certain number of iterations, the mutation strategy (*mut_check*) is implemented by exchanging any pair of machines within the same cell based on a prescribed probability β. For each machine in the same cell, a random number from (0, 1) is first drawn. If the value is greater than β, then the machine sequence is exchanged with another randomly determined machines sequence within the same cell; otherwise, it remains in the current sequence. The procedure of machine sequence mutation strategy is presented in pseudo-code format in Figure 4.14.

```
Mutation_strategy ( β )
{
    Let the current solution (S) equal to the best solution (S*).
     FOR each machine in the same cell DO
    {
        Generate a random number  r ∈ U(0,1) .
        IF ( r > β )
            Exchange machine sequence with the other machines.
        ELSE
            Stay machine in the current sequence.
    }
}
```

Figure 4.14 Pseudo code of mutation strategy

(5) Tabu list

In the TS procedure, a two-dimensional array ($m \times m$) TL[m][m] ,where m is the number of machines, is used as a tabu list to check if a move from a solution to its neighborhood is forbidden or allowed. If a pair of machines k and k' are exchanged, then the exchanging of machine k' and k will be forbidden for a certain number of iterations, which is equal to the tabu list size *tls* (e.g., TL[k'][k] = *tls*).

CHAPTER 5

NUMERICAL ILLUSTRATIONS

As mentioned in Chapter 4, two hybrid meta-heuristic algorithms integrating SCM-based clustering algorithm and SA/TS/WFA are proposed to solve standard CFP and generalized CFP, respectively. To illustrate the effectiveness of our developed algorithms, two test examples are demonstrated in this chapter. Example #1 includes 10 machines and 10 parts that comprise a simple CFP with a 0-1 machine-part incidence matrix. Example #2 consists of 10 machines and 10 parts that form a generalized CFP. The proposed algorithms were coded in C++ using Microsoft Visual Studio 6.0 and implemented on an Intel(R) 1.66GHz PC with 1GB RAM. Computation results for both types of CFP are shown and discussed separately in this chapter.

5.1 An Illustrative Example for Standard CFP

The 0-1 machine-part incidence matrix for example #1 is given in Figure 5.1. The minimum number of machines in each cell (L_m) is limited to 2 (i.e., singletons are not allowed). The objective function aims to determine machine cells and part families in which grouping efficacy can be maximized. The implementation of the proposed method for standard CFP is described as follows:

M\P	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
M1	0	1	0	0	1	0	0	1	0	0
M2	1	0	1	0	0	0	1	0	0	0
M3	0	0	0	1	0	1	0	0	1	1
M4	1	0	0	0	0	0	1	0	0	0
M5	0	1	0	0	1	0	0	1	1	0
M6	1	0	0	0	0	0	1	0	0	0
M7	0	0	1	1	0	1	0	0	1	1
M8	0	0	1	1	0	1	0	0	1	1
M9	0	1	0	0	0	0	0	1	0	0
M10	0	1	0	0	1	0	0	0	0	0

Figure 5.1 0-1 machine-part matrix of example #1

Stage I of HCFA:

Step 1. Set $NC = 2$, $f(S^0) = f(S^*) = 0$.

Step 2. Apply the SCM-based clustering algorithm to generate an initial solution S^0.

As mentioned in Section 4.1.1, the SCM-based clustering algorithm consists of three steps: (1) computation of similarity values between machine pairs and construction of a similarity matrix, (2) utilization of a clustering rule to process the values in the similarity matrix and formation of machine cells, and (3) assignment of parts to machine cells using a parts assignment procedure. They are described as follows:

(1) Calculation of machine similarity matrix

The corresponding similarity matrix for machines can be obtained by using Eq. (2.1) and is shown in Table 5.1.

Table 5.1 Similarity matrix for machines in example #1

Machine	1	2	3	4	5	6	7	8	9	10
1	-									
2	0	-								
3	0	0	-							
4	0	0.67	0	-						
5	0.75	0	0.14	0	-					
6	0	0.67	0	1.00	0	-				
7	0	0.14	0.80	0	0.13	0	-			
8	0	0.14	0.80	0	0.13	0	1.00	-		
9	0.67	0	0	0	0.50	0	0	0	-	
10	0.67	0	0	0	0.50	0	0	0	0.33	-

(2) Formation of machine cells

The similarity matrix shows that the largest coefficient in the matrix is 1, appearing in pairs (7, 8) and (4, 6). Since pairs (7, 8) and (4, 6) do not have any relationship, they are assigned to cell #1 and cell #2, respectively. The second largest coefficient in the matrix (0.8) appears in pairs (3, 7); because machine 7 has been assigned to cell #1, machine 3 is assigned to the same cell. Next in line is pair (1, 5); machines 1 and 5 have not been

assigned to any cell so they should be assigned to cell #3. However, the initial NC is 2, so that machines 1 and 5 are assigned to cell #2. Using the same logic, we can assign machines 2, 9, and 10 to cell #2. Thus, machines 3, 7, and 8 are assigned to cell #1, while machines 1, 2, 4, 5, 6, 9, and 10 are assigned to cell #2, as shown in Figure 5.2.

Cell No.	M\P	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	M3	0	0	0	1	0	1	0	0	1	1
	M7	0	0	1	1	0	1	0	0	1	1
	M8	0	0	1	1	0	1	0	0	1	1
2	M1	0	1	0	0	1	0	0	1	0	0
	M2	1	0	1	0	0	0	1	0	0	0
	M4	1	0	0	0	0	0	1	0	0	0
	M5	0	1	0	0	1	0	0	1	1	0
	M6	1	0	0	0	0	0	1	0	0	0
	M9	0	1	0	0	0	0	0	1	0	0
	M10	0	1	0	0	1	0	0	0	0	0

Figure 5.2 Assignment of machines

(3) Formation of part families

After calculating the sum of voids and exceptional elements for each part-cell combination (Figure 5.3), it became apparent that parts 3, 4, 6, 9, and 10 should be assigned to cell #1 and that parts 1, 2, 5, 7, and 8 should be assigned to cell #2 because this arrangement results in the least sum of voids and exceptional elements. Thus, the initial machine-part incidence matrix has been generated with a total grouping efficacy (Γ) of 57.69%, as shown in Figure 5.4.

Cell No.	M\P	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	M3	0	0	0	1	0	1	0	0	1	1
	M7	0	0	1	1	0	1	0	0	1	1
	M8	0	0	1	1	0	1	0	0	1	1
2	M1	0	1	0	0	1	0	0	1	0	0
	M2	1	0	1	0	0	0	1	0	0	0
	M4	1	0	0	0	0	0	1	0	0	0
	M5	0	1	0	0	1	0	0	1	1	0
	M6	1	0	0	0	0	0	1	0	0	0
	M9	0	1	0	0	0	0	0	1	0	0
	M10	0	1	0	0	1	0	0	0	0	0
	Sum of voids and exceptional elements										
	Cell #1	6	7	2	0	6	0	6	6	1	0
	Cell #2	4	3	8	10	4	10	4	4	9	10

Figure 5.3 Assignment of parts

Cell No.	M\P	P3	P4	P6	P9	P10	P1	P2	P5	P7	P8
1	M3	0	1	1	1	1	0	0	0	0	0
	M7	1	1	1	1	1	0	0	0	0	0
	M8	1	1	1	1	1	0	0	0	0	0
2	M1	0	0	0	0	0	0	1	1	0	1
	M2	1	0	0	0	0	1	0	0	1	0
	M4	0	0	0	0	0	1	0	0	1	0
	M5	0	0	0	1	0	0	1	1	0	1
	M6	0	0	0	0	0	1	0	0	1	0
	M9	0	0	0	0	0	0	1	0	0	1
	M10	0	0	0	0	0	0	1	1	0	0
Grouping efficacy (Γ) =57.69%											

Figure 5.4 Solution configuration for *NC*=2

Step 3. Since $0.5769 > 0$, then set $S^* \leftarrow S^0$, Let $C^* = 2$, $NC = 2+1$. Repeat Steps 2 and 3 until $f(S^0) < f(S^*)$.

The relationship between the *NC* and the resulting grouping efficacy ($f(S^0)$) is shown in Figure 4.1. It is observed that $f(S^0)$ increases as *NC* increases, and the optimal/near-optimal is achieved when *NC*=3. After that, efficacy starts to decrease as *NC*

increases. The suggested number of cells (C^*=3) and the cell configurations (S^*) are shown in Figure 5.6.

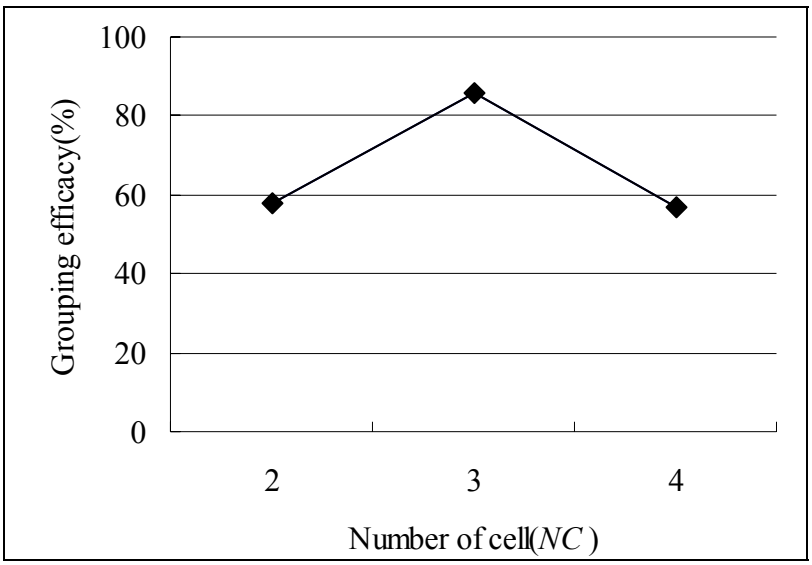


Figure 5.5 Relationship between grouping efficacy and number of cells for example #1

Cell No.	M\P	P1	P7	P3	P4	P6	P9	P10	P2	P5	P8
1	M2	1	1	1	0	0	0	0	0	0	0
	M4	1	1	0	0	0	0	0	0	0	0
	M6	1	1	0	0	0	0	0	0	0	0
2	M3	0	0	0	1	1	1	1	0	0	0
	M7	0	0	1	1	1	1	1	0	0	0
	M8	0	0	1	1	1	1	1	0	0	0
3	M5	0	0	0	0	0	1	0	1	1	1
	M9	0	0	0	0	0	0	0	1	0	1
	M10	0	0	0	0	0	0	0	1	1	0
	M1	0	0	0	0	0	0	0	1	1	1
	Grouping efficacy (Γ)=85.71%										

Figure 5.6 Solution configuration for *NC*=3

Stage II of HCFA:

Step 1. Read solutions from stage one, including number of cells (C^*) and cell configurations (S^*).

Step 2. Let $S^0 \leftarrow S^*$, $NC=C^*$.

Step 3. Apply HSAM algorithm (as mentioned in Section 4.1.2.1) to improve S^0 (Figure 5.6). Using C^*=3, a total grouping efficacy (Γ) of 85.71% can be obtained. Note that the solution is the same as the initial solution found in stage I. This implies that our proposed SCM-based clustering algorithm can produce a good initial solution.

To evaluate the performance of our proposed HCFA, the mathematical model described in Section 3.2.2 is solved using Lingo 8.0 software. The Lingo solver status for example #1 is shown in Figure 5.7. It took about 32 seconds to obtain the optimal solution (0.857143). In contrast, our proposed HCFA was able to find the optimal solution in 0.14 seconds, thus illustrating the superiority of HCFA in solution efficiency. We believe this superiority will be even more significant as the problem size increases.

Figure 5.7 Lingo solver status for example #1

5.2 An Illustrative Example for Generalized CFP

Example #2 consists of 10 machines, 10 parts, and 18 process routings. The production data (production volumes, alternative process routings, and processing times) are

summarized in Table 5.2. In this table, entry (j, k) means the order of machine k in part route j, while the number in parentheses refers to production time. For instance, the order of machine #1 (M1) in routing #1 (R1) for part #1 (P1) is 1. The production time of machine #1 (M1) in routing #1(R1) for part #1 (P1) is 2 minutes. Machine reliability information is shown in Table 5.3. For instance, the MTBF for machine #1 is 2241 minutes and breakdown cost is 1300. The maximum number of machines in each cell (U_m) is limited to 4 and the minimum number of machines in each cell (L_m) is 2. Intercellular movement unit cost is assumed to be 5. The linear single-row layout is chosen. Implementation of the proposed HGCFA for cell formation, cell layout, and intracellular machine layout simultaneously with considerations of alternative routing, production volume, and machine reliability is described as follows:

Table 5.2 Initial machine-part matrix of example #2

PN	P1		P2		P3	P4		P5	P6		P7		P8		P9		P10	
PV	150		95		130	80		120	95		135		145		100		150	
RN	R1	R2	R1	R2	R1	R1	R2	R1	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
M1	*1(2)			1(6)				1(2)	1(4)			1(5)	1(3)	1(3)			1(5)	
M2	2(5)	2(2)			1(2)				2(6)		1(6)	2(6)						
M3						1(6)	1(6)		3(3)	1(3)					1(6)	1(4)	2(2)	1(5)
M4	3(6)	1(4)									2(4)	3(4)						
M5			3(5)	4(6)				3(5)					5(2)	3(3)	4(5)			
M6		3(3)									3(4)							
M7			2(4)		3(3)		3(5)			3(6)					3(6)		3(4)	3(5)
M8			1(6)		2(4)	2(3)	2(5)			2(2)			3(4)		2(3)	2(6)		2(6)
M9				3(2)		3(3)							4(5)	2(5)		3(4)		
M10				2(6)				2(2)					2(5)					

PV: Production Volume; PN: Part Number; RN: Routing Number; * Process Sequence (Production Times)

Table 5.3 Machine reliability information for example #2

Machine	Breakdown cost	MTBF(min)
1	1300	2241
2	1200	3403
3	1000	2133
4	1600	3600
5	1400	2271
6	1000	2159
7	1200	3374
8	1400	4380
9	1500	1316
10	1300	854

Stage I of HGCFA:

(1) Determination of the initial number of cells and selection of cell layout

The initial number of cells is calculated: $NC = \lceil 10/4 \rceil = 3$. Then, cells are arranged in a linear single-row cellular layout, as shown in Figure 5.8.

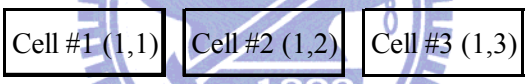


Figure 5.8 Initial linear single-row layout

(2) Calculation of machine similarity matrix

The corresponding similarity matrix for machines can be obtained by using Eq. (4.4) and is presented in Table 5.4.

Table 5.4 Similarity matrix for machines in example #2

Machine	1	2	3	4	5	6	7	8	9	10
1	-									
2	0.37	-								
3	0.23	0.11	-							
4	0.32	0.56	0.00	-						
5	0.36	0.00	0.13	0.00	-					
6	0.00	0.56	0.00	1.00	0.00	-				
7	0.11	0.13	0.65	0.00	0.21	0.00	-			
8	0.09	0.11	0.54	0.00	0.37	0.00	0.82	-		
9	0.22	0.00	0.27	0.00	0.38	0.00	0.00	0.37	-	
10	0.40	0.00	0.00	0.00	0.78	0.00	0.00	0.14	0.44	-

(3) Assignment of machines to cells

Three cells are to be formed initially. The largest coefficient in the similarity matrix is 1, indicating that machines 4 and 6 must be assigned to cell #1. The second largest coefficient in the matrix (0.82) appears in pair (7, 8). Since machines 7 and 8 have not been assigned to any cell, they are assigned to cell #2. Pair (5, 10) is considered next, and since machines 5 and 10 have not been assigned to any cell, they are assigned to cell #3. The next choice is pair (3, 7); machine 3 is assigned to cell #2 as well because machine 7 has been assigned there. Next is pair (2, 4); machine 2 is also assigned to cell #1 because machine 4 has been assigned there. Applying the same logic, we can determine that machine 1 should be assigned to cell #3, while machine 9 should be assigned to cell #3. Thus, machines 2, 4, and 6 are assigned to cell #1; machines 3, 7, and 8 are assigned to cell #2; and machines 1, 5, 9, and 10 are assigned to cell #3, as shown in Table 4.

Table 5.5 Formation of machine cells for numerical example #2

Cell No.	PN	P1		P2		P3	P4		P5	P6		P7		P8		P9		P10	
	PV	150		95		130	80		120	95		135		145		100		150	
	RN	R1	R2	R1	R2	R1	R1	R2	R1	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
1	M2	2(5)	2(2)			1(2)				2(6)		1(6)	2(6)						
	M4	3(6)	1(4)									2(4)	3(4)						
	M6		3(3)									3(4)							
2	M3						1(6)	1(6)		3(3)	1(3)					1(6)	1(4)	2(2)	1(5)
	M7			2(4)		3(3)		3(5)			3(6)					3(6)		3(4)	3(5)
	M8			1(6)		2(4)	2(3)	2(5)			2(2)			3(4)		2(3)	2(6)		2(6)
3	M1	1(2)			1(6)				1(2)	1(4)			1(5)	1(3)	1(3)			1(5)	
	M5			3(5)	4(6)				3(5)					5(2)	3(3)	4(5)			
	M9				3(2)		3(3)							4(5)	2(5)		3(4)		
	M10				2(6)				2(2)					2(5)					

(4) Selection of routings for each part

Routing 2 is selected by parts 1, 4, 6, 8, 9 and 10; while routing 1 is selected by parts 2, 3, 5 and 7, because these selections will result in the least total ICMC and MBC. Thus, Table 5.5 is rearranged as Table 5.6.

Table 5.6 Part routing assignment for numerical example #2

Cell No.	PN	P1		P2		P3	P4		P5	P6		P7		P8		P9		P10	
	PV	150		95		130	80		120	95		135		145		100		150	
	RN	R1	R2	R1	R2	R1	R1	R2	R1	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
1	M2	2(5)	2(2)			1(2)				2(6)		1(6)	2(6)						
	M4	3(6)	1(4)									2(4)	3(4)						
	M6		3(3)									3(4)							
2	M3						1(6)	1(6)		3(3)	1(3)					1(6)	1(4)	2(2)	1(5)
	M7			2(4)		3(3)		3(5)			3(6)					3(6)		3(4)	3(5)
	M8			1(6)		2(4)	2(3)	2(5)			2(2)			3(4)		2(3)	2(6)		2(6)
3	M1	1(2)			1(6)				1(2)	1(4)			1(5)	1(3)	1(3)			1(5)	
	M5			3(5)	4(6)				3(5)					5(2)	3(3)	4(5)			
	M9				3(2)		3(3)							4(5)	2(5)		3(4)		
	M10				2(6)				2(2)					2(5)					
Total inter-cell move cost (ICMC) and the machine breakdown cost (MBC)																			
ICMC		1500	0	475	0	650	400	0	0	1425	0	0	1350	1450	0	500	500	750	0
MBC		839	581	610	1766	397	575	495	874	555	397	776	917	2584	1347	899	835	789	906
TC		2339	581	1085	1766	1047	975	495	874	1980	397	776	2267	4034	1347	1399	1335	1539	906

(5) Formation of part families

After calculating the sum of voids and exceptional elements for each part-cell combination (Figure 5.6), we observed that parts 1 and 7 should be assigned to cell #1, parts 2, 3, 4, 6, 9 and 10 should be assigned to cell #2, and parts 5 and 8 should be assigned to cell #3 because this arrangement results in the least sum of voids and exceptional elements. Thus, the initial machine-part incidence matrix has been generated with total ICMC and MBC of 8843, as shown in Figure 5.9.

Cell No.	PN	P1	P7	P2	P3	P4	P6	P9	P10	P5	P8
	PV	150	135	95	130	80	95	100	150	120	145
	RN	R2	R1	R1	R1	R2	R2	R2	R2	R1	R2
1	M2	2(2)	1(6)		1(2)						
	M4	1(4)	2(4)								
	M6	3(3)	3(4)								
2	M3					1(6)	1(3)	1(4)	1(5)		
	M7			2(4)	3(3)	3(5)	3(6)		3(5)		
	M8			1(6)	2(4)	2(5)	2(2)	2(6)	2(6)		
3	M1									1(2)	1(3)
	M5			3(5)						3(5)	3(3)
	M9							3(4)			2(5)
	M10									2(2)	
Total inter-cell move cost (ICMC) and the machine breakdown cost (MBC)=8843											
ICMC		0	0	475	650	0	0	500	0	0	0
MBC		581	776	610	397	495	397	835	906	874	1347
TC		581	776	1085	1047	495	397	1335	906	874	1347
Sum of voids and exceptional elements											
Cell #1		0	0	6	4	6	6	6	6	6	6
Cell #2		6	6	2	2	0	0	2	0	6	6
Cell #3		7	7	5	7	7	7	5	7	1	1

Cell #1 (1,1)
M2, M4, M6

Cell #2 (1,2)
M3, M7, M8

Cell #3 (1,3)
M1, M5, M9, M10

Figure 5.9 Initial solution of stage I for example #2

(6) Improvement of ICMC through HGCFA algorithm in stage I

The initial solution generated in Figure 5.9 with total ICMC and MBC of 8843 can be improved through the HGCFA algorithm in stage I. After 0.64 seconds CPU time, the final solution with a total ICMC and MBC of 8843 can be obtained. Note that the final solution is similar to the initial solution. This means that our proposed generalized SCM-based clustering algorithm can produce a good initial solution. So far, cells have been formed and cell layout has been determined. Solution regarding the machine layout (sequence) for each cell is left to be determined in the next step.

In order to get the optimal solution, a pure integer liner model described in Section 3.4.2.1 is solved using a branch and bound (B&B) algorithm with the Lingo 8.0 software. The Lingo solver status for example #2 is shown in Figure 5.10. The optimal solution (8843.22) is obtained in 4 seconds. In contrast, our proposed HGCFA was able to find the optimal solution in 1 second, thus implying the superiority of HGCFA in solution efficiency. Similarly, we believe this superiority will be even more significant as problem size increases.

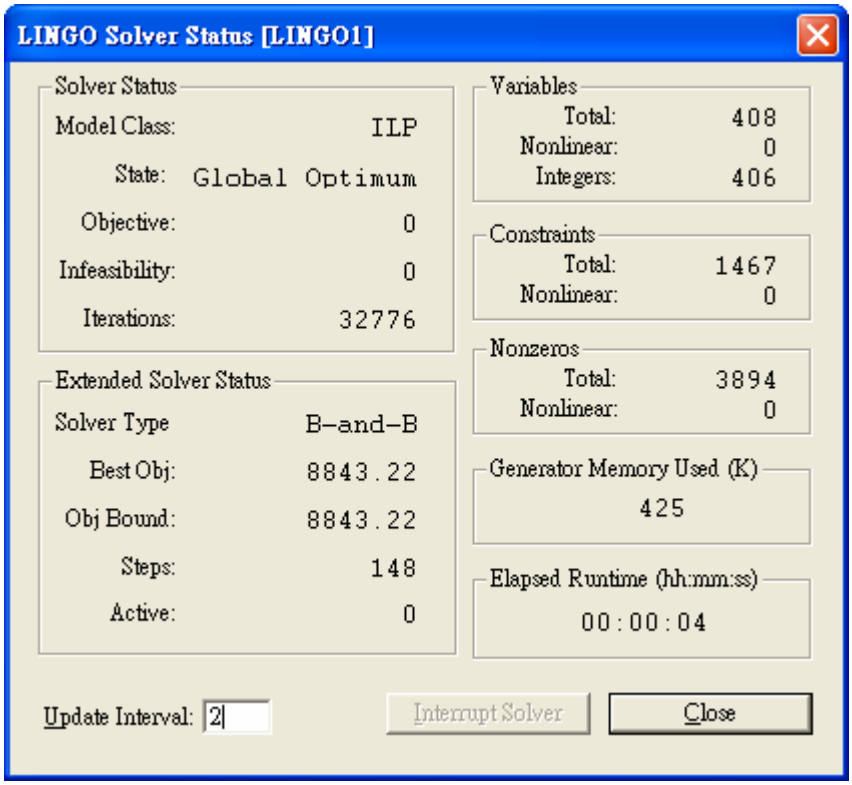


Figure 5.10 Lingo solver status for stage I (cell formation and inter-cell layout)

Stage II of HGCFA:

(1) Generation of initial solution in stage II

The initial solution of the sequence of machines in each cell can be generated by the following procedure:

Step 1. Read the machine cells determined in stage one.

Step 2. Arrange the machine cells by cell number in an ascending order.

Step 3. Arrange the sequences of machines in each cell in an ascending order.

Using the above procedure, Figure 5.11, showing the initial machine sequences within each cell, can be constructed.

Cell #	1				2			3		
Sequence #	1	2	3	4	1	2	3	1	2	3
Machine #	1	5	9	10	3	7	8	2	4	6

Figure 5.11 Initial configuration of machine sequence for example #2

Based on the solution from stage I and the initial configuration of machine sequence, the corresponding flow matrix for machines can be obtained by using Eq. (3.25). This is presented in Table 5.7. Using Eq. (3.26), the initial CFFI can be calculated as follows:

$$\text{CFFI}=\frac{N_{cff}}{N_{tf}}=\frac{270}{2400}=11.25\%.$$

Table 5.7 Flow matrix

Machine	1	2	3	4	5	6	7	8	9	10
1	-	0	0	0	0	0	0	0	145	120
2	0	-	0	135	0	150	0	130	0	0
3	0	0	-	0	0	0	0	425	0	0
4	0	150	0	-	0	135	0	0	0	0
5	0	0	0	0	-	0	0	0	0	0
6	0	0	0	0	0	-	0	0	0	0
7	0	0	0	0	95	0	-	0	0	0
8	0	0	0	0	0	0	550	-	100	0
9	0	0	0	0	145	0	0	0	-	0
10	0	0	0	0	120	0	0	0	0	-

(2) Improvement of CFFI through HGCFA algorithm in stage II

Through the proposed HTSCF in stage II, the CFFI can be improved to 65.21(%) after 0.19 seconds CPU time. The final corresponding configuration for the cell formation, cell layout, and intracellular machine layout is displayed in Figure 5.12.

Cell No.	PN	P1	P7	P2	P3	P4	P6	P9	P10	P5	P8
	PV	150	135	95	130	80	95	100	150	120	145
	RN	R2	R1	R1	R1	R2	R2	R2	R2	R1	R2
1	M4	1(4)	2(4)								
	M2	2(2)	1(6)		1(2)						
	M6	3(3)	3(4)								
2	M3					1(6)	1(3)	1(4)	1(5)		
	M8			1(6)	2(4)	2(5)	2(2)	2(6)	2(6)		
	M7			2(4)	3(3)	3(5)	3(6)		3(5)		
3	M1									1(2)	1(3)
	M9							3(4)			2(5)
	M5			3(5)						3(5)	3(3)
	M10									2(2)	
Total inter-cell move cost (ICMC) and the machine breakdown cost (MBC)=8843											
ICMC		0	0	475	650	0	0	500	0	0	0
MBC		581	776	610	397	495	397	835	906	874	1347
TC		581	776	1085	1047	495	397	1335	906	874	1347
Consecutive forward flow index (CFFI) = 65.21(%)											

Cell #1 (1,1)	Cell #2 (1,2)	Cell #3 (1,3)
M4, M2, M6	M3, M8, M7	M1, M9, M5, M10

Figure 5.12 Final solution of stage II (cell formation, inter-cell layout and intra-cell layout)

To evaluate the performance of our proposed HGCFA, the mathematical model described in Section 3.4.2.2 is solved using Lingo 8.0 software. The Lingo solver status is shown in Figure 5.13. The optimal solution (0.6521) is obtained in 1 second. In contrast, our proposed HGCFA was able to find the optimal solution in 0.2 seconds, thus illustrating the superiority of HGCFA in solution efficiency. We believe this superiority will be even more significant as problem size increases.

Figure 5.13 Lingo solver status for stage II (intra-cell layout)

CHAPTER 6

COMPUTATIONAL RESULTS AND COMPARISONS

Two types of CFP are addressed in this thesis. Standard CFP is represented by a binary machine-part incidence matrix, while the generalized CFP had layout design and machine reliability considerations. In order to solve these problems quickly and effectively, two hybrid algorithms merging an SCM-based clustering algorithm and SA/WFA/TS meta-heuristics are presented in this thesis. The proposed algorithms were coded in C++ using Microsoft Visual Studio 6.0 and implemented on an Intel(R) 1.66 GHz PC with 1 GB RAM. Since the proposed methods might have stochastic features, five independent runs were performed for each test. The computation results for both types of CFP are shown and discussed separately in this chapter.

6.1 Computational Results for Standard CFP

To validate the quality of the solutions provided by the proposed algorithms for standard CFP, 35 test instances, each represented by a binary machine-part incidence matrix, from literature (Table 6.1) are used to evaluate the computational characteristics of our proposed HCFA. The matrices of the test problems range from 5×7 to 40×100, and consist of both well-structured and unstructured matrices. The parameters values for HCFA-HSAM, HCFA-HWFAM, and HCFA-HTSM throughout all runs are described in Table 6.2. Some studies in the literature allowed the existence of singletons in the solutions, while some did not. To make comparisons fair and meaningful, the computational results are shown and discussed separately in two subsections.

Table 6.1 Test instances from the literature for standard CFP

No.	Source	Size ($m \times p$)
1	King and Nakornchai (1982)	5×7
2	Waghodekar and Sahu (1984)	5×7
3	Seifoddini (1989)	5×18
4	Kusiak and Cho (1992)	6×8
5	Kusiak and Chow (1987)	7×11
6	Boctor (1991)	7×11
7	Seifoddini and Wolfe (1986)	8×12
8	Chandrasekharan and Rajagopalan (1986a)	8×20
9	Chandrasekharan and Rajagopalan (1986b)	8×20
10	Mosier and Taube (1985a)	10×10
11	Chan and Milner (1982)	10×15
12	Askin and Subramanian (1987)	14×23
13	Stanfel (1985)	14×24
14	McCormick *et al.*(1972)	14×24
15	Srinivasan *et al.*(1990)	16×30
16	King (1980)	16×43
17	Carrie (1973)	18×24
18	Mosier and Taube (1985b)	20×20
19	Kumar *et al.* (1986)	20×23
20	Carrie (1973)	20×35
21	Boe and Cheng (1991)	20×35
22	Chandrasekharan and Rajagopalan (1989)	24×40
23	Chandrasekharan and Rajagopalan (1989)	24×40
24	Chandrasekharan and Rajagopalan (1989)	24×40
25	Chandrasekharan and Rajagopalan (1989)	24×40
26	Chandrasekharan and Rajagopalan (1989)	24×40
27	Chandrasekharan and Rajagopalan (1989)	24×40
28	McCormick *et al.* (1972)	27×27
29	Carrie (1973)	28×46
30	Kumar and Vannelli (1987)	30×41
31	Stanfel (1985)	30×50
32	Stanfel (1985)	30×50
33	King and Nakornchai (1982)	30×90
34	McCormick *et al.* (1972)	37×53
35	Chandrasekharan and Rajagopalan (1987)	40×100

Table 6.2 Parameters setting for HCFA-HSAM, HCFA-HWFAM, and HCFA-HTSM

Algorithm	Parameter settings
HCFA-HSAM	Initial temperature (T_0): 80 Final temperature (T_f): 0.002 Cooling rate (α): 0.7 Markov chain length (L): 30 Mutation probability (β): 0.8
HCFA-HWFAM	Iteration limit (N_{max}) : 100 Initial mass (W_0): 40 Initial velocity (V_0): 15 Base momentum (T_m) : 100 Mutation probability (β): 0.8
HCFA-HTSM	Maximum number of iterations (N_{max}) : 3000 Maximum number of solution has not been improved (N_{stag}): 1000 Tabu list size (*tls*): 7 Mutation probability (β): 0.8

6.1.1 Solutions allowing singletons

The HGA (Tariq *et al*., 2009) and the GA (Mahdavi *et al*., 2009) are compared in order to demonstrate the power of the proposed algorithm given that singletons are allowed (i.e., L_m=1). The computational results are summarized and compared in Table 6.3. The best values for grouping efficacy (Γ) achieved by our proposed HCFA-HSAM, HCFA-HWFAM, and HCFA-HTSM are the same. In addition, our proposed algorithms are better than or equal to previously reported results except in problems #18 and #27. To be more specific, our proposed algorithms were able to improve the best values for grouping efficacy compared with the HGA method for 23 problems (#1, #5, #6, #7, #10, #12, #13, #14, #15,

#16, #17, #19, #20, #25, #26, #27, #28, #29, #30, #31, #32, #33, and #34); while for 11 problems, our proposed algorithms obtained grouping efficacy values that are equal to the best results of the HGA method. Compared with the GA method, our proposed algorithms improved the best values for grouping efficacy for 5 problems (#16, #18, #20, #25, and #26); while for 16 problems, our proposed algorithms obtained grouping efficacy values that are equal to the best results of the GA method. Thus, it can be concluded that our proposed algorithms perform better than HGA and GA, especially in test problems with larger sizes. One noteworthy observation is that all the best solutions can be found in less than 63 seconds regardless of the problem size, thus illustrating the superiority of our proposed algorithms in solution efficiency.

6.1.2 Solutions not allowing singletons

In order to demonstrate the power of the proposed algorithms when singletons are not allowed (i.e., L_m=2), comparisons against the KHMCF (Unler and Gungor, 2009) were performed. The computational results are summarized and compared in Table 6.4. The best values for grouping efficacy achieved by our proposed HCFA-HSAM, HCFA-HWFAM, and HCFA-HTSM methods are similar and our proposed algorithms yielded values better than or equal to those previously reported except in problem #32. To be more specific, HCFA improved the best grouping efficacy values of the KHMCF method for 9 problems (#12, #14, #16, #17, #18, #20, #26, #31, and #34); while for 10 problems, our proposed algorithms obtained grouping efficacy values that are equal to the best results of the KHMCF method. However, it should be noted that our proposed algorithms can achieve the best solutions in less than 56 seconds regardless of the problem size, thereby illustrating the superiority of our proposed algorithms in solution efficiency.

Table 6.3 The computational results in the case where singletons are allowed (L_m=1)

Test instances			HGA	GA [a]			HCFA-HSAM [b]			HCFA-HWFAM [b]			HCFA-HTSM [b]		
No.	Source	Size (m×p)	Γ(%)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)
1	King and Nakornchai (1982)	5×7	73.68	-	-	-	3	75.00^{+}	0.17	3	75.00^{+}	0.02	3	75.00^{+}	0.14
2	Waghodekar and Sahu (1984)	5×7	69.57	2	69.57	0.01	2	69.57	0.13	2	69.57	0.01	2	69.57	0.12
3	Seifoddini (1989)	5×18	79.59	2	79.59	0.02	2	79.59	0.19	2	79.59	0.03	2	79.59	0.16
4	Kusiak and Cho (1992)	6×8	76.92	2	76.92	0.01	2	76.92	0.15	2	76.92	0.02	2	76.92	0.13
5	Kusiak and Chow (1987)	7×11	58.62	5	60.87	0.02	5	60.87	0.28	5	60.87	0.03	5	60.87	0.22
6	Boctor (1991)	7×11	70.37	4	70.83	0.03	4	70.83	0.24	4	70.83	0.03	4	70.83	0.20
7	Seifoddini and Wolfe (1986)	8×12	68.30	-	-	-	4	69.44^{+}	0.29	4	69.44^{+}	0.03	4	69.44^{+}	0.22
8	Chandrasekharan and Rajagopalan (1986a)	8×20	85.25	3	85.25	0.03	3	85.25	0.29	3	85.25	0.07	3	85.25	0.22
9	Chandrasekharan and Rjagopalan (1986b)	8×20	58.72	2	58.72	0.03	2	58.72	0.28	2	58.72	0.03	2	58.72	0.21
10	Mosier and Taube (1985a)	10×10	70.59	5	75.00	0.03	5	75.00	0.42	5	75.00	0.09	5	75.00	0.32
11	Chan and Milner (1982)	10×15	92.00	3	92.00	0.03	3	92.00	0.29	3	92.00	0.09	3	92.00	0.22
12	Askin and Subramanian (1987)	14×23	70.83	-	-	-	7	73.13^{+}	1.03	7	73.13^{+}	0.38	7	73.13^{+}	0.79
13	Stanfel (1985)	14×24	70.51	7	71.83	0.57	7	71.83	1.12	7	71.83	0.40	7	71.83	0.79
14	McCormick et al.(1972)	14×24	51.96	-	-	-	8	53.26^{+}	2.13	8	53.26^{+}	1.65	8	53.26^{+}	1.78
15	Srinivasan et al.(1990)	16×30	67.83	-	-	-	6	68.99^{+}	1.72	6	68.99^{+}	1.24	6	68.99^{+}	1.24
16	King (1980)	16×43	54.86	7	56.13	1.53	8	56.85^{+}	2.74	8	56.85^{+}	1.30	8	56.85^{+}	2.26
17	Carrie (1973)	18×24	54.95	-	-	-	9	57.73^{+}	2.25	9	57.73^{+}	2.24	9	57.73^{+}	2.00
18	Mosier and Taube (1985b)	20×20	43.45	5	42.94	0.62	5	43.36^{-}	1.83	5	43.36^{-}	1.73	5	43.36^{-}	1.68
19	Kumar et al. (1986)	20×23	49.65	-	-	-	7	50.81^{+}	1.27	7	50.81^{+}	0.97	7	50.81^{+}	1.12

20	Carrie (1973)	20×35	76.14	5	77.91	1.25	5	78.40^{+}	1.58	5	78.40^{+}	0.99	5	78.40^{+}	1.12
21	Boe and Cheng (1991)	20×35	58.38	-	-	-	5	58.38	1.14	5	58.38	1.11	5	58.38	1.19
22	Chandrasekharan and ajagopalan (1989)	24×40	100.00	7	100.00	1.60	7	100.00	1.73	7	100.00	0.73	7	100.00	1.26
23	Chandrasekharan and Rajagopalan (1989)	24×40	85.11	7	85.11	1.92	7	85.11	1.74	7	85.11	2.14	7	85.11	1.29
24	Chandrasekharan and Rajagopalan (1989)	24×40	73.51	7	73.51	1.48	7	73.51	1.76	7	73.51	1.56	7	73.51	1.28
25	Chandrasekharan and Rajagopalan (1989)	24×40	52.50	10	52.87	3.26	11	53.29^{+}	6.23	11	53.29^{+}	7.05	11	53.29^{+}	5.63
26	Chandrasekharan and ajagopalan (1989)	24×40	46.84	12	48.85	6.24	12	48.95^{+}	6.30	12	48.95^{+}	6.15	12	48.95^{+}	7.48
27	Chandrasekharan and Rajagopalan (1989)	24×40	44.85	12	47.26	11.23	12	46.26^{-}	5.76	12	46.26^{-}	5.98	12	46.26^{-}	6.54
28	McCormick et al. (1972)	27×27	54.31	-	-	-	5	54.82^{+}	2.57	5	54.82^{+}	6.21	5	54.82^{+}	2.29
29	Carrie (1973)	28×46	46.43	-	-	-	11	47.23^{+}	6.57	11	47.23^{+}	24.57	11	47.23^{+}	9.76
30	Kumar and Vannelli (1987)	30×41	60.74	-	-	-	14	62.86^{+}	7.88	14	62.86^{+}	13.91	14	62.86^{+}	9.15
31	Stanfel (1985)	30×50	59.66	13	60.12	19.30	13	60.12	9.11	13	60.12	12.78	13	60.12	7.84
32	Stanfel (1985)	30×50	50.51	14	50.83	22.21	14	50.83	11.68	14	50.83	18.21	14	50.83	13.37
33	King and Nakornchai (1982)	30×90	44.67	-	-	-	16	47.85^{+}	22.81	16	47.85^{+}	63.89	16	47.85^{+}	32.82
34	McCormick et al. (1972)	37×53	59.60	-	-	-	3	60.50^{+}	2.20	3	60.50^{+}	22.97	3	60.50^{+}	2.33
35	Chandrasekharan and Rajagopalan (1987)	40×100	84.03	10	84.03	99.63	10	84.03	8.34	10	84.03	17.43	10	84.03	5.67

[a] Run on a Pentium IV, 2.1 GHz PC.
[b] Run on a Pentium IV, 1.6 GHz PC.
$^{+}$: Solutions obtained by the proposed approach are superior to best solutions found in the literature.
$^{-}$: Solutions obtained by the proposed approach are inferior to best solutions found in the literature.

Table 6.4 The computational results in the case where singletons are not allowed (L_m=2)

Test instances			KHMCF [a]			HCFA-HSAM [b]			HCFA-HWFAM [b]			HCFA-HTSM [b]		
No.	Source	Size (m×p)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)	NC	Γ(%)	CPU time(s)
1	King and Nakornchai (1982)	5×7	2	73.68	0.37	2	73.68	0.10	2	73.68	0.01	2	73.68	0.08
2	Waghodekar and Sahu (1984)	5×7	2	62.50	0.34	2	62.50	0.09	2	62.50	0.00	2	62.50	0.08
3	Seifoddini (1989)	5×18	2	79.59	0.96	2	79.59	0.11	2	79.59	0.00	2	79.59	0.09
4	Kusiak and Cho (1992)	6×8	2	76.92	0.78	2	76.92	0.13	2	76.92	0.02	2	76.92	0.10
5	Kusiak and Chow (1987)	7×11	3	53.13	0.37	3	53.13	0.13	3	53.13	0.00	3	53.13	0.10
6	Boctor (1991)	7×11	3	70.37	0.37	3	70.37	0.10	3	70.37	0.00	3	70.37	0.09
7	Seifoddini and Wolfe (1986)	8×12	-	-	-	3	68.29	0.16	3	68.29	0.02	3	68.29	0.11
8	Chandrasekharan and Rajagopalan (1986a)	8×20	-	-	-	3	85.25	0.21	3	85.25	0.03	3	85.25	0.13
9	Chandrasekharan and Rjagopalan (1986b)	8×20	-	-	-	2	58.72	0.26	2	58.72	0.03	2	58.72	0.20
10	Mosier and Taube (1985a)	10×10	3	70.59	1.36	3	70.59	0.20	3	70.59	0.02	3	70.59	0.18
11	Chan and Milner (1982)	10×15	-	-	-	3	92.00	0.26	3	92.00	0.06	3	92.00	0.22
12	Askin and Subramanian (1987)	14×23	5	65.75	0.76	5	69.86^{+}	0.46	5	69.86^{+}	0.07	5	69.86^{+}	0.33
13	Stanfel (1985)	14×24	5	69.33	0.82	5	69.33	0.45	5	69.33	0.07	5	69.33	0.36
14	McCormick et al.(1972)	14×24	6	50.48	2.37	6	51.96^{+}	1.20	6	51.96^{+}	0.60	6	51.96^{+}	1.01
15	Srinivasan et al.(1990)	16×30	4	67.83	0.67	4	67.83	1.04	4	67.83	0.68	4	67.83	0.62
16	King (1980)	16×43	5	54.80	1.52	6	55.90^{+}	1.33	6	55.90^{+}	0.34	6	55.90^{+}	1.13
17	Carrie (1973)	18×24	6	52.83	1.63	6	54.46^{+}	1.02	6	54.46^{+}	0.65	6	54.46^{+}	0.89
18	Mosier and Taube (1985b)	20×20	5	40.29	2.15	5	42.96^{+}	1.52	5	42.96^{+}	1.47	5	42.96^{+}	1.33
19	Kumar et al. (1986)	20×23	-	-	-	6	49.61	0.73	6	49.61	0.39	6	49.61	0.68
20	Carrie (1973)	20×35	4	76.22	0.59	5	76.54^{+}	1.53	5	76.54^{+}	1.21	5	76.54^{+}	1.17

21	Boe and Cheng (1991)	20×35	-	-	-	5	58.15	1.07	5	58.15	0.63	5	58.15	0.84
22	Chandrasekharan and ajagopalan (1989)	24×40	7	100.00	2.88	7	100.00	1.44	7	100.00	0.19	7	100.00	1.06
23	Chandrasekharan and Rajagopalan (1989)	24×40	-	-	-	7	85.11	1.45	7	85.11	0.53	7	85.11	1.20
24	Chandrasekharan and Rajagopalan (1989)	24×40	-	-	-	7	73.51	1.44	7	73.51	0.75	7	73.51	1.10
25	Chandrasekharan and Rajagopalan (1989)	24×40	-	-	-	10	51.97	3.64	10	51.97	2.25	10	51.97	3.36
26	Chandrasekharan and ajagopalan (1989)	24×40	9	47.17	2.26	10	47.37^{+}	2.91	10	47.37^{+}	1.38	10	47.37^{+}	3.40
27	Chandrasekharan and Rajagopalan (1989)	24×40	-	-	-	10	44.87	2.35	10	44.87	2.00	10	44.87	3.13
28	McCormick et al. (1972)	27×27	-	-	-	4	54.27	1.93	4	54.27	3.53	4	54.27	1.50
29	Carrie (1973)	28×46	-	-	-	9	46.06	5.03	9	46.06	12.80	9	46.06	6.90
30	Kumar and Vannelli (1987)	30×41	-	-	-	10	59.52	3.66	10	59.52	2.84	10	59.52	3.99
31	Stanfel (1985)	30×50	12	59.43	4.12	12	60.00^{+}	5.66	12	60.00^{+}	4.53	12	60.00^{+}	6.91
32	Stanfel (1985)	30×50	12	58.86	4.17	11	50.51^{-}	5.77	11	50.51^{-}	6.37	11	50.51^{-}	6.69
33	King and Nakornchai (1982)	30×90	-	-	-	12	46.15	11.35	11	46.15	25.19	12	46.15	13.14
34	McCormick et al. (1972)	37×53	2	56.42	0.73	3	59.85^{+}	2.83	3	59.85^{+}	55.74	3	59.85^{+}	3.19
35	Chandrasekharan and Rajagopalan (1987)	40×100	-	-	-	10	84.03	7.31	10	84.03	7.85	10	84.03	5.19

[a] Run on a Pentium IV, 3.2 GHz PC.
[b] Run on a Pentium IV, 1.6 GHz PC.

6.2 Computational Results for Generalized CFP

To validate the quality of the solutions provided by the proposed algorithms for generalized CFP, we prepared suitable test instances. Since only a limited amount of research on CFP has dealt with machine breakdown or reliability issues, suitable test problems are very rare in the literature. Eight test instances, as shown in Table 6.5, are solved in this thesis. Among them, two (#1 and #5) were drawn from the literature (Jabal Ameli and Arkat, 2008 and Jabal Ameli *et al*., 2008). The remaining six problems were prepared by adding self-creating data, such as MBC, MTBF, and production time (PT), to test situations selected from the literature that have machine-part incidence matrix and process routing data ready. Detailed data of each new test problem are presented in Appendix A. These data include production data of each part and machine reliability information.

Table 6.5 describes the basic problem data and how MBC, MTBF, PT data were created:

1. MBC is set to be any number between 1000 and 1700;
2. MTBF is set to be any number between 800 and 5000;
3. PT is set to be any number between 2 and 6.

Table 6.5 Test instances for generalized CFP

No.	Original source	Size (m×p×r)	L_m	U_m	Randomly generated data
1	Jabal Ameli *et al*. (2008)	9×8×20	2	6	-
2	Kim *et al*.(2004)	10×10×25	2	5	MBC, MTBF
3	Sofianopoulou (1999)	12×20×26	2	5	MBC, MTBF, PT
4	Sofianopoulou (1999)	14×20×45	2	5	MBC, MTBF, PT
5	Jabal Ameli and Arkat (2008)	17×30×63	2	5	-
6	Sofianopoulou (1999)	18×30×59	2	7	MBC, MTBF, PT
7	Lee *et al*. (1997)	30×40×89	2	7	MBC, MTBF, PT
8	Hu and Yasuda (2006)	30×70×149	2	8	MBC, MTBF, PT

MBC: Machine Breakdown Cost (1000~1700) *(rand() %8+10)*100*);
MTBF: Mean Time Between Failure (800~5000) (*rand() %4201+800*);
PT: Production Times (2~6) (*rand() %5+2*).

The parameter values for HGCFA-HSAM, HGCFA-HWFAM, and HGCFA-HTSM

throughout all test instances are described in Table 6.6.

Table 6.6 Parameters setting for HGCFA-HSAM, HGCFA-HWFAM, and HGCFA-HTSM

Algorithm	Parameter settings
HGCFA-HSAM	Initial temperature (T_0): 80 Final temperature (T_f): 0.0002 Cooling rate (α): 0.7 Markov chain length (L): 100 Mutation probability (β): 0.7
HGCFA-HWFAM	Iteration limit (N_{max}) : 100 Initial mass (W_0): 40 Initial velocity (V_0): 15 Base momentum (T_m) : 100 Mutation probability (β): 0.8
HGCFA-HTSM	Maximum number of iterations (N_{max}) : 3000 Maximum number of solution has not been improved (N_{stag}): 1000 Tabu list size (tls): 7 Mutation probability (β): 0.8

Table 6.7 shows the comparison of the computation results for different cellular layout types. The best values obtained for each test problem between the two cellular layout types are indicated by bold characters. As expected, cellular layout type does have meaningful effects on TC and CFFI. Out of the 8 test problems, both cellular layout types yielded the same results in 4 test problems (#1, #2, #5, and #7), while linear double-row layout produced better results than linear single-row layout in the remaining 4 problems (#3, #4, #6, and #8). Furthermore, we observed the explicit preference for linear double-row layout over linear single-row layout based on their run times. Thus, it can be concluded that linear double-row layout (r=2) performs better than linear single-row layout (r=1).

Table 6.8 shows comparisons of our proposed algorithms with the branch and bound (B&B) algorithm using the Lingo 8.0 software in stage I given that linear double-row layout (r=2) is considered. Results show that B&B is able to achieve global optimum in 4 out of 8 test instances (i.e., test instances #1 to #4) in less than 120839 seconds (34 hours). As for other test instances, Lingo was not able to find the optimal solution in a reasonable time due to their gigantic problem sizes. In contrast, our proposed algorithms were able to find the optimal solution for test instances #1 to #4 within 4 seconds. For the other test instances, our proposed algorithms found the best solution in 33 seconds. These findings illustrate the superiority of our proposed algorithms in solution efficiency.

Table 6.9 shows the comparisons of our proposed algorithms with the B&B algorithm in stage II using Lingo 8.0 software given that the linear double-row layout (r=2) is considered. Results show that Lingo was able to achieve global optimum in 7 out of 8 test instances in less than 3 seconds. As for test instance #8, B&B was not able to find the optimal solution after 152017 seconds (42 hours) of running with an objective value (CFFI) of 17.80% due to the large problem size. In contrast, our proposed HGCFA-HSAM yielded a final CFFI value of 19.27% after less than 0.7 seconds. These findings indicate the superiority of our proposed algorithms in solution efficiency.

Table 6.7 Comparisons of computation results for different cellular layout type

Test instances					Linear single-row layout (r=1)					Linear double-row layout (r=2)				
					Stage I			Stage II		Stage I			Stage II	
No.	Source	Size(m×p×r)	L_m	U_m	NC	TC	CPU time (s)	CFFI (%)	CPU time (s)	NC	TC	CPU time (s)	CFFI (%)	CPU time (s)
1	Jabal Ameli *et al*. (2008)	9×8×20	2	6	2	5696	0.50	58.06	0.14	2	5696	0.29	58.06	0.11
2	This study	10×10×25	2	5	2	1919	0.70	70.18	0.16	2	1919	0.39	70.18	0.12
3	This study	12×20×26	2	5	3	407	1.22	24.62	0.20	3	401	0.66	24.62	0.14
4	This study	14×20×45	2	5	3	357	1.38	30.77	0.25	3	348	0.73	32.31	0.16
5	Jabal Ameli and Arkat (2008)	17×30×63	2	5	4	50164	2.32	79.65	0.29	4	50164	1.22	79.65	0.18
6	This study	18×30×59	2	7	3	486	1.91	24.42	0.43	3	478	1.01	24.42	0.26
7	This study	30×40×89	2	7	5	41228	3.43	72.41	0.98	5	41228	1.77	72.41	0.55
8	This study	30×70×149	2	8	4	2426	8.81	18.54	1.21	4	2196	4.47	19.27	0.64

Table 6.8 The computational results for our proposed algorithms (stage I) in the case where linear double-row layout (r=2) is considered

Test instances					HGCFA-HSAM			HGCFA-HWFAM			HGCFA-HTSM			Lingo (B&B)		
No.	Source	Size(m×p×r)	L_m	U_m	NC	TC	CPU time (s)	NC	TC	CPU time (s)	NC	TC	CPU time (s)	NC	TC	CPU time (s)
1	Jabal Ameli *et al*. (2008)	9×8×20	2	6	2	5696*	0.29	2	5696*	0.64	2	5696*	0.44	2	5696*	3
2	This study	10×10×25	2	5	2	1919*	0.39	2	1919*	0.81	2	1919*	0.48	2	1919*	10
3	This study	12×20×26	2	5	3	401*	0.66	3	401*	2.23	3	401*	1.10	3	401*	113
4	This study	14×20×45	2	5	3	348*	0.73	3	348*	3.16	3	348*	1.32	3	348*	120839
5	Jabal Ameli and Arkat (2008)	17×30×63	2	5	4	50164	1.22	4	50164	4.46	4	50164	2.10	-	-	-
6	This study	18×30×59	2	7	3	478	1.01	3	478	4.68	3	478	1.72	-	-	-
7	This study	30×40×89	2	7	5	41228	1.77	5	41228	8.32	5	41228	3.27	-	-	-
8	This study	30×70×149	2	8	4	2196	4.47	4	2196	32.37	4	2196	10.42	-	-	-

*: Global optimum

Table 6.9 The computational results for our proposed algorithms (stage II) in the case where linear double-row layout (r=2) is considered

Test instances					HGCFA-HSAM		HGCFA-HWFAM		HGCFA-HTSM		Lingo (B&B)	
No.	Source	Size(m×p×r)	L_m	U_m	CFFI (%)	CPU time (s)	CFFI (%)	CPU time (s)	CFFI (%)	CPU time (s)	CFFI (%)	CPU time (s)
1	Jabal Ameli *et al*. (2008)	9×8×20	2	6	58.06*	0.11	58.06*	0.09	58.06*	0.11	58.06*	3
2	This study	10×10×25	2	5	70.18*	0.12	70.18*	0.11	70.18*	0.14	70.18*	1
3	This study	12×20×26	2	5	24.62*	0.14	24.62*	0.10	24.62*	0.17	24.62*	4
4	This study	14×20×45	2	5	32.31*	0.16	32.31*	0.13	32.31*	0.19	32.31*	4
5	Jabal Ameli and Arkat (2008)	17×30×63	2	5	79.65*	0.18	79.65*	0.69	79.65*	0.22	79.65*	1
6	This study	18×30×59	2	7	24.42*	0.26	24.42*	0.20	24.42*	0.31	24.42*	95
7	This study	30×40×89	2	7	72.41*	0.55	72.41*	2.35	72.41*	0.69	72.41*	23
8	This study	30×70×149	2	8	19.27	0.64	19.27	1.03	19.27	0.74	17.80	152017

*: Global optimum

CHAPTER 7

FURTHER ANALYSES

In this section, the effects of several strategies and mechanisms adopted in proposed HSAM, HTSM and HWFAM are further analyzed statistically using the statistical software, STATISTICA, to examine their corresponding effects on overall solution efficiency and efficacy.

7.1 Effects of prior estimation of number of cells

As mentioned in section 4.1, a two-stage HCFA was proposed to solve the standard CFP. In the first stage, an initial number of cells was generated quickly, and this is then used as input to the second stage to search for the optimal/near-optimal solution. Hopefully, employing this procedure can not only significantly reduce the time to reach satisfactory solutions, but also make the proposed algorithm more efficient in solving large-scale problems. To further understand the effects of prior estimation of the number of cells (PENC), two options (with and without PENC) were performed in our proposed algorithms (HTSM, HSAM, and HWFAM) to solve the 35 problems shown in Table 6.1. Table 7.1 summarizes the ANOVA (analysis of variance) for algorithm, PENC, and interactions between algorithm and PENC. The small p-value ($p = 0.00364 < 0.05$) suggests that factor PENC is significant at 5% significance level. Furthermore, Figure 7.1 reveals that the option with PENC obviously takes substantially less CPU time to achieve a target value than the option without PENC. Therefore, the effects of employing PENC in our proposed algorithms have been positively confirmed.

Table 7.1 ANOVA for the effects of algorithm and PENC

Source of variation	SS	DF	MS	F	P
Algorithm	137.77548	2	68.88774	1.21012	0.30029
PENC	492.57665	1	492.57665	8.65289	**0.00364***
Algorithm *PENC	4.37942	2	2.18971	0.03847	0.96227
Error	11612.95094	204	56.92623		

*: significance

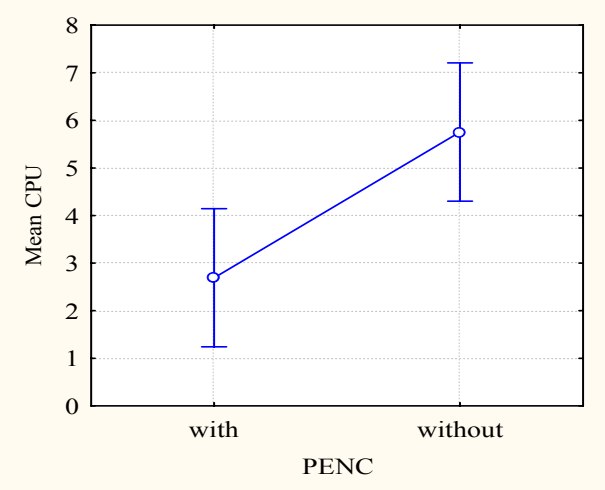


Figure 7.1 Mean CPU comparisons of with and without PENC

The CPU time saving ratio of HTSM, HSAM, and HWFAM with PENC can be calculated via Eq. 7.1. Results are diagrammed in Figure 7.2.

$$SR_{ij} = \frac{T_{ij}^{wpe} - T_{ij}^{pe}}{T_{ij}^{wpe}} \times 100, \tag{7.1}$$

where

SR_{ij} = CPU time saving ratio of i^{th} algorithm in j^{th} numerical example ($0 \leq SR_{ij} \leq 100$),

T_{ij}^{wpe} = Average CPU time of i^{th} algorithm in j^{th} numerical example without prior estimation of cell size,

T_{ij}^{pe} = Average CPU time of i^{th} algorithm in j^{th} numerical example with prior

estimation of cell size.

On average, the CPU time saving ratios of HTSM, HSAM, and HWFAM are 35%, 35%, and 38%, respectively. Additionally, the savings in run time are even more significant as the cell size increases. The effects of employing PENC in our proposed HSAM, HTSM and HWFAM have been successfully justified again through these findings. .

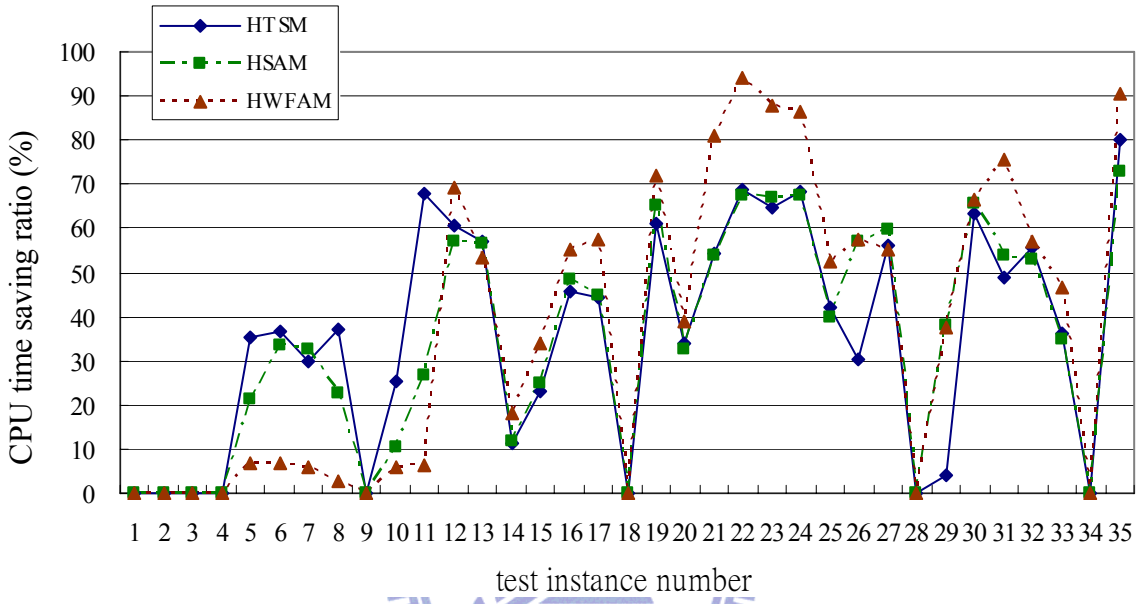


Figure 7.2 The CPU time saving ratios of HTSM, HSAM, and HWFAM

7.2 Effects of mutation strategy in HTSM and HSAM

The mutation strategy has been employed to increase the probability of finding more diversified solutions to bring the searching process to a new and unexplored solution space. In order to further elaborate the effectiveness of the mutation operator for HTSM and HSAM, a paired T-test of 95% statistical analysis of mean difference equal to zero was also carried out. The details are given in Tables 7.2 and 7.3. The small p-value ($p<0.05$) suggests that the data is inconsistent with null hypothesis $H_0: \mu_d = 0$; that is, the two scenarios do not perform equally. Specifically, the HTSM and HSAM approaches with mutation operator performed better than the approaches without mutation operator in terms of finding grouping efficacy for all 35 problems from the literature.

Table 7.2 Paired T-test for the effects of mutation operator for HTSM

Scenario	Mean	Std.Dv.	N	df	T	P
Without mutation operator	63.90	14.89				
With mutation operator	64.25	14.62	35	34	-3.68	0.0008*

*: significance

Table 7.3 Paired T-test for the effects of mutation operator for HSAM

Scenario	Mean	Std.Dv.	N	df	T	P
Without mutation operator	63.79	14.93				
With mutation operator	64.31	14.55	35	34	-3.86	0.0005*

*: significance

7.3 Effects of evaporation, precipitation, and insertion-move in WFA

The water evaporation and precipitation operations are synonymous with the "escaping from local optima" mechanism that many heuristic algorithms nowadays possess to avoid being trapped and to explore more solution spaces. Additionally, the insertion-move included assures that high-quality neighborhood solutions can be found at each iteration of the algorithm. These three factors and their effects on solution qualities are thus examined in an experiment. The evaporation factor has three settings: no evaporation, fixed-ratio evaporation, and velocity-based evaporation. The precipitation factor has three settings: no precipitation, regular precipitation, and evaporation-based precipitation. Finally, the insertion-move factor has two options: with or without insertion-move. Legitimate combinations of the three factors comprise 10 testing scenarios as listed in Table 7.4. In each scenario, 35 test instances of Table 6.1 are computed. The average grouping efficacy of the 35 instances was recorded and ANOVA was also been carried out, and the results are shown in Table 7.5. The small p-value suggests that at 0.05 level of significance, insertion-move beats the other two factors and becomes the most significant and dominant factor in terms of solution quality.

The 10 scenarios were separated into two groups. Scenarios adopting the insertion-move (scenarios #1, #3, #5, #7, and #9) result in better objective values and obviously surpass the other group (scenarios #2, #4, #6, #8, and #10), which does not

include the insertion-move in the algorithm. Insertion-move beats the other two factors and becomes the most significant and dominant factor in terms of solution quality. This implies that the water evaporation and precipitation operations may not be as significant as thought in the solution process of CFP, though considerable efforts have been spent in designing the contents of both operations. The importance of a good neighborhood-searching method, such as the insertion-move we proposed, can never be overemphasized. This conclusion is applicable to any meta-heuristic algorithms. Although the water evaporation and precipitation operations may not be critical in solving the CFP in this thesis, we still believe that the water-flow-like logic, with proper design and collaboration in its internal operations, can be applied to solve other combinatorial optimization problems.

Table 7.4 Experimental testing scenarios

Scenario #	Evaporation setting	Precipitation setting	Insertion-move Option	Mean grouping efficacy (%)
1	velocity based	Evaporation-based	With	64.2832
2	velocity based	Evaporation-based	Without	59.1495
3	velocity based	regular (t = 20)	With	64.2816
4	velocity based	regular (t = 20)	Without	59.3005
5	fixed ratio (0.05)	Evaporation-based	With	64.2954
6	fixed ratio (0.05)	Evaporation-based	Without	59.3558
7	fixed ratio (0.05)	regular (t = 20)	With	64.2625
8	fixed ratio (0.05)	regular (t = 20)	Without	59.2771
9	no evaporation	no precipitation	With	64.2664
10	no evaporation	no precipitation	Without	59.4040

Table 7.5 ANOVA for the effects of evaporation, precipitation, and insertion-move

Source of variation	SS	DF	MS	F	p
evaporation	3	2	1	0.005	0.995
precipitation	4	2	2	0.006	0.994
Insertion	6408	1	6408	20.768	0.000*
evaporation*precipitation	11	4	3	0.009	1.000
evaporation*insertion	2	2	1	0.003	0.997
precipitation*insertion	6	2	3	0.010	0.990
evaporation*precipitation*insertion	9	4	2	0.008	1.000
Error	188841	612	309		

*: significance

7.4 Effects of tabu list size

In order to elaborate the effects of tabu list size (TLS), six levels of TLS (0, 7, 12, 17, 22, and 27) were set and ANOVA was carried out. The details are shown in Table 7.6. The high p-value (p = 1) suggests that at the 0.05 level of significance, we can not reject the null hypothesis that TLS does not have a significant effect on grouping efficacy. From Figure 7.3, it can be observed that the six levels of TLS do not have significant differences in terms of mean grouping efficacy. Since TLS-7 works better than others, we used it as the suggested TLS setting in this thesis.

Table 7.6 ANOVA for the effects of tabu list size

Source of variation	SS	DF	MS	F	p
TLS	0.006	5	0.001	5.944E-06	1
Error	43264.064	204	212.079		

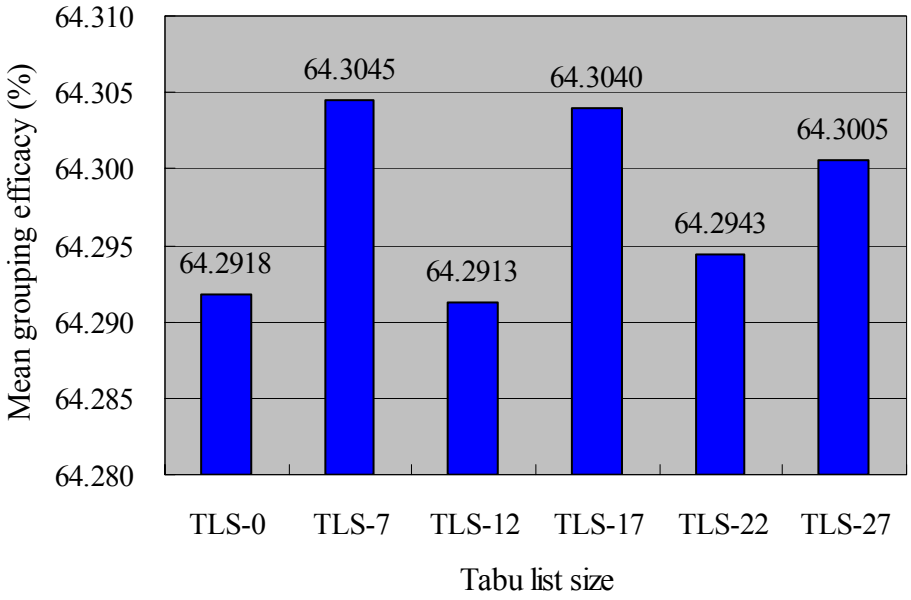


Figure 7.3 Mean grouping efficacy for six levels of TLS

CHAPTER 8

CONCLUSIONS

CFP is the first and most difficult aspect of constructing a preliminary CMS. Considering the issues of production volume, production sequence, alternative process routings, machine reliability, cell layout, and the sequence of machines within cells in the design of CMS make the CFP complex but more realistic. However, very few researchers have addressed these issues simultaneously in the design of CMS. In this thesis, two types of CFP are addressed: the standard CFP with a binary machine-part incidence matrix consideration, and the generalized CFP with more factors and system constraints. A mathematical model in terms of maximization of grouping efficacy has been formulated to express the standard CFP in scenarios where singletons are allowed or not. Due to the complexity of this model, a two-stage HCFA merging an SCM-based clustering algorithm and SA/WFA/TS method has been proposed to solve this model quickly and effectively. In the generalized CFP, a two-stage multi-objective mathematical programming model has been formulated to integrate cell formation, cell layout, and intracellular machine layout simultaneously with the considerations of alternative process routings, operation sequences, production volume, production times, machine reliability, and different cellular layout type. As problems in the two stages are NP-hard, a two-stage HGCFA merging a generalized SCM-based clustering algorithm and SA/WFA/TS method has been proposed to solve this model quickly and effectively. Unlike most existing methods, the proposed approach does not demand a priori specification of the number of cells. It is automatically calculated and determined, such that the best objective value may be achieved. Illustrative examples, comparisons, and experimental analyses have demonstrated the effectiveness of the proposed models and solution algorithms.

The main contributions of this thesis may be highlighted as follows: (1) we have

formulated a mathematical programming model to express the standard CFP in cases where singleton/no singleton is allowed; (2) a two-stage HCFA merging an SCM-based clustering algorithm and SA/WFA/TS method has been proposed to solve the standard CFP quickly and effectively; (3) we have formulated a two-stage multi-objective mathematical programming model to integrate cell formation, cell layout, and intracellular machine layout simultaneously with the considerations of alternative process routings, operation sequences, production volume, production times, machine reliability, and different cellular layout type; (4) we have presented a new performance measure, the CFFI, to evaluate the goodness of intracellular machine layout; (5) a two-stage hybrid generalized CF algorithm (HGCFA) merging a generalized SCM-based clustering algorithm and SA/WFA/TS method has been proposed to determine cell formation, cell layout, and intracellular machine layout in the cellular manufacturing system; and (6) in addition to the commonly used linear single-row layout, effects of adopting the linear double-row layout have been investigated in this study.

Several opportunities exist for further research. They are summarized as follows:

1. Extension of the proposed generalized cell formation model

Several other factors may be added into the current proposed generalized cell formation model or even treat them as decision objectives. These factors include the cell load variation, machine utilization, machine duplication and the group scheduling.

2. Application of WFA to other combinatorial optimization problems

The WFA is a novel heuristic approach that deserves more attention. It may also be interesting to adopt the WFA logic to solve many other combinatorial optimization problems.

3. Integration of cell formation, cell layout, and intracellular machine layout by one single approach/stage

In the generalized CFP, a two-stage approach has been proposed to solve the cell formation, cell layout, and intracellular machine layout. The aim of stage I is to solve cell

formation and inter-cell layout simultaneously and the primary work of stage II is to determine machine layout (sequence) in each cell based on the given cell formation determined in stage I. Study could be carried out to see whether cell formation, cell layout, and intracellular machine layout can be solved by one single approach/stage.

APPENDIX A

Table A.1 Machine reliability information for test instance 2

Machine	Breakdown Cost	MTBF (min)
1	1000	1019
2	1700	1699
3	1300	3821
4	1700	1543
5	1600	1961
6	1500	3535
7	1100	4121
8	1500	3327
9	1500	3612
10	1100	889

Table A.2 Machine reliability information for test instance 3

Machine	Breakdown Cost	MTBF (min)
1	1500	4935
2	1500	1080
3	1400	3155
4	1200	3171
5	1600	1227
6	1100	1638
7	1000	3095
8	1100	2201
9	1400	2297
10	1300	1719
11	1600	1119
12	1600	2416

Table A.3 Machine reliability information for test instance 4

Machine	Breakdown Cost	MTBF (min)
1	1000	1463
2	1400	3578
3	1700	1331
4	1700	4885
5	1300	3097
6	1500	1211
7	1700	4503
8	1200	1581
9	1500	3204
10	1300	2381
11	1400	3858
12	1200	2185
13	1500	3513
14	1700	3649

Table A.4 Machine reliability information for test instance 6

Machine	Breakdown Cost	MTBF (min)
1	1400	3714
2	1600	2510
3	1400	1771
4	1400	3555
5	1400	3169
6	1200	3937
7	1700	1759
8	1000	2239
9	1400	2087
10	1700	1984
11	1100	4175
12	1700	2105
13	1600	1531
14	1700	4111
15	1500	899
16	1700	2206
17	1400	1771
18	1700	1759

Table A.5 Machine reliability information for test instance 7

Machine	Breakdown Cost	MTBF (min)
1	1000	2463
2	1200	1493
3	1500	4030
4	1500	1313
5	1600	1941
6	1700	1351
7	1200	1168
8	1500	2668
9	1500	2874
10	1200	2504
11	1200	1668
12	1500	3758
13	1100	4303
14	1600	2477
15	1200	916
16	1600	4380
17	1300	3465
18	1700	862
19	1700	3452
20	1200	856
21	1000	4776
22	1000	1127
23	1600	3128
24	1300	4805
25	1000	3913
26	1600	2659
27	1200	4730
28	1200	2571
29	1000	4140
30	1200	3201

Table A.6 Machine reliability information for test instance 8

Machine	Breakdown Cost	MTBF (min)
1	1200	2887
2	1600	3476
3	1700	4977
4	1700	1136
5	1500	3809
6	1500	3245
7	1400	1976
8	1500	2552
9	1600	1832
10	1700	3682
11	1100	1740
12	1200	4241
13	1600	1446
14	1000	1020
15	1200	3027
16	1300	3284
17	1500	4316
18	1000	1161
19	1200	4012
20	1300	2195
21	1600	1968
22	1300	3403
23	1500	2056
24	1700	1838
25	1100	1137
26	1100	1715
27	1400	2616
28	1200	3491
29	1000	1118
30	1600	1019

Table A.7 Production data of test instance 3

PN	PV	RN	PS	PT (min)
1	1	1	6 5 3 12 8 11	4 2 3 3 4 4
2		1	10 11 6 5 7	6 2 6 4 3
		2	6 9 2 9 5	6 5 4 5 5
3	1	1	10 2 4 1 5 11	4 3 6 5 5 2
4	1	1	4 1 10 3 6	4 3 2 3 4
5	1	1	12 2 6	3 3 5
		2	9 3 7	2 2 5
6	1	1	8 5 2 6	2 6 6 4
7	1	1	12 8	4 5
8	1	1	9 2 4	5 6 5
9	1	1	2 7 3 11 12	6 5 6 3 5
10	1	1	1 7 4 2 9	6 3 6 2 2
11	1	1	12 3 2 11 8 5	3 4 3 5 6 6
12	1	1	11 10 5 8	5 4 6 3
		2	10 9 3 7	3 4 3 3
13	1	1	10 7 11 5	2 6 2 2
14	1	1	3 4 10 7	2 4 3 5
		2	7 9 1 5	2 3 2 6
15	1	1	5 2 4	5 5 3
16	1	1	6 7 11 3 2	2 3 3 2 3
17	1	1	2 3 11 6	4 4 6 5
		2	5 8 9 10	6 4 4 3
		3	8 4 3 7	6 3 3 2
18	1	1	4 8 5	4 4 4
19	1	1	3 2 10 9 12	6 4 5 6 3
20	1	1	6 7 2	4 4 2

PN: Part Number; PV: Production Volume; RN: Routing Number; PS: Production Sequence; PT: Production Times

Table A.8 Production data of test instance 4

PN	PV	RN	PS	PT (min)	PN	PV	RN	PS	PT (min)
1	1	1	6 5 3 12 8 11	6 5 3 6 5 4	11	1	1	12 3 2 11 8 5	5 4 3 2 5 5
		2	6 14 3 12 8 11	6 3 6 2 3 3			2	12 3 13 11 8 5	6 4 4 2 3 2
2	1	1	10 11 6 5 7	4 4 5 5 4			3	12 3 2 11 8 14	6 6 3 4 4 5
		2	10 11 6 14 7	4 6 4 6 5			4	12 3 13 11 8 14	5 3 6 6 6 2
3	1	1	10 2 4 1 5 11	3 3 6 2 4 5	12	1	1	11 10 5 8	5 5 2 6
		2	10 13 4 1 5 11	3 3 5 2 4 4			2	11 10 14 8	5 2 6 2
		3	10 2 4 1 14 11	5 3 6 5 2 6	13	1	1	10 7 11 5	5 3 5 5
		4	10 13 4 1 14 11	5 6 4 2 2 6			2	10 7 11 14	2 5 2 4
4	1	1	4 1 10 3 6	2 5 2 2 3	14	1	1	3 4 10 7	6 6 4 3
5	1	1	12 2 6	4 2 2	15	1	1	5 2 4	4 3 3
		2	12 13 6	5 4 2			2	5 13 4	6 3 6
6	1	1	8 5 2 6	4 6 2 6			3	14 2 4	5 2 5
		2	8 5 13 6	5 2 3 6			4	14 13 4	3 4 3
		3	8 14 2 6	3 5 3 6	16	1	1	6 7 11 3 2	4 5 4 4 4
		4	8 14 13 6	4 4 3 6			2	6 7 11 3 13	6 4 2 5 4
7	1	1	12 8	2 6	17		1	2 3 11 6	2 5 6 5
8	1	1	9 2 4	6 5 2			2	13 3 11 6	6 2 4 5
		2	9 13 4	6 5 6	18		1	4 8 5	3 4 2
9	1	1	2 7 3 11 12	6 4 2 2 6			2	4 8 14	4 3 2
		2	13 7 3 11 12	3 3 3 3 6	19		1	3 2 10 9 12	3 4 3 5 4
10	1	1	1 7 4 2 9	5 4 6 5 5			2	3 13 10 9 12	5 2 5 5 3
		2	1 7 4 13 9	3 4 6 2 4	20		1	6 7 2	5 2 3
							2	6 7 13	5 3 5

Table A.9 Production data of test instance 6

PN	PV	RN	PS	PT (min)	PN	PV	RN	PS	PT (min)
1	1	1	3 9 2 15	6 5 6 5	15	1	1	13 11 15	2 6 6
		2	17 9 2 15	6 5 6 5	16	1	1	10 6 9 16	5 4 3 5
		3	14 12 15 4	4 3 6 5			2	8 11 5 4	6 3 5 2
		4	5 14 6 12	6 4 4 4	17	1	1	9 12 14 10	2 3 5 5
2	1	1	2 15 7 9	4 3 6 5	18	1	1	15 8 9 10 2	2 5 6 4 5
		2	2 15 18 9	4 3 6 5	19	1	1	16 12 4 10	2 5 4 3
3	1	1	2 9 7 13	3 5 4 2	20	1	1	4 11 6	2 3 6
		2	2 9 18 13	3 5 4 2	21	1	1	4 3 15 6	3 2 5 3
		3	10 4 3 9	5 6 5 4			2	4 17 15 6	3 2 5 3
		4	10 4 17 9	5 6 5 4	22	1	1	10 6 4 12	5 5 6 4
4	1	1	13 6 2 14 7	6 5 4 5 2	23	1	1	12 10 3 14 2	3 6 4 4 5
		2	13 6 2 14 18	6 5 4 5 2			2	12 10 17 14 2	3 6 4 4 5
5	1	1	11 14 5 8	2 6 6 4	24	1	1	8 5 7	3 5 6
6	1	1	7 10 14 12	6 6 2 6			2	8 5 18	3 5 6
		2	18 10 14 12	6 6 2 6	25	1	1	12 15 13 2	3 4 3 3
		3	1 10 6 9	2 3 5 3	26	1	1	7 6 12	2 4 4
7	1	1	4 12 5 16 11	6 5 2 2 6			2	18 6 12	2 4 4
8	1	1	10 12 11 4	6 6 5 5	27	1	1	4 2 12	3 3 4
		2	2 7 8 5	3 2 3 5	28	1	1	16 4 10 2	5 5 4 2
		3	2 18 8 5	3 2 3 5	29	1	1	7 11 3	4 4 6
		4	1 11 3 9	6 2 3 6			2	18 11 3	4 4 6
		5	1 11 17 9	6 2 3 6			3	7 11 17	4 4 6
9	1	1	11 8 13 1 2	2 6 2 4 5			4	18 11 17	4 4 6
10	1	1	7 13 10 4 6	4 2 3 5 3			5	7 5 4	3 2 4
		2	18 13 10 4 6	4 2 3 5 3			6	18 5 4	3 2 4
11	1	1	15 1 14 6	4 4 2 3	30	1	1	2 14 11	5 4 2
		2	5 6 8 1	5 2 2 6					
12	1	1	8 7 4	5 2 4					
		2	8 18 4	5 2 4					
		3	4 13 15	2 3 4					
13	1	1	15 9 3	3 5 3					
		2	15 9 17	3 5 3					
14	1	1	10 16 12	6 6 2					

Table A.10 Production data of test instance 7

PN	PV	RN	PS	PT	PN	PV	RN	PS	PT(min)
1	155	1	8 9 15 16 19 20	6 2 3 5 6 3	21	75	1	1 22 16 13	2 4 5 3
		2	8 30 28 2	3 3 3 3			2	1 22 24	6 2 3
2	160	1	1 22 24 30	2 3 2 5			3	6 14 18	5 5 4
		2	10 12 13 17	2 4 6 2	22	100	1	1 2 21 22 24	4 2 3 6 4
3	135	1	7 12 16 13	2 4 2 5			2	8 9 15 24 2	2 2 2 5 4
		2	12 13 17	3 5 5			3	9 5 11 2	2 3 3 4
		3	9 13 17 30	5 2 6 2	23	140	1	8 9 24 2	6 3 5 2
4	150	1	6 11 1 4 7	5 4 6 3 5			2	1 22 24	4 2 6
		2	1 2 22 24	6 4 2 3			3	1 22 24 20	5 2 4 6
		3	6 11 28 30	6 4 6 2			4	7 30 8 5	3 4 3 6
5	210	1	3 4 5 23 1	2 4 6 3 5	24	62	1	11 14 18	6 6 5
		2	6 11 5	4 5 2	25	85	1	7 26 28 30	6 4 4 4
		3	7 26 30	5 5 4			2	7 30 2 5	6 5 3 5
6	230	1	26 28 30	3 5 2	26	185	1	8 30 12 17	4 6 2 3
		2	12 14 20 22	6 4 4 3			2	14 18 15 18	3 5 6 3
7	85	1	3 4 5 23 27 29	3 4 5 2 3 4			3	12 15 6 12	4 4 3 4
8	90	1	7 30 8 2	3 4 4 3			4	10 12 17	3 3 3
		2	7 26 30	4 2 2	27	55	1	6 14 18 12 13	3 4 3 2 3
		3	9 26 5 11 2	4 4 2 4 6			2	6 11 14 18	3 6 6 6
9	95	1	8 10	2 6	28	130	1	2 22 24	5 6 3
		2	6 7 26	6 2 4			2	22 24 5 22	3 5 4 5
		3	1 27 4	2 6 5	29	125	1	1 2 22 24	4 4 5 2
10	86	1	12 13 17	4 5 3	30	135	1	26 8 5 4	4 3 2 6
		2	13 19 17	5 4 2	31	65	1	12 5 8 9	4 6 3 6
		3	8 9 12	4 5 3			2	13 15 6 3	2 5 5 4
11	55	1	18 19 22	2 5 4			3	10 12 13 17	4 4 6 2
		2	9 15 16	5 2 5	32	90	1	3 4 23 27	4 5 6 3
		3	12 13 17 5	5 4 2 4			2	3 24 23 6	3 4 4 6
12	120	1	12 13 17	5 5 4	33	100	1	26 28 30	6 5 4
		2	8 12 9 30 7	4 6 2 2 4			2	5 8 9 21	3 4 4 2
13	142	1	8 9 16 19 20	2 4 2 4 5	34	90	1	12 13 17 5 2	5 4 5 4 5
14	140	1	11 18	2 6			2	3 4 5 23 25 27	6 5 5 2 4 3
		2	7 12 16 13	5 4 5 3	35	120	1	9 15 19 20	5 2 5 3
15	100	1	2 22 24	6 2 2	36	130	1	3 4 25 27 29	6 2 2 2 4
		2	9 26 5 2	3 4 4 4			2	6 11 28 30	2 5 6 2
		3	7 28 30	5 2 2			3	8 12 9 30	2 5 4 2
16	65	1	11 2 9	5 2 2	37	145	1	13 19 17	3 6 5
		2	9 15 16 19 20	3 3 6 3 4			2	3 4 5 23 25 29	3 6 4 6 2 4
17	85	1	13 19 17	2 6 5			3	10 14 11 13	5 3 3 2
		2	10 14 11 13	5 3 6 5	38	250	1	7 26 30	2 3 3
		3	1 2 22 24	5 6 4 3			2	10 11 10 13	5 3 5 2
18	125	1	8 9 15 16 19 20	5 2 5 6 5 4	39	60	1	6 15 9 6	3 2 6 4
		2	8 9 15 24 2	6 5 3 3 5			2	6 11 18	5 5 5
19	102	1	1 22 24	5 4 3	40	90	1	7 26 28	5 5 6
20	105	1	10 13 17	3 5 3					

Table A.11 Production data of test instance 8

PN	PV	RN	PS	PT	PN	PV	RN	PS	PT(min)
1	1	1	3 9 2 15 25	2 5 2 4 4	18	1	1	15 7 10 29 10 2 12	3 6 5 6 6 3 5
		2	17 9 2 30 15	6 2 3 4 6	19	1	1	16 12 8 4 10 30	5 6 2 6 5 2
		3	14 12 28 11 8	4 4 6 3 3	20	1	1	4 10 18 20 27	6 2 4 5 4
		4	14 21 6 12 27	3 2 2 3 2	21	1	1	4 3 15 21 6 29 1	5 4 6 5 6 3 2
2	1	1	2 15 19 7 9 22 24	6 5 2 6 3 6 5			2	4 17 15 18 6 21 28	6 2 6 5 5 2 5
		2	2 15 23 12 9 3 28	3 5 3 2 3 6 3	22	1	1	10 6 4 27 12 17 25	2 6 4 2 3 3 6
3	1	1	2 9 22 7 13 19	5 4 4 3 3 5	23	1	1	12 10 3 18 14 2 24	4 2 6 5 2 5 3
		2	2 9 18 13 26 25	5 4 3 5 4 3			2	12 10 17 14 2 21 25	6 3 2 6 5 3 2
		3	10 20 4 3 9 20	3 4 6 5 5 3	24	1	1	8 5 7 13 18 24 28	6 5 6 2 3 5 3
		4	10 4 30 15 9 28	5 2 4 4 3 4			2	8 5 18 22 26 19 1	5 4 4 5 2 4 6
4	1	1	13 6 22 2 14 7 26 1	4 4 6 3 2 2 6 2	25	1	1	12 17 23 2 18 22 20 19 28	3 6 3 2 4 3 3 4 3
		2	13 6 2 21 14 18 24 19	4 4 5 4 5 4 5 3	26	1	1	7 6 17 30 12 22 29 25	3 4 3 2 2 4 5 5
5	1	1	11 14 3 5 7 13 27	2 6 3 5 4 6 5			2	5 8 9 12 15 23 17 22	2 4 4 6 6 4 4 5
6		1	7 10 14 12	2 5 3 5	27	1	1	4 15 23 12 22 26 19 18	4 5 6 6 5 6 3 5
		2	18 10 14 12	6 3 6 6	28	1	1	16 4 17 21 10 1 28 22 20	4 6 4 3 4 2 5 2 4
		3	1 10 6 9	4 3 3 4	29	1	1	7 27 11 3 14 18 28	6 6 5 3 2 2 5
7	1	1	4 12 5 16 11 19 22 27	2 4 6 2 2 5 3 5			2	18 11 26 3 21 9 27	3 6 4 4 5 6 4
8	1	1	10 12 11 4	6 2 3 5			3	7 11 22 24 17 23 30	3 2 2 6 4 2 2
		2	2 7 8 5	2 2 3 3			4	18 11 30 17 23 25 27	6 4 5 5 3 4 5
		3	2 18 8 5	5 5 2 6			5	7 5 16 4 29 22 20	2 5 2 4 4 6 5
		4	1 11 3 9	5 2 4 4			6	18 10 5 4 23 22 24	6 4 3 2 5 2 5
		5	1 11 17 9	6 6 2 4	30	1	1	2 14 11 24 22 26 19 20 27 28	5 2 5 5 4 2 6 5 4 2
9	1	1	11 8 13 1 2 22 19 28	4 2 2 2 3 5 3 5	31	1	1	6 5 3 12 8 11	5 6 4 4 4 4
10	1	1	7 13 10 4 6 23	5 5 6 5 2 3			2	6 14 3 12 8 11	3 5 6 3 5 4
		2	18 13 10 4 6 30	4 4 3 6 3 6	32	1	1	10 11 6 5 7 14 20 22	3 4 5 3 2 6 4 2
11	1	1	15 1 14 6	5 4 2 5			2	10 11 6 14 7 26 22 28	3 2 2 4 4 2 3 6
		2	5 6 8 1	4 5 6 2	33	1	1	10 2 4 1 5 11 21 30	3 6 2 5 4 5 3 4
12	1	1	8 7 4	5 3 2			2	10 13 4 1 5 11 18 28	6 4 4 5 2 4 2 4
		2	8 18 4	3 4 6			3	10 2 4 1 14 11 30 22	2 3 6 2 3 3 3 5
		3	4 13 15	4 2 2			4	10 13 4 14 26 27 19 22	6 2 3 5 3 3 3 5
13	1	1	15 9 3	6 2 2	34	1	1	5 8 9 12 15 17 23 4 1 29 16	6 4 5 2 6 3 3 2 3 6 6
		2	15 9 17	3 6 3	35	1	1	12 2 29 6 22 25	4 6 6 2 4 6
14	1	1	20 18 22 24 26 19	4 2 6 6 6 3			2	12 13 6 30 24 26	5 6 3 3 3 4
15	1	1	13 11 15 18 20 28	2 5 2 3 4 6	36	1	1	8 5 2 6 23 22 20	2 4 6 4 5 3 6
16	1	1	24 4 13 29 16 1 30	3 3 4 3 5 4 4			2	8 5 13 6 30 16 22	2 4 6 6 4 4 4
		2	5 8 9 17 23 12 15	5 2 2 6 4 3 4			3	8 14 2 6 30 27 28	2 3 5 5 6 2 3
17	1	1	9 17 21 11 22 28 19 20	6 5 4 4 5 6 4 2			4	8 14 22 13 6 24 29	5 3 2 4 4 6 4

Table A.11 Continued

PN	PV	RN	PS	PT	PN	PV	RN	PS	PT(min)
37	1	1	12 8 18 20 27	4 3 2 2 4	55	1	1	12 2 30 6 20 18	4 3 5 2 2 3
38	1	1	9 2 4 12 18 22 28 16	6 2 3 5 5 6 2 6			2	12 20 13 6 22 24	3 3 5 3 4 4
		2	9 14 13 4 21 22 19 27	6 6 4 4 5 5 2 3	56	1	1	8 5 25 2 6 18 13	6 4 5 3 2 3 6
39	1	1	2 7 3 11 12 17 19 20 30	4 2 6 5 3 2 2 4 4			2	8 5 13 6 16 29 20	2 3 4 6 4 5 3
		2	13 7 3 11 12 23 22 24 1	6 4 6 4 3 4 3 2 4			3	8 14 2 6 24 21 28	2 3 6 4 2 3 3
40	1	1	1 7 4 2 9 13 23	6 6 3 5 5 6 2			4	8 14 30 13 6 29 21	3 6 6 3 4 2 6
		2	1 7 4 13 9 23 28	6 4 6 6 6 3 4	57	1	1	12 21 9 10 22 27 19 28	2 3 6 3 5 2 2 6
41	1	1	12 3 11 8 5 18 15 26 2	3 6 4 2 3 5 5 4 2	58		1	9 1 4 10 13 23 30 25	4 3 5 5 2 2 6 5
		2	12 3 13 11 8 5 26 28 1	6 6 6 2 2 5 4 4 3			2	9 13 4 30 26 24 19 23	5 3 2 6 4 6 4 2
		3	12 3 2 11 19 14 19 22 27	2 2 6 4 5 5 3 3 5	59	1	1	2 7 3 11 12 23 30	5 4 6 5 5 3 5
		4	12 3 13 11 8 14 22 21 17	6 3 2 5 3 2 3 2 6			2	13 7 3 11 12 22 18	4 5 4 6 3 6 4
42	1	1	11 10 15 5 8 26 28 18	6 2 6 2 6 5 6 3	60	1	1	1 7 4 2 9 18 30	3 6 3 5 2 5 4
		2	11 10 14 8 21 15 23 1	2 2 6 4 3 2 6 5			2	1 7 4 13 9 22 28	3 3 3 5 4 3 4
43	1	1	10 7 11 20 5 30	3 5 3 6 3 5	61	1	1	12 3 2 11 8 5 28 22 21	5 2 2 2 3 3 6 2 6
		2	10 7 11 15 14 27	2 2 3 6 2 5			2	12 3 13 11 8 5 23 26 19	6 6 5 2 3 5 5 3 6
44	1	1	3 14 11 10 7 13 16 29	5 4 2 5 5 4 4 3			3	12 3 2 20 8 14 23 27 19	2 2 2 2 2 4 3 2 6
45	1	1	5 17 2 4 21 18 24	4 5 2 2 6 2 4			4	12 3 13 11 8 14 23 30 27	3 3 3 4 2 5 4 3 3
		2	5 13 4 30 20 25 1	5 6 6 5 4 2 5	62	1	1	11 10 22 5 8 20 24	4 2 4 3 2 3 6
		3	14 2 13 19 21 26 30	3 2 4 6 4 6 2			2	11 10 14 30 8 15 28	6 2 4 5 6 4 3
		4	14 13 18 4 22 25 20	6 3 6 4 3 2 4	63	1	1	10 7 11 5 21 17 30	3 3 5 5 5 4 3
46	1	1	6 7 11 3 16 2 23 30	5 5 3 6 5 3 6 2			2	10 7 11 14 22 26 19	6 5 4 2 3 6 6
		2	6 7 11 3 13 30 23 28	2 4 3 5 6 2 5 2	64	1	1	1 4 10 7 28 22 19 26	2 5 2 4 5 6 5 6
47	1	1	2 3 11 6 25	5 5 4 4 6	65	1	1	5 12 14 10 18 25 30	3 6 2 4 6 5 5
		2	13 3 11 18 6	4 3 3 4 6			2	5 13 4 26 24 17 22	3 4 6 3 4 2 5
48	1	1	4 8 13 5	3 6 6 6			3	14 2 4 30 22 25 24	4 2 5 5 6 2 3
		2	4 8 14 25	4 3 6 4			4	14 13 4 24 30 22 26	4 3 4 5 5 4 6
49	1	1	3 2 28 10 9 12	5 2 5 6 3 2	66	1	1	6 7 11 3 2 20 18	2 3 5 3 4 3 5
		2	3 13 10 25 9 12	4 5 6 6 5 3			2	6 7 11 3 13 30 27	3 6 5 2 3 6 5
50	1	1	6 14 7 21 2 30	3 4 4 5 4 4	67	1	1	2 3 11 30 6 25 21	2 5 4 4 6 4 3
		2	6 27 7 13 25 1	4 5 3 3 4 4			2	13 3 11 6 22 29 17	3 4 6 6 2 3 3
51	1	1	6 5 3 12 8 11 18	6 3 3 6 3 6 2	68	1	1	4 13 8 17 5 29	5 5 5 4 3 6
		2	6 14 3 12 8 11 25	3 3 6 5 6 5 5			2	4 8 19 14 22 30	4 4 4 4 5 2
52	1	1	10 11 6 5 7 28	2 4 3 6 6 4	69	1	1	3 2 10 9 12 1	6 2 4 5 3 2
		2	10 11 6 26 14 7	5 2 2 5 2 3			2	3 20 13 10 9 12	5 6 5 4 5 6
53	1	1	10 2 27 4 1 5 11	6 4 4 5 3 6 4	70	1	1	6 17 7 2	4 3 5 4
		2	10 13 4 1 5 11 19	6 4 2 5 2 2 3			2	6 30 7 13	2 2 4 5
		3	10 2 4 1 14 11 27	6 6 4 6 2 3 2					
		4	10 13 4 1 14 11 22	6 2 3 2 6 6 6					
54	1	1	1 4 24 13 16 29 30 24 15 10	5 5 6 5 5 3 2 2 3 3					

REFERENCES

Aarts, E.H.L., and Van Laarhoven, P.J.M., 1985, Statistical cooling: approach to combinatorial optimization problems. Philips Journal of Research, 40, 193–226.

Adenso-Diaz, B., Lozano, S., Racero, J. and Guerrero, F., 2001. Machine cell formation in generalized group technology. Computers and Industrial Engineering, 41. 227–240.

Ahi, A., Aryanezhad, M.B., Ashtiani, B. and Makui, A., 2009. A novel approach to determine cell formation, intracellular machine layout and cell layout in the CMS problem based on TOPSIS method. Computers and Operations Research, 36(5), 1478–1496.

Akturk, M.S. and Turkcan, A., 2000. Cellular manufacturing system design using a holonistic approach. International Journal of Production Research, 38(10), 2327–2347.

Alfa, S., Chen M. and Heragu S.S., 1992. Integrating the grouping and layout problems in cellular manufacturing systems. Computer and Industrial Engineering, 23 , 55–58.

Andres, C. and Lozano S., 2006. A Particle Swarm Optimization Algorithm for Part-machine Grouping. Robotics and Computer-Integrated Manufacturing, 22, 468–474.

Aneke, N.A.G. and Carrie, A.S., 1986. A design technique for the layout of multi-product flowlines. International Journal of Production Research, 24(3), 471–481.

Arkat, J., Saidi, M. and Abbasi, B., 2007. Applying Simulated Annealing to Cellular Manufacturing System Design. International Journal of Advanced Manufacturing Technology, 32(6), 531–536.

Askin, R.G. and Subramanian, S., 1987, A cost-based heuristic for group technology configuration. International Journal of Production Research, 25, 101–113.

Ballakur, A. & H.J. Steudel. (1987). A within cell utilization based heuristic for designing cellular manufacturing systems. International Journal of Production Research, 25, 639–655.

Bazargan-Lari, M., Kaebernick, H. and Harraf, A., 2000. Cell formation and layout designs in

a cellular manufacturing environment—a case study. International Journal of Production Research, 38, 1689–1709.

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

Boe, W., Cheng, C.H., 1991. A close neighbor algorithm for designing cellular manufacturing systems. International Journal of Production Research, 29(10), 2097–2116.

Bonomi, E., and Lutton, J., 1984. The N-city traveling salesman problems: statistical mechanics and the metropolis algorithm. SIAM Review, 26, 551–568.

Boulif, M. and Atif, K., 2006. A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. Computers and Operations Research, 33(8), 2219–2245.

Burbidge, J.L., 1975. The Introduction of Group Technology. New York: Wiley.

Carrie, S., 1973. Numerical taxonomy applied to group technology and plant layout. International Journal of Production Research, 11(4), 399–416.

Chan, F.T.S., Lau, K.W., Chan, L.Y., and Lo, V.H.Y., 2008. Cell formation problem with consideration of both intracellular and intercellular movements. International Journal of Production Research, 46(10), 2589–2620.

Chan, F.T.S., Lau, K.W., Chan, P.L.Y. and Choy, K.L., 2006. Two-stage approach for machine-part grouping and cell layout problems. Robotics and Computer-Integrated Manufacturing, 22 (3), 217–238.

Chan, H.M., Milner, D.A., 1982. Direct clustering algorithm for group formation in cellular manufacture. Journal of Manufacturing System, 1 (1), 65–75.

Chandrasekharan, M.P. and Rajagopalan, R., 1989. Groupability: an analysis of the properties of binary data matrices for group technology. International Journal of Production Research, 27 (6), 1035–1052.

Chandrasekharan, M.P., Rajagopalan, R., 1987. ZODIAC - An algorithm for concurrent

formation of part families and machine cells. International Journal of Production Research. 25 (6), 835–850.

Chandrashekharan, M.P. and Rajagopalan, R., 1986a. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. International Journal of Production Research, 24 (2), 451–464.

Chandrashekharan, M.P. and Rajagopalan, R., 1986b. MODROC: An extension of rank order clustering for group technology. International Journal of Production Research, 24 (5), 1221–1233.

Cheng, C. H., Gupta, Y. P., Lee, W. H., and Wong, K. F., 1998. A TSP-based heuristic for forming machine groups and part families. International Journal of Production Research, 36 (5), 1325–1337.

Chiang, C.P. and Lee, S.D., 2004. A genetic-based algorithm with the optimal partition approach for the cell formation in bi-directional linear flow layout. International Journal of Computer Integrated Manufacturing, 17 (4), 364–375.

Dougherty, D.E. and Marryott, R.A., 1991. Optimal groundwater management. Water Resources Research, 27, 2493–2508.

Glover F., 1986. Future paths for integer programming and links with artificial intelligence. Computers and Operations Research, 13 (5), 533–549.

Glover, F. and Laguna, M., 1997. Tabu search. Kluwer Academic Publishers, Boston.

Glover, F., 1989. Tabu search—Part I. ORSA Journal on Computing, 1 (3), 190–206.

Glover, F., 1990. Tabu search—Part II. ORSA Journal on Computing 2 (1), 4–32.

Glover, F., and Languna, M., 1993. Tabu Search – Modern heuristic techniques for combinatorial problems. Colin R. Reeves (Ed.), 70–150, Blackwell Scientific Publications, Oxford.

Glover, F., McMillan, C., and Novick, B., 1985. Interactive Decision Software and Computer Graphics for Architectural and Space Planning. Annals of Operations Research, 5,

557–573.

Gonçalves, J.F. and Resende, M.G.C., 2004, An evolutionary algorithm for manufacturing cell formation. Computers and Industrial Engineering, 47, 247–273.

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

Harhalakis, G., Nagi, R., and Proth, J. M., 1990. An efficient heuristic in manufacturing cell formation for group technology applications. International Journal of Production Research, 28, 185–198.

Ho, Y.C., Lee, C.E.C., and Moodie, C.L., 1993. Two sequence-pattern, matching-based, flow analysis methods for multi-flow lines layout design. International Journal of Production Research, 31, 1557–1578.

Hsu, C., 1990. Similarity Coefficient Approaches to Machine Component Cell Formation in Cellular Manufacturing: A Comparative Study. University of Wisconsin, Milwaukee.

Hu, L. and Yasuda, K., 2006. Minimising material handling cost in cell formation with alternative processing routes by grouping genetic algorithm. International Journal of Production Research, 44 (11), 2133–2167.

Jabal Ameli, M. S. and Arkat, J., 2008. Cell formation with alternative process routings and machine reliability consideration. International Journal of Advanced Manufacturing Technology, 35 (7-8), 761–768.

Jabal Ameli, M. S., Arkat, J. and Barzinpour, F., 2008. Modelling the effects of machine breakdowns in the generalized cell formation problem. International Journal of Advanced Manufacturing Technology, 39 (7–8), 838–850.

Jaccard, P., 1908. Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. Sci. Nat., 44, 223–270.

Jeon, G., M. Broering, H.R. Leep, H.R. Parsaei and Wong, J.P., 1998. Part family formation

based on alternative routes during machine failure. Computers and Industrial Engineering, 35 (1–2), 73–76.

Kao, Y. and Fu, S.C., 2006. An ant-based clustering algorithm for manufacturing cell design. International Journal of Advanced Manufacturing Technology, 28 (11–12), 1182–1189.

Kim, C.-O., Baek J.-G. and Baek J.-K., 2004. A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences. International Journal of Production Research, 42 (18), 3911–3927.

King, J.R. and Nakornchai, V., 1982. Machine-component group formation in group technology: Review and extension. International Journal of Production Research, 20 (2), 117–133.

King, J.R., 1980. Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. International Journal of Production Research, 18 (2), 213–232.

Kirkpatrick, S., Gelatt, C. and Vecchi, M., 1983. Optimization by simulated annealing. Science, 220 (4598), 671–680.

Kumar, C.S. and Chandrasekharan, M.P., 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, K.R. and Vannelli, A., 1987. Strategic subcontracting for efficient disaggregated manufacturing. International Journal of Production Research, 25 (12), 1715–1728.

Kumar, K.R., Kusiak, A. and Vannelli, A., 1986. Grouping of parts and components in flexible manufacturing systems. European Journal of Operations Research, 24 (3), 387–397.

Kusiak, A. and Cho, M., 1992. Similarity coefficient algorithm for solving the group technology problem. International Journal of Production Research, 30, 2633–2646.

Kusiak, A. and Chow, W., 1987. Efficient solving of the group technology problem. Journal of Manufacturing Systems, 6 (2), 117–124.

Kusiak, A., 1990. Intelligent Manufacturing Systems, Englewood Cliffs, NJ: Prentice Hall.

Lee, M.K., Luong, H.S. and Abhary, K., 1997. A genetic algorithm based cell design considering alternative routing. Computer Integrated Manufacturing Systems, 10 (2), 93–108.

Lei, D. and Wu, Z., 2005. Tabu search-based approach to multi-objective machine-part cell formation. International Journal of Production Research, 43 (24), 5241–5252.

Lin, F.T., Kao, C.Y., and Hsu, C.C., 1993. Apply the genetic approach to simulated annealing in solving some NP-hard problem. IEEE Transactions on System, Man, and Cybernetics, 23, 1752–1767.

Lundy, M., and Mees, A., 1986. Convergence of an annealing algorithm. Mathematical Programming, 34, 111–124.

Mahdavi I., Paydar M. M., Solimanpur M., and Heidarzade A., 2009. Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. Expert Systems with Applications, 36 (3), 6598–6604.

Mahdavi, I. and Mahadevan, B., 2008. CLASS: An algorithm for cellular manufacturing system and layout design using sequence data. Robotics and Computer-Integrated Manufacturing, 24 (3), 488–497.

McAuley, J., 1972. Machine grouping for efficient production. Production Engineering, 51 (2), 53–57.

McCormick, W. T., Schweitzer, P. J. and White, T. W., 1972. Problem decomposition and data reorganization by a clustering technique. Operations Research, 20, 993–1009.

Mckendall, A.R., Shang, J. and Kuppusamy, S., 2006. Simulated annealing heuristics for the dynamic facility layout problem. Computers & Operations Research, 33, 2431–2444.

Metropolis, N., Rosenbluth, A. W. and Teller, A. H., 1953. Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21 (6), 1087–1092.

Miltenburg, J., Zhang, W., 1991. A comparative evaluation of nine well-known algorithms for

solving the cell formation problem in group technology. Journal of Operations Management, 10 (1), 44–72.

Mitrovanov, S.P., 1966. The Scientific Principles of Group Technology. National Lending Library Translation, Boston Spa, Yorkshire.

Mosier, C.T. and Taube, L., 1985a. The facets of group technology and their impact on implementation. OMEGA, 13 (6), 381–391.

Mosier, C.T. and Taube, L., 1985b. Weighted similarity measure heuristics for the group technology machine clustering problem. OMEGA, 13 (6), 577–583.

Nair G.J., Narendran T.T., 1998. CASE: a clustering algorithm for cell formation with sequence data. International Journal of Production Research, 1998, 36, 157–79.

Nair, G.J.K., Narendran, T.T., 1996. Grouping index: A new quantitative criterion for goodness of block diagonal forms in group technology. International Journal of Production Research, 34 (10), 2767-2782.

Ng, S.M., 1993. Worst-case analysis of an algorithm for cellular manufacturing. European Journal of Operational Research, 69 (2), 384-398.

Onwubolu, G.C. and Mutingi, M., 2001. A genetic algorithm approach to cellular manufacturing systems. Computers and Industrial Engineering, 39 (1–2), 125–144.

Park, S. and Suresh, N.C., 2003. Performance of Fuzzy ART neural network and hierarchical clustering for part-machine grouping based on operation sequences. International Journal of Production Research, 41 (14), 3185–3216.

Qiao, L.H., Yang, Z.B. and Wang, H.P., 1994. A computer-aided process planning methodology. Computers in Industry, 255, 83–94.

Romesburg, H.C., 1984. Cluster Analysis for Researchers. Lifetime Learning Publications, Wadsworth, Inc., Belmont, CA.

Sarker, B.R., 2001. Measures of grouping efficiency in cellular manufacturing systems. European Journal of Operational Research, 130 (3), 588–611.

Sarker, B.R., Li, Z., 1998. Measuring matrix-based cell formation with alternative routings. Journal of the Operational Research Society, 49 (9), 953-965.

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 Tjahjana, B., 1999. Part-family formation for cellular manufacturing: a case study at Harnischfeger. International Journal of Production Research, 37 (14), 3263–3273.

Seifoddini, H. and Wolfe, P. M., 1986. Application of the similarity coefficient method in group technology. IIE Transactions, 18 (3), 266–270.

Seifoddini, H., 1989. Single linkage versus average linkage clustering in machine cells formation applications. Computers and Industrial Engineering, 16 (3), 419–426.

Selim S. Z. and Alsultan K., 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognition, 24 (10), 1003–1008.

Sofianopoulou, S., 1999. Manufacturing cells design with alternative process plans and/or replicate machines. International Journal of Production Research, 37 (3), 707–720.

Srinivasan, G., Narendran, T., Mahadevan, B., 1990. An assignment model for part-families problem in group technology. International Journal of Production Research, 28 (1), 145–152.

Stanfel, L., 1985. Machine clustering for economic production. Engineering Costs and Production Economics, 9, 73–78.

Su, C.T. and Hsu, C.M., 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 and Industrial Engineering, 1995, 28, 485–494.

Tam, K.Y., and Li, S.G., 1991. A hierarchical approach to the facility layout problem. International Journal of Production Research, 29 (1), 165–184.

Tariq, A., Hussain, I. and Ghafoor, A., 2009. A hybrid genetic algorithm for machine-part grouping. Computers and Industrial Engineering, 56 (1), 347–356.

Unler A and Gungor Z., 2009. Applying K-harmonic means clustering to the part-machine classification problem. Expert Systems with Applications, 36 (2), 1179–1194.

Waghodekar, P. H. and Sahu, S., 1984. Machine-component cell formation in group technology MACE. International Journal of Production Research, 22, 937–948.

Wemmerlov U. and Hyer N.L., 1986. Procedures for the part family/machine group identification problem in cellular manufacture. Journal of Operations Management, 6, 125–147.

Wemmerlov, U. and Hyer, N., 1987. Research issues in cellular manufacturing. International Journal of Production Research, 25 (3), 413–431.

Won, Y.K. and Kim, S.H., 1997. Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings. Computers and Industrial Engineering, 32 (1), 207–220.

Wu, T.-H., Low, C. and Wu, W.-T., 2004. A tabu search approach to the cell formation problem. International Journal of Advanced Manufacturing Technology, 23, 916–924.

Wu, X., Chu, C.H., Wang, Y. and Yan, W., 2007a. A genetic algorithm for cellular manufacturing design and layout. European Journal of Operational Research, 181, 156–167.

Wu, X., Chu, C.H., Wang, Y. and Yue, D., 2007b. Genetic algorithms for integrating cell formation with machine layout and scheduling. Computers and Industrial Engineering, 53, 227–289.

Yang, F.-C. and Wang, Y.-P., 2007. Water flow-like algorithm for object grouping problems. Journal of the Chinese Institute of Industrial Engineers, 24 (6), 475–488.

Yang, M.S. and Yang, J.H., 2008. Machine-part cell formation in group technology using a modified ART1 method. European Journal of Operational Research, 188 (1), 140–152.

Yin Y. and Yasuda K., 2005. Similarity coefficient methods applied to the cell formation problem: a comparative investigation. Computers and Industrial Engineering, 48 (3), 471–489.

Yu, V.F., Lin, S.W., Lee, W. and Ting, C.J., 2010. A simulated annealing heuristic for the capacitated location routing problem. Computers and Industrial Engineering, 58, 288–299.

作者簡介(Biography)

姓名(Name)：張欽智(Chang, Chin-Chih)

學歷：

高中	嘉義高中		77/9 ~ 80/6
大學	大葉大學	工業工程	80/9 ~ 84/6
碩士	大葉大學	工業工程	84/9 ~ 86/6
博士	交通大學	工業工程與管理學系	94/9 ~ 目前

經歷：

志合電腦	IE 工程師	89/3 ~ 91/2
仁德醫護管理專科學校	資管科專任講師	91/2 ~ 目前

著作：

一、已接受之期刊論文

1. Wu, T.-H., **Chang, C.-C.** and Chung, S.-H., 2008. "A simulated annealing algorithm for manufacturing cell formation problems," *Expert Systems with Applications*, 34 (3), 1609–1617(SCI).
2. Wu, T.-H., Chung, S.-H. and **Chang, C.-C.**, 2009. "Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings," *Expert Systems with Applications*, 36 (2), 3652–3661(SCI).
3. Wu, T.-H., Chung, S.-H. and **Chang, C.-C.**, 2010. "A water flow-like algorithm for manufacturing cell formation problems," *European Journal of Operational Research*, 205 (2), 346–360 (SCI).
4. Chung, S.-H., Wu, T.-H. and **Chang, C.-C.**, 2010. "An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations," *Computers & Industrial Engineering*, 60 (1), 7–15(SCI).

二、研討會論文

1. T-H Wu and **C-C Chang**, 2006, "A Simulated Annealing Approach to Cellular Manufacturing Systems", *The 11th International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE2006)*, 378–383 , Nagoya, JAPAN.
2. **Chin-Chih Chang**, Tai-Hsi Wu and Shu-Hsing Chung, 2009. "A novel approach for cell formation and cell layout design in cellular manufacturing system." *Special track within EMS2009: Information systems and Management Track (ISM 2009)*, Beijing, China (EI).