Expert Systems with Applications 36 (2009) 3652–3661

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings

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

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

ARTICLE INFO

Keywords:
Cell formation problem with alternative routings
Generalized GT
Hybrid simulated annealing algorithm

ABSTRACT

In this study, a hybrid simulated annealing algorithm with mutation operator is proposed to solve the manufacturing cell formation problem considering multiple process routings for parts, so that either the intercellular movements are minimized or the grouping efficacy is maximized, depending on the definition of the decision objective. The proposed algorithm is designed mainly to explore solution regions efficiently and to expedite the solution search process. The performance of the proposed algorithm is tested by a range of test problems, some of which are from the literature and some of which are generated within this study. The comparative study shows that the proposed algorithm improves the best results found in the literature for 28.6% of the test problems and the percentages of improvement are even higher than 18% in several test instances.

1. Introduction

Group technology (GT) groups parts that have similar design characteristics or manufacturing characteristics into part families in order to make manufacturing systems more efficient and productive. Cellular manufacturing is the implementation of group technology in the manufacturing process. Cellular manufacturing decomposes the entire production system into several mutually separable production cells, then assigns machines to these cells to process one or more part families. Each cell is operated independently; the intercellular movements are minimized, i.e., parts do not have to move from one cell to another for processing. Extensive research has been devoted to cell formation (CF) problems for identifying machine cells and part families. Selim, Askin, and Vakharia (1998) have provided comprehensive reviews of the methodologies for CF problems.

Many CF researches assume that each part has a unique process routing which indicates the sequence of machines used to process each part. This assumption obviously ignores real situations, in which each operation of a particular part may be performed on alternative machines, i.e., parts may have multiple process routings. The manufacturing industry has noted the flexibility and other benefits of parts with multiple process routings (Kusaik, 1987).

Limited studies of the cell formation problem considering multiple process routings, also called the generalized GT problem (Won & Kim, 1997), can be found. Kusaik (1987) presented a p-median model to select process routings and to form part families simultaneously. Nagi, Harlarakis, and Proth (1990) and Sankaran and Kasilingam (1990) proposed mathematical models for solving the problem. In addition to mathematical approaches, many cell formation methods use similarity measures between parts or machines to form part-machine groups. Kusiak and Cho (1992) presented a similarity coefficient method that defines a similarity coefficient between process routings of parts. In regard to the decision objectives of the problem under study, Won and Kim (1994) presented an assignment model to maximize the sum of similarity coefficients between process routings in the same family, while Adil, Rajamani, and Strong (1996) developed a non-linear integer programming model that considered both the minimization of a weighted sum of the voids and the exceptional elements in the objective function. Won and Kim (1997) later defined the generalized machine similarity coefficient and used multiple clustering criteria to effectively form machine cells with the lowest possible number of intercellular flows. Their method, however, generates singleton machine cells or requires human judgment in the solution procedure. Motivated by previous work, Won (2000) used the generalized machine similarity coefficient between machine pairs to propose two new p-median models. Spiliopoulos and Sofianopoulou (2007) presented a bounding scheme that examines all combinations of alternate routings and solves only a few cell formation problems, thereby reducing the complexity of the solution space.

Due to their excellent performances in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic

* Corresponding author. Tel./fax: +886 2 86746574.
E-mail address: taiwu@mail.ntpu.edu.tw (T.-H. Wu).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.02.060

algorithms (GA), simulated annealing (SA), neural networks (NN) and tabu search (TS) make up another class of search methods that has been adopted to solve the CF problem and its variants efficiently. Sofianopoulou (1999) presented a simulated annealing algorithm to select process routings for parts and to construct machine cells sequentially. Adenso-Díaz, Lozano, Racero, and Guerrero (2001) proposed an efficient tabu search algorithm to select process routings for parts, to group parts into families, and to group machines into cells simultaneously. Wu, Chen, and Yeh (2004) decomposed and solved the problem sequentially in three stages: process routing selection, part assignment, and machine assignment, respectively. Among these three subproblems, determination of part assignment consumes most of the computational effort and was hence solved by a tabu search algorithm. Lei and Wu (2005) published an algorithm in which an initial solution is first generated by a new similarity coefficient-based method and is later improved iteratively by a fast, effective tabu search algorithm.

The literature cited above shows that several meta-heuristic approaches have been used to solve the cell formation problem with alternative process routings. However, each of the aforementioned meta-heuristic algorithms has different strengths and weaknesses. For example, GA explores the solution space by means of a population of search points and operators such as selection, crossover and mutation. It produces diversified solutions yet suffers from poor convergence properties. In contrast with GA, SA converges easily at local optima but may not be able to explore the solution space and find the global optima. A strategy which combines GA and SA can reasonably be expected to give rise to complementary strengths. This idea has been implemented in some researches and has obtained positive results (Wong, 2001; Soke & Bingul, 2006).

The purpose of this study is to develop a procedure that is efficient and effective for the cell formation problem with alternative process routings. This research combines SA and GA to generate synergy. However, to avoid excessive consumption of computational effort, only the mutation operator from the GA is adopted – mainly to escape from local solutions and to prevent premature convergence. Two sets of test problems with various sizes, one from the literature and one generated in this study, are used to test the performance of the proposed hybrid algorithm. The corresponding results are compared to the best results of several well-known published algorithms.

The remainder of this article is organized as follows. In Section 2, we describe the problem definition. The proposed hybrid heuristic is presented in Section 3. Section 4 shows the computational results on test problems, and Section 5 concludes the paper.

2. Cell formation problem with alternative routings

Cell formation in a given 0–1 machine-part incidence matrix involves rearrangement of its rows and columns to create part families and machine cells. Researches usually attempt to determine a rearrangement by which the inter-cellular movement can be minimized and the utilization of the machines within a cell can be maximized. Fig. 1a gives a sample machine-part matrix. A rearrangement for that machine-part matrix, in which two blocks can be observed along the diagonal of the matrix, is given in Fig. 1b.

Parts

Machines	1a	1b	2a	2b	3a	4a	4b	5a	5b	5c	5d	6a	6b	7a	8a	8b	9a	9b	10a	10b	11a	11b
1					1			1	1					1		1					1	
2	1	1	1	1		1	1					1			1	1	1	1			1	1
3		1						1	1	1	1			1					1		1	1
4			1	1	1	1	1	1		1	1	1	1		1	1	1		1	1	1	
5	1		1			1			1		1				1			1		1		1
6	1	1	1	1			1						1		1	1	1	1			1	

Fig. 2. Incidence matrix of sample CF problem with alternative process routings.

There have been several measures of goodness of machine-part groups in cellular manufacturing in the literature. Two measures frequently used are the grouping efficiency (Chandrashekharan & Rajagopalan, 1986) and the grouping efficacy (Kumar & Chandrasekharan, 1990), because they are easy to implement. Although grouping efficiency has been used widely, critics argue that in some cases the size of the matrix impairs its discrimination ability. The grouping efficacy Γ can be defined as:

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

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

Cases in which each part may have more than one process routing, such as the case shown in Fig. 2, are even more complicated than the simple cell formation problem. Formation of part families, formation of machine cells, and selection of routings for each part need to be determined in order to achieve the decision objectives, such as the minimization of inter-cellular movement or the maximization of grouping efficacy.

3. Proposed hybrid algorithm

SA was originally proposed by Metropolis, Rosenbluth, and Teller (1953) to simulate the annealing process. After generating an initial solution, SA attempts to move from the current solution to one of the neighborhood solutions. The changes in the objective function values (ΔE) are calculated. If the new solution results in a better objective value, it is accepted. However, if the new solution yields a worse value, it can still be accepted according to a probability function, i.e., the Boltzmann function, $P(\Delta E) = \exp(-\Delta E/k_B T)$, where k_B is Boltzmann's constant and T is the current temperature. This check is done by selecting a random number from (0, 1). If the

(a) machine-part matrix

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

(b) rearrangement of machine-part matrix

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

Fig. 1. Sample machine-part matrix and its rearrangement.

random number chosen is less than or equal to the probability value, the new solution is accepted; otherwise, it is rejected. By accepting worse solutions, SA can avoid being trapped on local optima. The parameter T is gradually decreased by a cooling function as SA proceeds until the stopping criterion is met.

GA is inspired by the evolution of living things that occurs in natural biology. Some individuals are selected as parents to generate offspring via the crossover operator. All the individuals are then evaluated for fitness and the fittest are selected to survive. The process of reproduction, evaluation, and selection is repeated until the stopping criterion is met. There is usually a certain probability that a mutation operator might be applied to the individuals; mutations can cause GA to escape from local solutions and can prevent premature convergence. It is used mainly to increase the diversity of the population and to ensure that an extensive search will be performed.

We anticipate the synergy effect between the SA and the GA by presenting a hybrid algorithm employing the SA, together with the mutation operator from the GA to increase the quality and efficiency of solutions. The solution procedure of the proposed algorithm consists of two stages: initial solution construction and later solution improvements, which will be detailed in Sections 3.1 and 3.2, respectively.

3.1. Initial solution construction

A large number of similarity coefficients methods (SCM) have been proposed for grouping entities such as parts or machines in the simple CF problem, far fewer SCMs have been designed especially for the CF problem with alternate routings. The works by Kusiak and Cho (1992), and by Won and Kim (1997) can be considered as the two most widely used approaches. Kusiak and Cho's method is a part-based approach which uses a similarity coefficient defined between part routings, while Won and Kim's study defined a machine-based similarity coefficient in their algorithm. Compared to machine-based similarity coefficient methods, the part-based similarity coefficient methods suffer from a computational burden since the number of parts in a cell formation problem is usually much greater than the number of machines. We hence adopt Won and Kim's machine-based similarity coefficient method to generate the similarity matrix for forming machine cells when constructing the initial solution in this research.

After the machine cells have been obtained, the next task is to assign a process routing for each part to the machine cells. Unlike Won and Kim's work (1997), which followed the maximum density rule, i.e., a routing of a part is assigned to a machine cell that has the most of its processings, we assign the part routings to machine cells that will result in the least number of exceptional elements. The proposed part routing assignment procedure is described as follows:

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

Step 2. For all alternative routings of each part, evaluate the resulting number of exceptional elements of assignments to each machine cell. The routing with the least number of exceptional elements is selected as the process routing for later manufacturing. If a tie happens, the one with the least number of voids is chosen. If the least number of voids is tied again, make a random selection.

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

An initial solution for the CF problem with alternative routings can be obtained at this point by using the machine-based similarity coefficient methods and the above procedure for determining each part routing. This is illustrated by the following example.

Table 1
Similarity matrix for machines in Fig. 2

Machines	1	2	3	4	5	6
1	–	0.20	0.43	0.40	0.08	0.20
2		–	0.20	0.60	0.67	0.75
3			–	0.27	0.18	0.20
4				–	0.42	0.60
5					–	0.36
6						–

Consider the sample machine-part matrix with alternative process routings in Fig. 2: the corresponding similarity matrix for machines can be obtained by using the formula (Won & Kim, 1997) below and is listed in Table 1:

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

where S_{ij} = similarity coefficient between machines i and j, $N_i = \sum_{p=1}^{P} a_i^p, N_j = \sum_{p=1}^{P} a_j^p, N_{ij} = \sum_{p=1}^{P} a_{ij}^p$, P = number of parts.

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

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

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

Suppose there are two cells to be formed. The largest coefficient in the similarity matrix of Table 1 is 0.75, indicating that machines 2 and 6 must be assigned to the same cell. We proceed with the second largest coefficient in the matrix, 0.67, appearing in pair (2,5). Machine 5 is thus assigned together with machines 2 and 6. By repeating the same logic, it can finally be obtained that machines 2, 4, 5 and 6 should be assigned in the same cell, while machines 1 and 3 should be assigned in another cell. The incidence matrix of the sample problem displayed in Fig. 2 is thus rearranged as shown in Fig. 3. The number of exceptional elements for each part routing-machine cell combination is given in Fig. 4.

Using the methodology proposed above, the initial solution matrix for the CF problem with alternative routings can thus be obtained and shown in Fig. 5, in which three exceptional elements are found and the corresponding value of grouping efficacy is 0.683. This solution is superior to the one adopting the maximum density rule when assigning part routings to machine cells (Won & Kim, 1997), in which four exceptional elements are resulted in, as shown in Fig. 6.

Parts

Machines	1a	1b	2a	2b	3a	4a	4b	5a	5b	5c	5d	6a	6b	7a	8a	8b	9a	9b	10a	10b	11a	11b
2	1	1	1	1		1	1					1			1	1	1	1			1	1
4			1	1	1	1	1	1		1	1	1	1		1	1	1		1	1	1	
5	1		1			1			1		1				1			1		1		1
6	1	1	1	1			1						1		1	1	1	1			1	
1					1			1	1					1		1					1	
3		1						1	1	1	1			1				1			1	1

Fig. 3. Incidence matrix after rearrangement.

		1	1	2	2	3	4	4	5	5	5	5	6	6	7	8	8	9	9	10	10	11	11
		a	b	a	b	a	a	b	a	b	c	d	a	b	a	a	b	a	b	a	b	a	b
Cell 1 (machines 2+4+5+6)	e	0	1	0	0	1	0	0	2	2	1	1	0	0	2	0	1	0	0	1	0	2	1
Cell 2 (machines 1+3)	e	3	2	4	3	1	3	3	1	1	1	2	2	2	0	4	3	3	3	1	2	3	2

Fig. 4. Number of exceptional elements for each part routing-machine cell combination.

Parts

		1	2	4	6	8	9	10	11	3	5	7
		a	a	a	a	a	a	b	b	a	a	a
Machines	2	1	1	1	1	1	1		1			
	4		1	1	1	1	1	1		1	1	
	5	1	1	1		1		1	1			
	6	1	1			1	1					
	1									1	1	1
	3								1		1	1

Fig. 5. Initial solution matrix obtained by using the proposed methodology.

Parts

		1	2	4	6	8	9	10	11	3	5	7
		a	a	a	a	a	a	b	a	a	a	a
Machines	2	1	1	1	1	1	1		1			
	4		1	1	1	1	1	1	1	1	1	
	5	1	1	1		1		1				
	6	1	1			1	1		1			
	1								1	1	1	1
	3								1		1	1

Fig. 6. Initial solution matrix obtained by adopting maximum density rule.

3.2. Solution improvements

At this stage in the solution procedure, the initial solution generated from Section 3.1 is improved through a sequence of neighborhood moves.

Note that when generating the initial solution, part routing selection and assignment to machine cells can not be implemented until the machine cells have been formed. The procedure for forming machine cells actually happens before the procedure for selection and assignment of part routings, and thus is critical to the quality of the entire solution. The insertion-move, which is a type of move used to search for better neighborhood solutions of the current machine cells, is introduced and defined in this section.

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

$$M(i',j) = \max\{obj^{(i',j)} - obj^{current}, \forall i' \in I, i' \neq i, \forall j \in J\},$$

where I and J are the sets for cells and machines, respectively.

3.3. Proposed hybrid algorithm HSAM

This section describes the proposed hybrid simulated annealing algorithm with mutation, HSAM. It is evident that the number of cells to be formed will affect the grouping solutions obtained in the CF problem. In our algorithm, the number of cells resulting in the best objective values is generated automatically. To preserve flexibility, users are permitted to specify the preferred number of cells.

In GA, application of the mutation operator enables the algorithm to explore unvisited solution regions and to generate new solutions better than currently best ones. Implementation of the mutation operator in this study is similar to the traditional gene-by-gene mutation with a small probability p. A mutation check is performed machine by machine on the incumbent solution of machine cells formed. For each machine, a random number from $(0,1)$ is first drawn. If the value is greater than or equal to p, then the machine stays in the current cell; otherwise it is moved to another cell that is randomly determined.

Before proceeding to the proposed algorithm HSAM, we introduce some notations.

NC number of cells (cell size)
M number of machines
U maximum number of machines per cell
T_0 initial temperature
T_f final temperature
L Markov chain length
α cooling rate
r_0 initial solution of part- route assignment
r current solution of part- route assignment
r' neighborhood solution of part- route assignment
r^* incumbent solution of part- route assignment of current cell size
r^{**} best solution of part- route assignment so far
m_0 initial solution of machines assignment
m 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
E(m,r) total number of intercellular moves of all parts
p mutation probability
counter_MC number of times a neighborhood solution is generated in a specific temperature
counter_BF number of times neighborhood solution fails in the Boltzmann test
counter_stag number of times incumbent solution did not improve

3.4. Algorithm HSAM

Step 1. Set $E(m^{**},r^{**}) = \infty$, $\mathrm{NC} = \lceil \frac{M}{U} \rceil$.
Step 2. Initialize counters, SA and other parameters: T_0, T_f, α, L, p, *counter_MC* = 0, *counter_stag* = 0, *counter_BF* = 0, and set $T_k = T_0$.
Step 3. Generate an initial solution of machine cells, m_0. Let $m = m_0$, $m^* = m_0$. On the base of initial solution m_0, gener-

ate an initial solution of routing selection and assignment to machine cells, r_0. Let $r = r_0$, $r^* = r_0$.

Step 4. If $T_k \geqslant T_f$ and $counter\_stag \leqslant stag\_check$ and $E\ (m^*, r^*) \neq 0$, repeat *Steps 5* and *6*; otherwise, go to *Step 7*.

Step 5. If $counter\_MC < L$, repeat *Steps 5.1–5.7*; otherwise, go to *Step 6*.

Step 5.1. If $counter\_BF \geqslant 1$, apply mutation operator to m^* and generate a new solution of machine cells m'.

Step 5.2. If $counter\_BF < 1$, generate a new solution of machine cells m' through neighborhood searching for m by performing the insertion-move.

Step 5.3. Read new solution of machine cells m' from above steps and generate corresponding solution of routing selection and assignment to machine cells r' using procedure in Section 3.1.

Step 5.4. Calculate $\Delta E = E(m', r') - E(m, r)$. If $\Delta E \leqslant 0$, $m = m', r = r'$, $counter\_BF = 0$, go to *Step 5.6*; otherwise, go to *Step 5.5*.

Step 5.5. Generate $u \in U(0,1)$, if $\exp\left(\frac{-\Delta E}{T_k}\right) > u$, $m = m', r = r'$, $counter\_BF = 0$; otherwise, $counter\_BF = counter\_BF + 1$.

Step 5.6. If $E(m', r') < E(m^*, r^*)$, then $m^* = m'$, $r^* = r'$, $counter\_stag = 0$; otherwise, $counter\_stag = counter\_stag + 1$.

Step 5.7. $counter\_MC = counter\_MC + 1$.

Step 6. $T_k = T_k \times \alpha$, $counter\_MC = 0$, $counter\_BF = 0$.

Step 7. If $E(m^*, r^*) < E(m^{**}, r^{**})$, then $E(m^{**}, r^{**}) = E(m^*, r^*)$, $m^{**} = m^*$, $r^{**} = r^*$, NC = NC + 1, *go to Step 2*; otherwise report the current $E(m^{**}, r^{**})$, m^{**}, r^{**}, NC-1, and stop the algorithm.

Note that algorithm HSAM consists of an SA procedure that is repeatedly applied until a cell formation resulting in the best objective function values, e.g., number of exceptional elements or grouping efficacy, has been found. In *Step 1*, initial number of cells is set at the nearest integer that is greater than M/U, which is a conservative setting; it gradually increases by increments of 1 as long as solution improvement is observed in *Step 7*. Every time the number of cells is increased, another SA 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 5.1–5.7* and *Step 6*. All algorithmic parameters and counters are initialized in *Step 2*. Initial solutions of machine cells, routing selections, and assignments to machine cells are generated in *Step 3*. *counter_BF* is used to record the number of times a solution fails in Boltzmann's test to avoid getting trapped in local solutions and wasting too much computational effort. As long as the value of *counter_BF* is 0, a new neighborhood solution is generated through the insertion-move in *Step 5.2*; otherwise, gene-by-gene mutation is applied in order to generate a new solution with higher degree of diversification in *Step 5.1*. If the newly generated neighborhood solution is better than the current solution, a replacement is made in *Step 5.4*. If the newly generated neighborhood solution is worse than the current solution, a Boltzmann function test is performed in *Step 5.5*. Comparison with the incumbent solution of current cell size then follows. The incumbent solution will be updated in *Step 5.6* if the newly generated neighborhood solution results in a better objective function value; otherwise, the *counter_stag*, monitoring the solution stagnancy, is increased by 1. The solution process repeats until any of the three stopping criteria in *Step 4* is met. The incumbent solution obtained at this point represents the best solution of current cell size. If larger cell sizes are considered, it is possible that better solutions may result. The incumbent solution of current cell size is thus compared to the best solution found so far in *Step 7* to determine whether to increase the cell size by 1 and restart another SA procedure to continue the search or to report the best solution found and terminate HSAM.

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 resulting in the best objective function values. The savings in run time become even more significant as the cell size increases.

After intensive testing, *stag_check* is set at 1000. Initial temperature T_0, final temperature T_f, cooling rate α, and the Markov chain length L of the SA procedure are set at 10, 1, 0.9, 2000, respectively. The mutation probability p of each gene is set at 0.8 in this study.

4. Computational results and discussion

This section uses test problems from the literature as well as newly created problems to illustrate the proposed solution method HSAM for cell formation considering alternative process routings. The computational results are compared with those of algorithms reported in the literature. The proposed algorithm HSAM was coded in C and implemented on a Pentium III 933 MHz personal computer with 256 MB RAM. Because of the stochastic features of SA, five independent runs were performed for each test instance.

4.1. Computational results

The test instances in the first problem set are from the open literature. For each instance, Table 2 shows the problem source, size

Table 2
Problem description and comparisons of computational results

Test instances					Other approaches								Proposed approach			
No.	Source	Size	L	U	TS1	TS2	SA	P0	MP1	MP2	TSPA	BS	HSAM	Cell size	CPU (s)	Efficacy (%)
1	Won and Kim (1997)	4 × 4 × 8	2	3	–	–	–	0	0	0	–	–	0	2	0.006	100.00
2	Kusaik (1987)	4 × 5 × 11	2	3	0	0	0	0	0	0	0	0	0	2	0.013	90.00
3	Moon and Chi (1992)	6 × 6 × 13	2	3	–	–	–	0	0	0	0	–	0	2	0.031	83.33
4	Sankaran and Kasilingam (1990)	6 × 10 × 20	2	4	–	–	–	4	2	2	2	–	2	2	0.050	69.44
5	Won and Kim (1997)	7 × 10 × 23	2	3	3	3	3	3	3	3	3	–	3	3	0.034	74.07
6	Logendram et al. (1994)	7 × 14 × 32	2	3	–	–	–	7	7	5	5	–	5	3	0.048	67.57
7	Adil et al. (1996)	10 × 10 × 24	2	4	–	–	–	5	2	4	–	–	2	3	0.102	80.00
8	Kasilingam and Lashkari (1991)	10 × 15 × 28	2	4	–	–	–	11	12	11	–	–	10	3	0.134	57.81
9	Won and Kim (1997)	11 × 10 × 22	2	3	4	4	4	3	3	3	3	4	3	4	0.105	77.42
10	Sofianopoulou (1999)	12 × 20 × 26	2	5	29	29	29	–	–	–	–	29	29	3	0.216	47.06
11	Sofianopoulou (1999)	14 × 20 × 45	2	5	25	25	29	–	–	–	–	25	24	3	0.313	50.83
12	Sofianopoulou (1999)	18 × 30 × 59	2	7	33	33	35	–	–	–	–	32	26	3	0.506	39.65
13	Nagi et al. (1990)	20 × 20 × 51	2	5	1	1	7	–	7	3	–	1	1	5	0.528	79.52
14	Won and Kim (1997)	26 × 28 × 71	2	7	23	23	34	–	25	22	–	–	13	5	0.569	62.21

in terms of the number of machines, number of parts, and the number of alternative process routings, the minimum (L) and maximum (U) number of machines allowed in each cell, and the number of exceptional elements found. The computational results were compared with the best results found in the literature, i.e., the TS-1 (Lei & Wu, 2005), TS-2 (Adenso-Díaz et al., 2001), SA (Sofianopoulou, 1999), P0 (Kusaik, 1987), MP1 (Won, 2000), MP2 (Won, 2000), TSPA(Wu et al., 2004), and BS (a bounding scheme by Spiliopoulos & Sofianopoulou, 2007). Note that in test problem #8, we followed Won (2000)'s setting and did not consider the duplication of machines #1 and #10.

According to Table 2, the best results obtained by HSAM are better than or equal to the reported best results in all test problems. To be more specific, for 10 problems (the first seven problems, #9, #10, and #13), HSAM obtains values of the number of exceptional elements that are equal to the best results found in the TS1, TS2, SA, P0, MP1, MP2, TSPA, and BS methods; HSAM improves the values of the number of exceptional elements for the remaining 4 problems (#8, #11, #12, and #14). The corresponding solution matrices of these four problems obtained by HSAM are presented in Appendix A. In test problem #12, the percentage improvement of HSAM is higher than 18%; the improvement even reaches 40.9% in test problem #14. Dominance of HSAM over other approaches reported in the literature becomes even more significant for problems with larger sizes.

In addition to the optimal number of exceptional elements for each test problem, corresponding values of grouping efficacy and run times are provided in Table 2 as well. It can be observed that HSAM solves all the test problems in an extremely efficient manner. The run time consumed has never been longer than 0.569 s. Taking test problems #11, #12, and #13 as examples, the bounding scheme proposed by Spiliopoulos and Sofianopoulou (2007) consumed 37, 687, and 82 s, respectively, to find the final solutions while HSAM only took 0.313, 0.506, and 0.528 s, respectively, to find the optimal solutions – these numbers are striking considering that the CPU of their computer had higher specifications (1.8 GHz with 512 MB RAM) than ours (933 MHz with 256 MB RAM).

Following commonly accepted practice for the CF problem considering alternative routings, this study adopts the decision objective of minimizing the number of exceptional elements in designing HSAM. However, HSAM is capable of generating a grouping plan which maximizes the grouping efficacy, another widely used measure of goodness of machine-part groups in cellular manufacturing, by a very minor revision. The same test problems as in Table 2 are used, and the computational results are given in Table 3. We call the HSAM minimizing the intercellular movements HSAM1, while the HSAM maximizing the grouping efficacy is referred to as HSAM2. It can be observed from Table 3 that the number of exceptional elements of HSAM1 are less than or equal to those of HSAM2, as expected. In contrast, HSAM2 performs better in values of grouping efficacy. In test problems #4, #5, #6, #7 and #9, HSAM2 not only produces better values of grouping efficacy than HSAM1, it even obtains the same number of exceptional elements as HSAM1 does. This indicates that, for some cases, HSAM2 is able to find a grouping plan resulting in the best grouping efficacy among grouping plans which all have the minimum number of exceptional elements.

It is hence suggested that HSAM2 be applied after HSAM1 has been used for test instances. If both approaches result in the same number of exceptional elements, then the solution produced by HSAM2 can be considered as a better decision alternative than the one by HSAM1 since it has taken into account both decision objectives, i.e., minimization of exceptional elements and maximization of grouping efficacy.

4.2. Further analysis

This section examines the performance of HSAM when solving ten large-sized test problems and performs further analysis on the effectiveness of some mechanisms designed in HSAM. Problems #20 and #24 are directly adopted from Wu et al. (2004); eight more large-sized test problems are randomly generated in this study. Firstly, eight large-sized cell formation test problems from the literature are chosen (problems #20, #26, #29, #30, #31, #33, #34, and #35 from Table 7, Conçalves & Resende, 2004). Secondly, for each test problem, the number of alternative routes for each part is randomly selected between 1 and 3. At last, the operations in each route are determined randomly and described as follows. The original machine-part incidence matrix is used as the base. For each operation in the process route, a random number is drawn and compared with a pre-set number, 0.8. If the random number is greater than 0.8, the operation is changed from 0 to 1, or from 1 to 0; otherwise, no change is made. The idea is that we would like the new part routing to have only about 20% difference from the original part routing.

This research presented a hybrid algorithm HSAM employing the SA, together with the mutation operator from the GA, and a counter for monitoring solution stagnancy to increase the quality and efficiency of solution. The excellent computational results

Table 3
Comparisons of computation results of different decision objectives

Test instances					HSAM1 (HSAM minimizing intercellular moves)				HSAM2 (HSAM maximizing grouping efficacy)			
No.	Source	Size	L	U	EE	Cell size	Efficacy (%)	CPU (s)	EE	Cell size	Efficacy (%)	CPU (s)
1	Won and Kim (1997)	$4 \times 4 \times 8$	2	3	0	2	100.00	0.006	0	2	100.00	0.006
2	Kusaik (1987)	$4 \times 5 \times 11$	2	3	0	2	90.00	0.013	0	2	90.00	0.014
3	Moon and Chi (1992)	$6 \times 6 \times 13$	2	3	0	2	83.33	0.031	0	2	83.33	0.034
4	Sankaran and Kasilingam (1990)	$6 \times 10 \times 20$	2	4	2	2	69.44	0.050	2	2	72.22	0.053
5	Won and Kim (1997)	$7 \times 10 \times 23$	2	3	3	3	74.07	0.034	3	3	81.48	0.039
6	Logendram et al. (1994)	$7 \times 14 \times 32$	2	3	5	3	67.57	0.048	5	3	69.44	0.063
7	Adil et al. (1996)	$10 \times 10 \times 24$	2	4	2	3	80.00	0.102	2	3	82.86	0.100
8	Kasilingam and Lashkari (1991)	$10 \times 15 \times 28$	2	4	10	3	57.81	0.134	11	3	61.90	0.150
9	Won and Kim (1997)	$11 \times 10 \times 22$	2	3	3	4	77.42	0.105	3	4	80.65	0.114
10	Sofianopoulou (1999)	$12 \times 20 \times 26$	2	5	29	3	47.06	0.216	36	4	49.47	0.349
11	Sofianopoulou (1999)	$14 \times 20 \times 45$	2	5	24	3	50.83	0.313	28	4	54.29	0.438
12	Sofianopoulou (1999)	$18 \times 30 \times 59$	2	7	26	3	40.18	0.506	45	6	47.45	1.564
13	Nagi et al. (1990)	$20 \times 20 \times 51$	2	5	1	5	79.52	0.528	1	5	79.52	0.573
14	Won and Kim (1997)	$26 \times 28 \times 71$	2	7	13	5	62.21	0.569	16	6	72.48	1.581

Note: EE denotes total number of exceptional elements.

Table 4
Comparison of computational results of SA, SA with stagnancy control and HSAM

Test instances					SA			SA with stagnancy control			HSAM		
No.	Source	Size	*L*	*U*	EE	Cell size	CPU (s)	EE	Cell size	CPU (s)	EE	Cell size	CPU (s)
1	Won and Kim (1997)	$4 \times 4 \times 8$	2	3	0	2	0.041	0	2	0.002	0	2	0.006
2	Kusaik (1987)	$4 \times 5 \times 11$	2	3	0	2	0.042	0	2	0.003	0	2	0.013
3	Moon and Chi (1992)	$6 \times 6 \times 13$	2	3	5	2	0.128	5	2	0.009	0	2	0.031
4	Sankaran and Kasilingam (1990)	$6 \times 10 \times 20$	2	4	2	2	0.423	2	2	0.025	2	2	0.050
5	Won and Kim (1997)	$7 \times 10 \times 23$	2	3	3	3	0.475	3	3	0.028	3	3	0.034
6	Logendram et al. (1994)	$7 \times 14 \times 32$	2	3	5	3	0.617	5	3	0.034	5	3	0.048
7	Adil et al. (1996)	$10 \times 10 \times 24$	2	4	2	3	1.533	2	3	0.078	2	3	0.102
8	Kasilingam and Lashkari (1991)	$10 \times 15 \times 28$	2	4	12	3	1.958	12	3	0.099	10	3	0.134
9	Won and Kim (1997)	$11 \times 10 \times 22$	2	3	3	4	1.422	3	4	0.069	3	4	0.105
10	Sofianopoulou (1999)	$12 \times 20 \times 26$	2	5	29	3	2.759	29	3	0.144	29	3	0.216
11	Sofianopoulou (1999)	$14 \times 20 \times 45$	2	5	27	3	3.675	27	3	0.181	24	3	0.313
12	Sofianopoulou (1999)	$18 \times 30 \times 59$	2	7	30	3	6.394	30	3	0.302	26	3	0.506
13	Nagi et al. (1990)	$20 \times 20 \times 51$	2	5	1	5	7.536	1	5	0.403	1	5	0.528
14	Won and Kim (1997)	$26 \times 28 \times 71$	2	7	14	5	17.328	14	5	0.805	13	5	0.569
15	This study	$20 \times 35 \times 79$	2	5	70	5	12.474	70	5	0.709	65	4	0.699
16	This study	$24 \times 40 \times 82$	2	5	112	5	12.322	112	5	0.569	104	5	0.895
17	This study	$28 \times 46 \times 89$	2	5	175	7	29.949	176	7	2.081	171	6	1.708
18	This study	$30 \times 41 \times 74$	2	7	119	5	16.586	115	5	0.813	107	5	1.309
19	This study	$30 \times 50 \times 102$	2	8	142	4	17.567	147	4	0.813	136	4	1.738
20	Wu et al. (2004)	$35 \times 40 \times 90$	2	8	391	5	23.217	396	5	2.072	389	5	3.009
21	This study	$36 \times 90 \times 181$	2	10	312	6	89.849	319	6	5.916	292	4	5.066
22	This study	$37 \times 53 \times 107$	2	15	370	3	24.663	381	3	1.780	363	3	2.864
23	This study	$40 \times 100 \times 198$	2	15	348	3	44.049	356	3	2.475	344	3	4.123
24	Wu et al. (2004)	$50 \times 50 \times 120$	2	15	626	4	38.331	626	4	3.205	625	4	5.077

obtained and shown in Section 4.1 assure the success of the proposed HSAM. An analysis is performed and the results are displayed in this section to further verify the effectiveness and study how the addition of the mutation operator and stagnancy-monitoring mechanism to the SA affects the solution quality and efficiency. The following three algorithms are tested by using 24 test instances with various problem sizes, sixteen from the literature and eight randomly generated by this study, and the computational results are given in Table 4: the SA; the SA with stagnancy-monitoring counter; and the SA with both stagnancy-monitoring counter and mutation operator (i.e., the HSAM).

It can be observed from Table 4 that SA and SA with stagnancy-monitoring counter are tied in terms of the number of intercellular movements in the first 16 test instances, and differ slightly for the other 8 problems. However, SA with stagnancy-monitoring counter is much more efficient, for it only takes less than 10% of the run time consumed by ordinary SA. This confirms the great effectiveness of the stagnancy-monitoring mechanism in HSAM. We next proceed to the comparisons of SA with stagnancy-monitoring counter only and the SA with both stagnancy-monitoring counter and mutation operator (i.e., the HSAM). The number of exceptional elements obtained by the SA with both stagnancy-monitoring counter and mutation operator are consistently better than or equal to those of SA with stagnancy-monitoring counter only in all 24 test problems. The dominance becomes even more obvious when problems with larger sizes are solved. The run time data shows that both approaches are very efficient. Even the largest problem can be solved in less than 6 s. A run time comparison of the three algorithms is given in Fig. 7. The great effectiveness and necessity of adding the stagnancy-monitoring counter and mutation operator to the HSAM can thus be assured by the computational results.

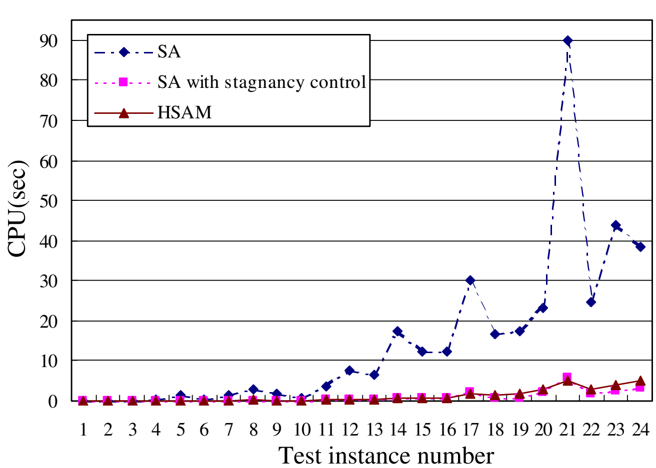

Fig. 7. Run time comparison of three algorithms.

5. Concluding remarks

A hybrid SA algorithm with a GA mutation operator for the cell formation problem considering alternative process routings, HSAM, has been proposed in this research. It is anticipated that complementary strengths and synergy effects of both the GA and the SA can be realized to increase the quality and efficiency of solutions. Considerable effort has been devoted to the design of a procedure to assign a routing for each part to machine cells. Computational results indicate the solution of this procedure is superior to those appearing in the literature, which adopt the maximum density rule. In the solution improvement stage of the proposed algorithm, the insertion-move has been utilized iteratively to guide the solution search. In addition, several counters have been used and collocated with the insertion-move in the algorithm to speed up the solution search process and to escape from the local optima.

Preexisting and newly generated test problems have been used to verify the proposed algorithm. Computational results obtained from running fourteen test instances from the literature have shown that HSAM improves the best values for the number of exceptional elements found in the open literature, i.e., the TS-1, TS-2, SA, P0, MP, MP2, TSPA, and BS for four (28.6%) problems; and that for ten (71.4%) problems, HSAM obtains values for the number of exceptional elements that are equal to the best results found in the aforementioned 7 methods. In test problem #12, the percentage improvement of HSAM is higher than 18%; the improvement even reaches 40.9% in test problem #14. Dominance of HSAM over other approaches becomes even more significant for problems with larger sizes.

In addition to minimization of the intercellular movements, HSAM is capable of generating grouping plans maximizing the grouping efficacy as well, by a very minor revision on HSAM. It is hence suggested that HSAM maximizing the grouping efficacy (HASM2) be applied after HSAM minimizing the intercellular movements (HSAM1) has been used. If both approaches result in the same number of exceptional elements, then the solution produced by HSAM2 can be considered as a better decision alternative. Furthermore, the great effectiveness and necessity of adding the stagnancy-monitoring counter and mutation operator to the HSAM have been assured by the computational results shown in Table 4.

Appendix A. Solution matrices of problems surpassing the best results in literature

#8

Machine: 10 Part: 15 Route: 28

	4	6	9	11	14	3	12	15	1	2	5	7	8	10	13
	b	a	a	c	b	b	a	c	a	c	a	a	b	a	a
2	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
6	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
8	0	1	1	1	1	0	0	1	0	0	0	0	0	1	1
10	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0
5	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0
7	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0
3	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1
4	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
9	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1

Cell: 3.
Total operations: 47.
Exceptional elements: 10.
Voids: 17.
Grouping efficacy: 57.8125.

#11

Machine: 14 Part: 20 Route: 45

	1	5	6	7	9	11	16	17	19	20	3	4	8	10	15	18	2	12	13	14
	a	a	a	a	a	c	a	a	a	a	b	a	b	b	b	a	b	b	b	a
2	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1
6	1	1	1	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0
8	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0
12	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	1
5	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
7	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	1
10	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	1
11	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	1	1	0
14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Cell: 3.
Total operations: 85.
Exceptional elements: 24.
Voids:35.
Grouping efficacy: 50.8333.

#12

Machine: 18 Part: 30 Route: 59

	5	8	11	24	6	7	10	14	16	17	19	20	21	22	23	26	27	28	1	2	3	4	9	12	13	15	18	25	29	30
	a	b	b	a	b	a	b	a	a	a	a	a	a	a	a	b	a	a	b	b	b	b	a	c	b	a	a	a	d	a
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
5	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
6	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	1	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
14	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
16	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	1	1	0	1
9	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0
11	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
13	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0
18	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0

Cell: 3.
Total operations: 116.
Exceptional elements: 26.
Voids: 108.
Grouping efficacy: 40.1786.

#14

Machine: 26 Part: 28 Route: 71

	1	7	9	14	19	24	27	4	10	15	17	20	22	23	25	2	**11**	18	5	12	13	16	21	3	6	8	26	28
	b	c	b	b	d	b	a	b	d	c	c	c	a	a	a	a	b	c	a	a	a	b	b	b	b	a	c	a
2	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0
3	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
24	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Cell: 5.
Total operations: 120.
Exceptional elements: 13.
Voids: 52.
Grouping efficacy: 62.2093.

References

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

Adil, G. K., Rajamani, D., & Strong, D. (1996). Cell formation considering alternate routeings. *International Journal of Production Research, 34*, 1361–1380.

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

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

Kasilingam, R. G., & Lashkari, R. S. (1991). Cell formation in the presence of alternate process plans in flexible manufacturing systems. *Production Planning and Control, 2*, 135–141.

Kumar, C. S., & 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*, 233–243.

Kusaik, 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.

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*, 4035–4047.

Logendram, R., Ramakrishna, P., & Sriskandarajah, C. (1994). Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research, 32*, 273–297.

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

Moon, Y. B., & Chi, S. C. (1992). Generalized part family formation using neural network techniques. *Journal of Manufacturing Systems, 11*, 149–159.

Nagi, R., Harlarakis, G., & Proth, J. M. (1990). Multiple routings and capacity considerations in group technology applications. *International Journal of Production Research, 28*, 2243–2257.

Sankaran, S., & Kasilingam, K. G. (1990). An integrated approach to cell formation and part routing in group technology. *Engineering Optimization, 16*, 235–245.

Selim, H. M., Askin, R. G., & Vakharia, A. J. (1998). Cell formation in group technology: Review, evaluation and directions for future research. *Computers and Industrial Engineering, 34*, 3–20.

Spiliopoulos, K., & Sofianopoulou, S. (2007). Manufacturing cell design with alternative routings in generalized group technology: Reducing the complexity of the solution space. *International Journal of Production Research, 45*, 1355–1367.

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

Soke, A., & Bingul, Z. (2006). Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems. *Engineering Applications of Artificial Intelligence, 19*, 557–567.

Won, Y. (2000). New p-median approach to cell formation with alternative process plans. *International Journal of Production Research, 38*, 229–240.

Won, Y. K., & Kim, S. H. (1994). An assignment method for the part-machine cell formation problem in the presence of multiple process routes. *Engineering Optimization, 22*, 231–240.

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.

Wong, S. Y. W. (2001). Hybrid simulated annealing/genetic algorithm approach to short-term hydro-thermal scheduling with multiple thermal plants. *International Journal of Electrical Power and Energy Systems, 23*, 565–575.

Wu, T.-H., Chen, J.-F., & Yeh, J.-Y. (2004). A decomposition approach to the cell formation problem with alternative process plans. *International Journal of Advanced Manufacturing Technology, 24*, 834–840.