Computers & Industrial Engineering 60 (2011) 7–15

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations ☆

Shu-Hsing Chung [a], Tai-Hsi Wu [b,*], Chin-Chih Chang [a]

[a] *Department of Industrial Engineering and Management, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan*
[b] *Department of Business Administration, National Taipei University, 151, University Road, San Shia, Taipei 237, Taiwan*

ARTICLE INFO

Article history:
Received 13 November 2008
Received in revised form 25 August 2010
Accepted 25 August 2010
Available online 31 August 2010

Keywords:
Cell formation
Alternative process routings
Machine reliability
Tabu search
Mutation operator

ABSTRACT

Cell formation is the first step in the design of cellular manufacturing systems. In this study, an efficient tabu search algorithm based on a similarity coefficient is proposed to solve the cell formation problem with alternative process routings and machine reliability considerations. In the proposed algorithm, good initial solutions are first generated and later on improved by a tabu search algorithm combining the mutation operator and an effective neighborhood solution searching mechanism. Computational experiences from test problems show that the proposed approach is extremely effective and efficient. When compared with the mathematical programming approach which took three hours to solve problems, the proposed algorithm is able to produce optimal solutions in less than 2 s.

© 2010 Published by Elsevier Ltd.

1. Introduction

Cellular manufacturing is the implementation of Group Technology (GT), a manufacturing philosophy in which similar parts are identified and grouped into part families; meanwhile, machines are grouped into machine cells to take advantage of their similarities in manufacturing and design. GT was originally introduced by Mitrovanov (1966) and was popularized in the west by Burbidge (1975). The implementation of cellular manufacturing has been reported to result in significant benefits such as reductions in set-up times, work-in-progress inventory, throughput times and material handling costs, simplified scheduling and improved quality (Wemmerlov & Hyer, 1987).

Although cellular manufacturing may provide great benefits, the design of cellular manufacturing systems (CMS) is complex for real life problems. It has been known that the cell formation problem (CFP) in CMS is one of the NP-hard combinational problems (Ballakur & Steudel, 1987). Many models and solution approaches have been developed to identify machine cells and part families, as it becomes difficult to obtain optimal solutions in an acceptable amount of time, especially for large-sized problems. These approaches can be classified into three main categories: mathematical programming (MP) models (e.g., Albadawi, Bashir, & Chen, 2005; Boctor, 1991; Kumar, Kusiak, & Vannelli, 1986; Lozano, Adenso-Diaz, & Onieva, 1999; Lozano, Guerrero, Eguia, & Onieva, 1999; Srinivasan, Narendran, & Mahadevan, 1990; Wang, 2003), heuristic/meta-heuristic solution algorithms (e.g., Diaz, Lozano, Racero, & Guerrero, 2001; Lei & Wu, 2005; Sofianopoulou, 1999; Sun, Lin, & Batta, 1995; Wu, Chang, & Chung, 2008; Wu, Chung, & Chang, 2008; Wu, Low, & Wu, 2004), and similarity coefficient methods (SCM) (e.g., Alhourani & Seifoddini, 2007; McAuley, 1972; Nair & Narendran, 1998; Yin & Yasuda, 2006).

Most of the above CF researches assume that each part has a unique process routing. However, it is well known that alternatives may exist in any level of a process plan. When each part has alternative process routings (APR), the CFP becomes the generalized CFP (Kusiak, 1987). Explicit consideration of APR may result in additional flexibility in the CMS design.

Over the past four decades, the machine-part cell formation problem has been the subject of numerous studies. Many researchers have applied various methodologies in an effort to determine the optimal clustering of machines and the optimal groupings of parts into families. However, only a limited amount of research in the context of the CFP has dealt with machine breakdowns or reliability issues (e.g., Das, Lashkari, & Sengupta, 2007; Diallo, Perreval, & Quillot, 2001; Jabal Ameli & Arkat, 2008; Jabal Ameli, Arkat, & Barzinpour, 2008; Logendran & Talkington, 1997; Savsar, 2000; Zakarian & Kusiak, 1997). Traditionally, CF and work allocation are performed, assuming that all the machines are 100% reliable. However, this is not always the case. Machines are key elements in manufacturing systems and oftentimes it is not

☆ This manuscript was processed by Area Editor Gursel A. Suer.

* Corresponding author. Tel.: +886 2 86746574; fax: +886 2 86715912.
E-mail addresses: shchung@mail.nctu.edu.tw (S.-H. Chung), taiwu@mail.ntpu.edu.tw (T.-H. Wu), chinju.chang@gmail.com (C.-C. Chang).

0360-8352/$ - see front matter © 2010 Published by Elsevier Ltd.
doi:10.1016/j.cie.2010.08.016

Nomenclature

m	number of machines
M	machines set
p	number of parts
P	parts set
NC	number of cells
C	cells set
V_i	production volume for part i
Q_i	number of routings for part i
U_m	maximum number of machines in each cell
L_m	minimum number of machines in each cell
A_i	unit cost of intercellular movement for part i
K_{ij}	number of operations in routing j of part i the operations of part i along route j are processed on a machines' set of $\left\{u_{ij}^{(1)}, u_{ij}^{(2)}, \ldots, u_{ij}^{(k)}, u_{ij}^{(k+1)}, \ldots, u_{ij}^{(K_{ij}-1)}, u_{ij}^{(K_{ij})}\right\}$
$u_{ij}^{(k)}$	machine index for the kth operation of part i along route j
$T_{ij}^{(k)}$	processing time for the kth operation of part i along route j
B_k	breakdown cost for machine k
MTBF_k	mean time between failures for machine k
$u_{ij}^{(K_{ij})}$	Machine's index in routing j of part i
Y_{kl}	1, if machine k locates in cell l; 0, otherwise
Z_{ij}	1, if routing j of part i selected; 0, otherwise
X_{ijklsl}	1, if routing j of part i is selected; machine k locates in cell l and machine s do not locate in cell l; 0, otherwise

possible to handle their breakdowns as quickly as the production requirements dictate. Their breakdowns can dramatically affect system performance measures and bring about detrimental effects on the due date performance. Machine failures should hence be taken into account during the design of CMS to improve the overall performance of the system (Jeon, Broering, Leep, Parsaei, & Wong, 1998). The machine breakdown cost generally consists of machine repairing costs, production suspension costs, and capacity lost costs, etc. (Jabal Ameli & Arkat, 2008).

Jabal Ameli and Arkat (2008) formulated a binary integer programming model for the CFP accounting for APR and the machine reliability issue. They solved the model using a mathematical programming approach. However, since CFP is an NP-hard problem, it usually takes a large amount of computational efforts to solve the problem when classical optimization methods are used, especially for large-sized problems. Thus, there is a need to develop an efficient and effective solution approach capable of handling this problem.

Due to their excellent performances in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic algorithm (GA), simulated annealing (SA), neural network (NN) and tabu search (TS) are grouped into another class of search methods that have been adapted to solve the CF problem and its variants efficiently. Among the aforementioned meta-heuristic algorithms, TS has been successfully used to solve many problems appeared in manufacturing system including cell formation problems (Lozano, Adenso-Diaz et al., 1999; Lozano, Guerrero et al., 1999). TS uses flexible memory structures to store information and attributes of solutions from the recent history of the search. TS gives some recently or frequently visited solutions (moves) a tabu restriction to keep the solution search process from being trapped at a local optimum.

The mutation operator of the genetic algorithm (GA) is another well-known technique, famous for its capability to escape from local solutions and prevent premature convergence. It is used mainly to increase the diversity of the population and to ensure that an extensive search will be performed.

This study anticipates the synergy effects between the TS and the GA by presenting an efficient algorithm using the TS, together with the mutation operator from the GA, to increase the quality and efficiency of solutions.

The remainder of this article is organized as follows: Section 2 describes the problem definition including the CFP with alternative routings and the issues of machine reliability. The mathematical model presented by Jabal Ameli and Arkat (2008) for solving the problem is given and reviewed in this section as well. Section 3 details the proposed TS algorithm including the generation of initial solutions and solution improvement procedures. Computational results on test problems are reported in Section 4. Section 5 concludes the paper.

2. Problem definition

This section describes the problem definition of CFP accounting for APR and the machine reliability issue. A 0–1 integer programming model formulated by Jabal Ameli and Arkat (2008) for solving this complicated problem is introduced as well.

2.1. Cell formation problem with alternative process routings

In a simple CFP, cell formation in a given 0–1 machine-part incidence matrix involves the rearrangement of its rows and columns to create part families and machine cells. Researches usually attempt to determine a rearrangement in which the intercellular movement can be minimized and the utilization of the machines within a cell maximized. After the rearrangement, blocks can be observed along the diagonal of the matrix. In the matrix, any 1s outside the diagonal blocks are called "exceptional elements"; any 0s inside the diagonal blocks are called "voids".

Cases in which each part may have more than one process routings are more complicated than the simple CFP. A process routing for a given part is a set of machines that have passed by this part. It is assumed that the sequence of machines in each process routing is identical with the operation sequence of the corresponding part. When parts are allowed to have more than one process routing, such as the case shown in Table 1, the CFP becomes generalized, wherein cases are more complicated than the simple cell formation problem. Under this circumstance, the formation of part families, machine cells, and selection of routings for each part need to be determined to achieve the decision objectives, such as the minimization of intercellular movement or the maximization of grouping efficacy.

2.2. Machine reliability

Machines are key elements in manufacturing systems. Thus, machine reliability should be taken into account during the design of the CMS. The reliability of a machine is defined as $R = \exp(\lambda t)$, where λ is the machine failure rate and t is the machine operating time. A common way of dealing with machines reliability concern in the design phase of a manufacturing system is by the evaluation of the quantities of the mean time between failures (MTBF). MTBF

Table 1
APR and processing times for the numerical example (Bhide, Bhandwale, & Kesavadas, 2005).

PV	75			130			110		145		110		105		140				115	
PN	P1			P2			P3		P4		P5		P6		P7				P8	
RN	R1	R2	R3	R1	R2	R3	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R3	R4	R1	R2
M1	*1(5)			1(4)			1(4)	1(4)	1(5)	1(5)		1(4)		1(4)	1(5)			1(5)	1(4)	
M2		1(5)	1(5)		1(5)	1(5)					1(3)		1(5)			1(3)	1(3)	2(3)		1(4)
M3							2(3)	2(3)	2(3)	2(3)										
M4	2(3)			2(4)				3(3)	3(5)	3(5)		2(4)			2(4)		2(4)			
M5	3(4)		2(4)	3(3)	2(3)	2(3)	3(4)			4(4)					3(3)		3(3)			
M6		2(5)														2(4)		3(4)	2(3)	2(4)
M7											2(5)	3(5)	2(5)	2(5)						
M8		3(4)		4(4)		3(4)	4(3)	4(3)	4(4)	5(4)			3(5)	3(5)			4(5)			
M9	4(5)		3(5)		3(3)						3(5)	4(5)			4(5)	3(5)		4(5)		

PV: Production Volume; PN: Part Number; RN: Routing Number.
[*] Production sequence (Production times).

can be obtained by taking the reciprocal of λ. 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 the machine breakdown cost. This is achieved by dividing the production time by the MTBF and then multiplying this quantity by the unit machine breakdown cost. The decision objective of their research is to minimize the sum of total intercellular movement cost and the machine breakdown cost. The 0–1 integer programming model that they formulated is given below, and the notations are introduced first.

Intercellular movement cost:

$$Inter\_C = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{k=1}^{K_{ij}-1}\sum_{l=1}^{NC} A_i V_i X_{ij\left(u_{ij}^{(k)}\right)l\left(u_{ij}^{(k+1)}\right)l} \quad (1)$$

Machine breakdown cost:

$$\mathrm{TBC} = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{k=1}^{K_{ij}} Z_{ij}\frac{V_i T_{ij}^{(k)} B_{\left(u_{ij}^{(k)}\right)}}{\mathrm{MTBF}_{\left(u_{ij}^{(k)}\right)}} \quad (2)$$

The 0–1 integer programming model is as follows:

$$\text{Min} \quad TC = Inter\_C + \mathrm{TBC} \quad (3)$$

s.t.

$$\sum_{j=1}^{Q_i} Z_{ij} = 1, \quad \forall i \in P \quad (4)$$

$$L_m \leqslant \sum_{k=1}^{m} Y_{kl} \leqslant U_m, \quad \forall l \in C \quad (5)$$

$$\sum_{l=1}^{NC} Y_{kl} = 1, \quad \forall k \in M \quad (6)$$

$$X_{ijklsl} \leqslant Z_{ij}, \quad \forall i,j,k,l,s \quad (7)$$

$$X_{ijklsl} \leqslant Y_{kl}, \quad \forall i,j,k,l,s \quad (8)$$

$$X_{ijklsl} \leqslant (1 - Y_{sl}), \quad \forall i,j,k,l,s \quad (9)$$

$$Z_{ij} + Y_{kl} + (1 - Y_{sl}) - X_{ijklsl} \leqslant 2, \quad \forall i,j,k,l,s \quad (10)$$

$$Y_{kl},\ Z_{ij},\ X_{ijklsl} \in \{0,1\} \quad \forall i,j,k,l,s \quad (11)$$

In the above model, Eqs. (1) and (2) show the calculation of the total intercellular movement cost and the machine breakdown cost, respectively. Eq. (3) is the objective function, which seeks the minimization of total cost of intercellular movement and machine breakdown. Eq. (4) indicates that only one process routing will be assigned to each part. Eq. (5) assigns the upper and lower limits of the cell size. Eq. (6) provides a restriction that each machine will be assigned to exactly one cell. Eqs. (7)–(9) ensure that if one of the primary binary variables takes has a zero value, then their corresponding new variables will take a zero value as well. Eq. (10) ensures that if all of the primary variables take unit values, then their corresponding new variables take unit values as well. Eq. (11) indicates that Y_{kl}, Z_{ij} and X_{ijklsl} are 0–1 binary decision variables.

Although the original objective function has been transformed into linear form, which makes several linear programming software empowered to solve this model, the large number of binary variables and constraints, makes it difficult to obtain optimal solutions when the problem sizes increase. Developing good heuristic approaches is more appropriate than using the exact method in terms of solution efficiency, especially for large-sized problems. This paper, thus, presents an efficient tabu search algorithm possessing the features of both the tabu search and the genetic algorithm and is collaborated with the delicate design of neighborhood solution searching. The proposed algorithm is described and explained in detail in the next section.

3. Proposed tabu search (TS) algorithm

When designing a heuristic search algorithm, there are several important things to keep in mind. The first is to develop a mechanism for searching the neighborhood solutions for improvement. Since the neighborhood is what will be searched next, the choice of the neighborhood function will strongly influence the direction that the search takes. Another item that needs to be considered is the mechanism for allowing the technique to escape local optima and settle only on a global optimum.

TS is a meta-heuristic algorithm developed by Glover which has been successfully used to generate solutions for a wide variety of combinatorial problems. The main ideas of TS are to avoid recently visited area of the solution space and to guide the search towards new and promising areas. Non-improving moves are allowed to escape from the local optima, and attributes of recently performed moves are declared tabu or forbidden for a number of iterations to avoid cycling. For more details about the tabu search methodology, see Glover (1989) and Glover (1990).

In GA, some initial solutions are selected to be parents to generate offspring via the crossover operator. All the solutions are then evaluated and selected based on Darwin's concept of survival of the fittest. The process of reproduction, evaluation, and selection is repeated until the stopping criterion is met. In GA, the mutation operator is usually applied to solutions at hand with a certain probability to escape from local solutions and/or to prevent premature convergence. This special feature of the mutation operator provides a higher degree of diversification in the solution searching process. It is expected that the synergy effects from both the TS and the GA can be appreciated through a proper collocation of both techniques.

The proposed TS procedure consists of two stages: the initial solution construction and the improvements stage. The similarity coefficient-based method (SCM) is adapted in the first stage to produce good initial solutions, while the TS continuously improves and generates more effective solutions through the neighborhood moves in the second stage. The details of these procedures are given below.

3.1. Initial solution construction

When generating the initial solutions, the SCM-based procedure follows three steps: (1) formation of machine cells; (2) selection of routings for each part; and (3) formation of part families.

3.1.1. Formation of machine cells

The part-based SCM of Kusiak and Cho (1992) and the machine-based SCM of Won and Kim (1997) are the two most widely used generalized similarity coefficient methods for considering alternative process routings. They are an extension of the Single Linkage Clustering Algorithm (SLCA) of McAuley (1972). Compared to the machine-based, the part-based SCM suffers from a computational burden since the number of parts in a cell formation problem is usually much greater than the number of machines. The machine-based SCM is adapted in this research.

According to Seifoddini and Djassemi (1995), incorporation of production volume into the similarity measures increases the chance 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. Won and Kim's machine-based SCM is thus modified to incorporate the production volume information. Consider a specific machine-part matrix with alternative routings and the information of production volume, the corresponding similarity matrix for machines can be obtained by using the following formula:

$$S_{ij} = \frac{N_{ij}}{N_i + N_j - N_{ij}} \tag{12}$$

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

After calculating the similarity matrix for each pair of machines, the initial machines assignment is generated by using the following rule: the higher similarity measure a pair of machines has, they should be placed in the same cell with higher priority. This process is repeated until all machines have been assigned to cells. Consider the numerical example with alternative process routings in Table 1. The corresponding similarity matrix for machines can be obtained by using Eq. (12) and is listed in Table 2.

Suppose there are two cells to be formed. The largest coefficient in the similarity matrix of Table 3 is 0.72, indicating that machines 2 and 6 must be assigned to cell 1. The second largest coefficient in the matrix, 0.70, appears in pair (4, 3). Because machines 3 and 4 have not been assigned to any cell, they are assigned to cell 2. Pair (8, 5) is the next choice. When determining to which cell machine 5 should be assigned, the similarity coefficients of machine 5 with machines in each cell are examined respectively. For cell 1, the largest similarity coefficient with machine 5 appears in machine 2, which is equal to 0.34. For cell 2, the largest similarity coefficient with machine 5 appears in machine 4, which is equal to 0.48. Machine 5 is thus assigned to cell 2 together with machines 3 and 4. By repeating the same logic, it can finally be determined that machines 2, 6 and 7 should be assigned to cell 1; while machines 1, 3, 4, 5, 8 and 9 are assigned to cell 2. The sample problem displayed in Table 1 is thus rearranged as shown in Fig. 1.

3.1.2. Selection of routings for each part

The next task deals with assigning a routing for each part after the machine cells have been obtained. The part routings are

Table 2
Similarity matrix for machines in numerical example.

Machine	1	2	3	4	5	6	7	8	9
1	–								
2	0.15	–							
3	0.42	0.00	–						
4	0.18	0.17	0.70	–					
5	0.00	0.34	0.24	0.48	–				
6	0.16	0.72	0.00	0.00	0.00	–			
7	0.17	0.32	0.00	0.23	0.00	0.00	–		
8	0.00	0.00	0.00	0.43	0.68	0.00	0.18	–	
9	0.00	0.00	0.00	0.00	0.58	0.32	0.26	0.00	–

Table 3
Machine reliability information for the numerical example.

Machine	Breakdown cost	MTBF (min)
M1	900	5400
M2	2000	3060
M3	2000	4380
M4	1600	3600
M5	1500	4560
M6	1800	3720
M7	1400	4260
M8	1700	3480
M9	1500	3900

PV	75			130			110		145		110		105		140				115	
PN	P1			P2			P3		P4		P5		P6		P7				P8	
RN	R1	R2	R3	R1	R2	R3	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R3	R4	R1	R2
M2		1	1		1	1					1		1			1	1	2		1
M6		2														2		3	2	2
M7											2	3	2	2						
M1	1			1			1	1	1	1		1		1	1			1	1	
M3							2	2	2	2										
M4	2			2				3	3	3		2			2		2			
M5	3		2	3	2	2	3			4					3		3			
M8		3		4		3	4	4	4	5			3	3			4			
M9	4		3		3						3	4			4	3		4		

Fig. 1. Assignment of machines.

assigned to machine cells that would result in the least cost of intercellular movement and the machine breakdown. The unit intercellular movement for each trip made is assumed to be five, and machine breakdown information such as shown in Table 3 has to be given in order to calculate the cost. The part routing assignment procedure is described below:

Step 1: Read the results of 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 the intercellular movement cost (*Inter_C)* and the machine breakdown cost (TBC). If a tie happens, make a random selection.
Step 3: Repeat *Step 2* until the process routing has been determined for each part. Results of machines assignment shown in Fig. 2 are used to demonstrate the above procedure.

3.1.3. Formation of part families

Wu, Chang, et al. (2008) presented a simple procedure for assigning machines to manufacturing cells in which the number of voids and exceptional elements – major components comprising the formula of grouping efficacy – are explicitly considered. Instead of using this procedure for assigning machines to cells, their approach is adapted for assigning parts to cells in this study. The procedure is summarized as follows:

Step 1: Read the results of machines 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 number of exceptional elements and the 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 machines assignment and routings selection shown in Fig. 2 are used to demonstrate the above procedure. After calculating the sum of numbers of voids and exceptional elements for each part-cell combination, it can be observed in Fig. 3 that parts 5, 6 and 8 are assigned to cell 1, while parts 1–4 and 7 are assigned to cell 2. The initial solution matrix for this CF problem has been generated and shown in Fig. 3 with total intercellular movement cost 1075 and machine breakdown cost 5236.

PV	110	105	115	75	130	110	145	140
PN	P5	P6	P8	P1	P2	P3	P4	P7
RN	R1	R1	R2	R1	R1	R1	R1	R1
M2	1	1	1					
M6			2					
M7	2	2						
M1				1	1	1	1	1
M3						2	2	
M4				2	2		3	2
M5				3	3	3		3
M8		3			4	4	4	
M9	3			4				4
TBC	608	772	523	405	700	530	925	773
Inter_C	550	525	0	0	0	0	0	0
TC	1158	1297	523	405	700	530	925	773

Fig. 3. Initial solution matrix obtained.

3.2. Solution improvements

The initial solution generated in Section 3.1 is to be improved through the tabu search iteratively to produce more effective solutions. The elements comprising the proposed tabu search algorithm are described below.

3.2.1. Configuration

An easy way to represent a configuration of a feasible solution of the CF problem is a string, the size of which is equal to the number of machines, as shown in Fig. 4. In such a configuration, the *j*th bit of the string stores the identifier of the cell to which the machine is assigned. From the string (2, 1, 2, 2, 2, 1, 1, 2, 2), it is known that machines 2, 6, 7 are assigned to cell 1, while machines 1, 3, 4, 5, 8 and 9 are assigned to cell 2.

PV	75			130			110		145		110		105		140				115	
PN	P1			P2			P3		P4		P5		P6		P7				P8	
RN	R1	R2	R3	R1	R2	R3	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R3	R4	R1	R2
M2		1	1		1	1					1		1			1	1	2		1
M6		2														2		3	2	2
M7											2	3	2	2						
M1	1			1			1	1	1	1		1		1	1			1	1	
M3							2	2	2	2										
M4	2			2				3	3	3		2			2		2			
M5	3		2	3	2	2	3			4					3		3			
M8		3		4		3	4	4	4	5			3	3			4			
M9	4		3		3						3	4			4	3		4		
TBC	405	573	488	700	703	807	530	532	925	1116	608	661	772	499	773	815	1004	931	244	523
Inter_C	0	375	375	0	650	650	0	0	0	0	550	1100	525	1050	0	700	700	1400	575	0
TC	405	948	863	700	1353	1457	530	532	925	1116	1158	1761	1297	1549	773	1515	1704	2331	819	523

Fig. 2. Selection of routings.

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

Fig. 4. Configuration of a feasible solution to the CF problem.

3.2.2. Neighborhood solution searching

In this study, the neighborhood of a given solution is defined as the set of all feasible solutions reachable by an insertion-move. The insertion-move is an operation that moves a machine j from its current cell i (source cell) to a new cell i' (destination cell). The new move is denoted as (i', j). For the insertion-move, a move that results in the most improvement in the objective function value from the current solution is selected – that is:

$$Z(i',j) = Max\{obj^{(i',j)} - obj^{(i,j)}, \quad \forall i, i' \in N^F \quad \text{and} \quad (\notin N^T \text{ or } \in N^A), \quad i' \neq i, \ \forall j \in M\} \tag{13}$$

where N^F is the set of solutions satisfying the upper and lower limits of cell size; N^T is the set of tabu list; N^A is the set of solutions satisfying the aspiration criterion; M is the set for machines. The above formula implies that every possible move will be evaluated as long as it is not in tabu status and it satisfies the upper and lower limits of the cell size.

3.2.3. 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 ($m \times NC \times NC$) 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 j moves from its current cell i to a new cell i', then moving machine j from cell i' to cell i will be forbidden for a certain number of iterations, which is equal to the tabu list size (e.g. TL[j][i'][i] = tls). Previous studies have shown that the best tabu list size is between 5 and 12 in many applications, with 7 being the most recommended one (Glover, 1990). This suggestion is followed in this study.

3.2.4. 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. This aspiration criterion is applied in the proposed algorithm.

3.2.5. Machine mutation strategy

The mechanism of mutation aims at maintaining diversity in the population so that the large areas of the space are searched. In this study, when the number of moves has not been improved within a certain number of iterations, *mut_check*, the machine mutation strategy is implemented by reassigning a machine to any cells other than the current one based on a prescribed probability γ. That is, all machines are probable to change 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 γ, then the machine is assigned to another randomly determined cell; otherwise, it stays in the current cell. Note that when machines change cell, the resulting solution would not be accepted unless it satisfies the upper and lower limits of cell size. 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 machine mutation strategy in the pseudo-code format is shown in Fig. 5. The value of *mut_check* is determined by the formula, $m(NC - 1)/2$; while γ is set to 0.8 in this study.

```
// Machine mutation procedure
// m_i^* : Incumbent solution of machine i is assigned
// m_i : current solution of machine i is assigned
// CS(k) : number of machines in cell k

FOR each machine i
{
    Let m_i= m_i^*;
    IF CS(m_i)>L_m
    {
        Generate a random number u∈ U (0, 1) ;
        IF (u > γ)
        {
            Generate a cell number k {k∈ U (1, NC) and k ≠ m_i and CS(k)<U_m};
            Let m_i = k ;
        }
    }
}
```

Fig. 5. Machine mutation strategy.

3.2.6. Stopping criterion

The proposed solution procedure will be terminated if a maximum number of iterations N_{max} has been reached, or the solution has not been improved within a certain number of iterations *stag_check*. After intensive testing, the values of N_{max} and *stag_check* are set at 9000, 3000 (one third of N_{max}), respectively.

3.3. Proposed algorithm TSM

This section describes the proposed TS algorithm with mutation (TSM) in detail. It is evident that the number of cells to be formed will affect the grouping solutions obtained in the CF problem. Unlike many researches in literature where the number of cells to be formed is prescribed beforehand, the number of cells resulting in the best objective values will be automatically calculated and used in the proposed TSM. However, to preserve flexibility, users are permitted to specify the preferred number of cells when implementing the algorithm. Before explaining the solution procedure, notations in addition to those in Section 2.2 are introduced.

N_{max}	maximum 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
N^T	set of tabu list
N^A	set of solutions satisfying aspiration criterion
N^F	set of solutions satisfying the upper and lower limits of the cell size
m_0	initial solution of machines assignment
m_c	current solution of machines assignment
m'	neighborhood solution of machines assignment
m^*	incumbent solution of machines assignment of current cell size
m^{**}	best solution of machines assignment so far
p_0	initial solution of parts assignment
p_c	current solution of parts assignment
p'	neighborhood solution of parts assignment
p^*	incumbent solution of parts assignment of current cell size
p^{**}	best solution of parts assignment so far
r_0	initial solution of routings selection
r_c	current solution of routings selection
r'	neighborhood solution of routings selection
r^*	incumbent solution of routings selection of current cell size
r^{**}	best solution of routings selection so far

S_0 objective function value of initial cell configuration (m_0, p_0, r_0)
S_c objective function value of current cell configuration (m_c, p_c, r_c)
S' objective function value of neighborhood cell configuration (m', p', r')
S^* objective function value of incumbent cell configuration (m^*, p^*, r^*)
S^{**} objective function value of best cell configuration (m^{**}, p^{**}, r^{**})so far

The proposed algorithm TSM is described as follows:

Step 1: Set $NC = \lceil m/U_m \rceil$.
Step 2: Generate initial cell configuration (m_0, p_0, r_0) using the initial solution construction procedure in Section 3.1. Calculate the objective function value S_0.
Step 3: Initialization: Let $counter\_iter = 0$, $counter\_stag = 0$, $m_c \leftarrow m_0$, $p_c \leftarrow p_0$, $r_c \leftarrow r_0$, $S_c \leftarrow S_0$, $S^* \leftarrow S_0$, $N^T = \varnothing$.
Step 4: If $counter\_iter \leqslant N_{\max}$ and $counter\_stag \leqslant stag\_check$, repeat *Steps 5–10*; otherwise, go to *Step 11*.
Step 5: If $counter\_mut \geqslant mut\_check$, then apply the mutation operator, as mentioned in Section 3.2.5, to generate a new cell configuration (m_c, p_c, r_c) and let $counter\_mut = 0$.
Step 6: Search for a best neighborhood cell configuration $\{(m', p', r') | m' \in N^F \ and \ m' \notin N^T \ or \ m' \in N^A\}$ by performing the insertion-move. Calculate the objective function value S'.
Step 7: Update tabu list N^T.
Step 8: If $S' < S^*$ then $S^* \leftarrow S'$, $m^* \leftarrow m'$, $p^* \leftarrow p'$, $r^* \leftarrow r'$, $counter\_stag = 0$, $counter\_mut = 0$; otherwise, $counter\_stag = counter\_stag + 1$, $counter\_mut = counter\_mut + 1$.
Step 9: Let $S_c \leftarrow S'$, $m_c \leftarrow m'$, $p_c \leftarrow p'$, $r_c \leftarrow r'$.
Step 10: $counter\_iter = counter\_iter + 1$, go to *Step 4*.
Step 11. If $S^* < S^{**}$ then $S^{**} \leftarrow S^*$, $m^{**} \leftarrow m^*$, $p^{**} \leftarrow p^*$, $r^{**} \leftarrow r^*$, $NC = NC + 1$, go to *Step 2*; otherwise report the best solutions so far, and stop the algorithm.

Note that algorithm TSM consists of a TS procedure that will be repeatedly applied until a cell formation resulting in the best objective function values, e.g., minimization of the total intercellular movement cost and the machine breakdown cost in this study, has been found. In *Step 1*, the initial number of cells is set at 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 11*. Every time the number of cells is increased, another TS procedure will be started. For a specific cell size, the best routing selection and grouping plan for parts and machines will be calculated iteratively and obtained in *Steps 5–10*. All algorithmic parameters and counters are initialized in *Step 3*. Initial solutions of machine cells, routing selections, and assignments to machine cells are generated 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 6*; otherwise, gene-by-gene mutation is applied to machines to generate a new solution with higher degree of diversification in *Step 5*. If the newly generated neighborhood solution results in a better objective function value, the incumbent solution will be updated and the *counter_stag* and *counter_mut* will be set to 0 in *Step 8*; otherwise, the *counter_stag* and *counter_mut* are increased by 1. The solution process repeats until any of the two stopping criteria in *Step 4* is met. The incumbent solution obtained at this point represents the best solution of the current cell size. If larger cell sizes are considered, it is possible that better solutions may be obtained. The incumbent solution of current cell size is thus compared to the best solution found so far in *Step 11* to determine whether to increase the cell size by 1 and restart another TS procedure to continue the search or to report the best solution found and terminate TSM.

For users having specific preferences in cell size, the proposed algorithm can save considerable amounts of run time since 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.

4. Computational results

To validate the quality of the solutions provided by the proposed algorithm TSM, we have to prepare suitable test instances. However, only a limited amount of research in the context of cell formation problem has dealt with machine breakdown or reliability issues, suitable test problem can very rarely be found from the literature. Eight test instances, as shown in Table 4, are solved in this research. Among them, two (#1 and #5) are drawn from the literature and have been solved optimally in previous studies (Jabal Ameli & Arkat, 2008). The remaining six problems are prepared by adding self-created data such as machine breakdown cost (BC), mean time between failure (MTBF), and production time (PT) to test instances chosen from the literature which have machine-part matrices and process routing data ready. Detailed data of each new test problems are available under request to the authors.

Table 5
Parameters setting for TSM.

Parameter	Value
tls	7
mut_check	$m(NC-1)/2$
γ	0.8
N_{max}	9000
stag_check	$N_{max}/3$

Table 4
Data of test problems.

No.	Original source	Size ($m \times p \times r$)	Randomly generated data
1	Bhide et al. (2005)	$9 \times 8 \times 20$	–
2	Kim, Baek, and Baek (2004)	$10 \times 10 \times 25$	BC, MTBF
3	Sofianopoulou (1999)	$12 \times 20 \times 26$	BC, MTBF, PT
4	Sofianopoulou (1999)	$14 \times 20 \times 45$	BC, MTBF, PT
5	Kazerooni, Luong, and Abhary (1997)	$17 \times 30 \times 63$	–
6	Sofianopoulou (1999)	$18 \times 30 \times 59$	BC, MTBF, PT
7	Lee, Luong, and Abhary (1997)	$30 \times 40 \times 89$	BC, MTBF, PT
8	Hu and Yasuda (2006)	$30 \times 70 \times 149$	BC, MTBF, PT

BC: 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).

Table 6
Results comparison of TSM and optimal solutions by LINGO 8.0.

Test instances					LINGO 8.0 software (B&B)					Proposed method (TSM)					
No.	Source	Size ($m \times p \times r$)	L_m	U_m	*NC*	*Inter_C*	TBC	TC	CPU (s)	*NC*	*Inter_C*	TBC	TC	Std.	CPU(s)
1	Bhide et al. (2005)	$9 \times 8 \times 20$	2	6	2	550	5146	5696*	30	2	550	5146	5696*	0	0.303
2	This study	$10 \times 10 \times 25$	2	5	2	380	1539	1919*	1	2	380	1539	1919*	0	0.366
3	This study	$12 \times 20 \times 26$	2	5	3	150	247	397*	21	3	150	247	397*	0	0.790
4	This study	$14 \times 20 \times 45$	2	5	3	125	213	338*	9226	3	125	213	338*	0	1.187
5	Kazerooni et al. (1997)	$17 \times 30 \times 63$	2	5	4	4300	45864	50164*	323	4	4300	45864	50164*	0	1.775
6	This study	$18 \times 30 \times 59$	2	7	3	165	305	470*	1053	3	165	305	470*	0	1.250
7	This study	$30 \times 40 \times 89$	2	7	5	2925	38192	41117*	7198	5	2925	38192	41117*	0	3.296
8	This study	$30 \times 70 \times 149$	2	8	4	1160	1205	2365	90000	4	895	1204	2099	0	8.467

* Global optimum.

Table 7
Comparison of computational results for ignoring and considering machine reliability.

Test instances					Ignoring reliability					Considering reliability					Cost decreased (%)
No.	Source	Size ($m \times p \times r$)	L_m	U_m	*NC*	*Inter_C*	TBC	TC	CPU(s)	*NC*	*Inter_C*	TBC	TC	CPU (s)	
1	Bhide et al. (2005)	$9 \times 8 \times 20$	2	6	2	525	5366	5891	0.315	2	550	5146	5696	0.303	3.31
2	This study	$10 \times 10 \times 25$	2	5	2	320	1667	1987	0.375	2	380	1539	1919	0.366	3.44
3	This study	$12 \times 20 \times 26$	2	5	3	145	265	410	0.765	3	150	247	397	0.790	3.08
4	This study	$14 \times 20 \times 45$	2	5	3	125	216	341	0.928	3	125	213	338	1.187	0.95
5	Kazerooni et al. (1997)	$17 \times 30 \times 63$	2	5	4	3800	46446	50246	1.593	4	4300	45864	50164	1.775	0.16
6	This study	$18 \times 30 \times 59$	2	7	3	160	323	483	1.203	3	165	305	470	1.250	2.72
7	This study	$30 \times 40 \times 89$	2	7	6	475	44152	44627	5.958	5	2925	38192	41117	3.296	7.87
8	This study	$30 \times 70 \times 149$	2	8	4	875	1316	2191	7.763	4	895	1204	2099	8.467	4.20

Table 4 describes basic problem data and how the machine breakdown cost (BC), mean time between failures (MTBF), and production time (PT) data are created:

1. BC 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.

The proposed algorithm was coded in C++ using Microsoft Visual Studio 6.0 and implemented on a Intel(R) 1.66 GHz PC with 1 GB RAM. Due to the proposed method might have stochastic features, five independent runs were performed for each test instance. The intercellular movement unit cost for all instances is assumed to be five. The computational results are compared with the optimal solutions obtained by the LINGO 8.0 software. Parameter settings for the proposed TSM are given in Table 5.

4.1. Results comparison

Eight test instances considering machine reliability are solved by the proposed TSM and compared with the branch and bound (B&B) algorithm with the LINGO 8.0 software. The maximum run time is set to be 25 h when running the LINGO. The computational results are summarized and compared in Table 6. The results show that the proposed TSM is able to achieve global optimum in seven out of eight test instances in less than 9 s. In addition, the standard deviation for all test instances is equal to 0, which indicates that TSM is able to consistently produce good solutions. As for test instance #8, due to its giant problem size, even the LINGO is not able to find the optimal solution after 25 h of running with the objective value of 2365 obtained. In contrast, the proposed TSM results in a final solution of 2099 in less than nine seconds, which is more than 10% better than the solution of LINGO in terms of solution quality. The final solutions for all eight test problems obtained by TSM are available under request to the authors.

Among the eight test instances, test instance #5 is a medium-sized example. Jabal Ameli and Arkat (2008) solved this problem using LINGO 8.0. Based on the data of the optimal machine-part matrix that appeared in Table 6 of their article, the authors have obtained 50,164 as the optimal objective function value. Jabal Ameli and Arkat (2008) reported that it took about 3 h to obtain the optimal solution. In contrast, the proposed TSM was able to find the optimal solution in 1.775 s, illustrating the superiority of TSM in solution efficiency over other approaches derived from the literature.

To the authors' knowledge, test instance #8 generated in this study is the largest example that has ever been used in literature in analyzing the cell formation problem with alternative process routings and machine reliability considerations. Although it has big size, it still can be solved by the TSM in less than 9 s. The surprisingly good solution efficiency should be attributed to the synergy effects through a proper collocation of both TS and GA techniques. It is also believed that several counters used in the TSM procedure, such as the *counter_stag* and *counter_mut*, play important roles in monitoring the situation of solution stagnancy and controlling the timing for activating the mutation strategy, which contribute to increase the quality and efficiency of the solutions.

4.2. Ignoring reliability vs. considering reliability

Table 7 gives the computational results when the reliability issue is considered in the model. From this table, it can be seen that a greater number of intercellular movements have resulted in when the machine reliability is considered, as opposed to the situation where reliability concern is ignored. But as would be expected, the machine break down cost (TBC) decreased, and so did the total system cost (TC). Thus, the reliability consideration does have meaningful effects on reducing the total system cost.

5. Conclusions

Very limited amount of articles have simultaneously considered the issues of production volume, production sequence, machine reliability and alternative process routings in cell formation problem so far. Accounting for these factors and integrating them into one model makes the CF problem complex but more realistic. Although Jabal Ameli and Arkat (2008) have formulated a binary integer programming model for solving this complicated problem, the resulting long computer run time has motivated the authors of this study to propose an efficient meta-heuristic algorithm. Since both the tabu search and the genetic algorithm have had excellent performances in solving many combinatorial optimization problems, the synergy effects of both heuristic approaches are anticipated in a hybrid algorithm designed to increase the quality and efficiency of solutions. The proposed TS procedure consists of two stages, the initial solution construction and the improvements stage. The similarity coefficient-based method is adapted in the first stage to produce good initial solutions, while the tabu search continuously improves and generates more effective solutions through the neighborhood moves in the second stage. Computational experiences of test problems from the literature as well as those newly generated by this study show that the reliability consideration has meaningful effects on reducing the total system cost. Additionally, for the test problem which took the mathematical approach about 3 h to solve, this study is able to solve it optimally in less than two seconds. Even the largest test problem that has ever been used in the literature has been solved in less than 9 s. The superiority of the proposed algorithm, TSM, in both solution effectiveness and efficiency over other approach from the literature can be easily observed.

For future research, several other factors may be added into the current model or even treat them as decision objectives. These factors may include the cell layout, intracellular machine layout and the cell load variation.

References

Albadawi, Z., Bashir, H. A., & Chen, M. (2005). A mathematical approach for the formation of manufacturing cells. *Computers and Industrial Engineering, 48*, 3–21.

Alhourani, F., & Seifoddini, H. (2007). Machine cell formation for production management in cellular manufacturing systems. *International Journal of Production Research, 45*(4), 913–934.

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

Bhide, P., Bhandwale, A., & Kesavadas, T. (2005). Cell formation using multiple process plans. *Journal of Intelligent Manufacturing, 16*, 53–65.

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

Burbidge, J. L. (1975). *The introduction of group technology*. New York: Wiley.

Das, K., Lashkari, R. S., & Sengupta, S. (2007). Reliability consideration in the design and analysis of cellular manufacturing systems. *International Journal of Production Economics, 105*, 243–262.

Diallo, M., Perreval, H., & Quillot, A. (2001). Manufacturing cell design with flexible routing capability in presence of unreliable machines. *International Journal of Production Economics, 74*(1–3), 175–182.

Diaz, B. A., Lozano, S., Racero, J., & Guerrero, F. (2001). Machine cell formation in generalized group technology. *Computers and Industrial Engineering, 41*, 227–240.

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.

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

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

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

Jeon, G., Broering, M., Leep, H. R., Parsaei, H. R., & Wong, J. P. (1998). Part family formation based on alternative routes during machine failure. *Computers and Industrial Engineering, 35*(1–2), 73–76.

Kazerooni, M., Luong, H. S., & Abhary, K. (1997). A genetic algorithm based cell design considering alternative routing. *Computer Integrated Manufacturing Systems, 10*(2), 93–107.

Kim, C.-O., Baek, J.-G., & 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.

Kumar, K. R., Kusiak, A., & Vannelli, A. (1986). Grouping of parts and components in flexible manufacturing systems. *International Journal of Production Research, 24*, 387–397.

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

Kusiak, A., & Cho, M. (1992). Similarity coefficient algorithms for solving the group technology problem. *International Journal of Production Research, 30*, 2633–2646.

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

Lei, D., & Wu, Z. (2005). Tabu search approach based on a similarity coefficient for cell formation in generalized group technology. *International Journal of Production Research, 43*(19), 4035–4047.

Logendran, R., & Talkington, D. (1997). Analysis of cellular and functional manufacturing system in the presence of machine breakdown. *International Journal of Production Economics, 53*(3), 239–256.

Lozano, S., Adenso-Diaz, B., & Onieva, L. (1999). A one-step tabu search algorithm for manufacturing cell design. *Journal of the Operational Research Society, 50*, 509–516.

Lozano, S., Guerrero, F., Eguia, I., & Onieva, L. (1999). Cell design and loading in the presence of alternative routing. *International Journal of Production Research, 37*, 3289–3304.

McAuley, J. (1972). Machine grouping for efficient production. *Production Engineer, 52*, 53–57.

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

Nair, G. J., & Narendran, T. T. (1998). CASE: A clustering algorithm for cell formation with sequence data. *International Journal of Production Research, 36*, 157–179.

Savsar, M. (2000). Reliability analysis of a flexible manufacturing cell. *Reliability Engineering & System Safety, 67*, 147–152.

Seifoddini, H., & Djassemi, M. (1995). Merits of the production volume based similarity coefficient in machine cell formation. *Journal of Manufacturing Systems, 14*, 35–44.

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

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

Sun, D., Lin, L., & Batta, R. (1995). Cell formation using tabu search. *Computers and Industrial Engineering, 28*, 485–494.

Wang, J. (2003). Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm. *Automatica, 39*, 1607–1615.

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

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

Wu, T.-H., Chang, C.-C., & Chung, S.-H. (2008). A simulated annealing algorithm to manufacturing cell formation problems. *Expert Systems with Applications, 34*, 1609–1617.

Wu, T.-H., Chung, S.-H., & Chang, C.-C. (2008). Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings. *Expert Systems with Applications, 36*, 3652–3661.

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

Yin, Y., & Yasuda, K. (2006). Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *International Journal of Production Economics, 101*, 329–352.

Zakarian, A., & Kusiak, A. (1997). Modeling manufacturing dependability. *IEEE Transactions Robotics and Automation, 13*(2), 161–168.