Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 34 (2008) 1609–1617

Expert Systems with Applications

www.elsevier.com/locate/eswa

A simulated annealing algorithm for manufacturing cell formation problems

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

[a] *Department of Industrial Engineering and Systems Management, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung 407, Taiwan*

[b] *Department of Industrial Engineering and Management, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan*

Abstract

The cell formation problem determines the decomposition of the manufacturing cells of a production system in which machines are assigned to these cells to process one or more part families so that each cell is operated independently and the intercellular movements are minimized or the number of parts flow processed within cells is maximized. In this study, a simple yet effective simulated annealing-based approach, SACF, is proposed to solve the cell formation problem. Considerable efforts are devoted to the design of parts and machine assignment procedures to direct SACF to converge to solutions with good values of grouping efficacy. A set of 25 test problems with various sizes drawn from the literature is used to test the performance of the proposed heuristic algorithm. The corresponding results are compared to several well-known algorithms published. The comparative study shows that the proposed SACF algorithm improves the grouping efficacy for 72% of the test problems. The proposed algorithm should thus be useful to both practitioners and researchers.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Simulated annealing; Cell formation problem; Grouping efficacy

1. Introduction

To make manufacturing systems more efficient and productive, group technology (GT) has been applied within a manufacturing environment. GT groups parts with similar design characteristics or manufacturing characteristics into part families. One application of GT is cellular manufacturing (CM). A number of benefits arise from adopting CM, such as: reduced inventory, reduced capacity, reduced labor and overtime costs, shorter manufacturing lead times, faster response to internal and external changes such as machine failures, product mix and demand changes (Wemmerlov & Hyer, 1989). Information such as parts to be produced, process plans and machines to perform all the required operations is needed when designing CM. The entire production system is decomposed into production cells. Machines are then assigned to these cells to process one or more part families so that each cell is operated independently and so that the intercellular movements are minimized or the number of part flow processed within cells is maximized, i.e., parts do not have to move from one cell to the other for processing.

This cell formation process is one of the most important steps in CM. It becomes difficult to obtain optimal solutions in an acceptable amount of time, especially for problems with large sizes. Extensive research has been devoted to cell formation (CF) problems, with many methods having been proposed for identifying machine cells and part families. Many of them are developed on the basis of heuristic clustering techniques to obtain approximate solutions, but some of them may be far from optimum. The research of Moon and Kim (1999) takes into account the

[*] Corresponding author. Present address: Department of Business Administration, National Taipei University, 151, University Road, San Shia, Taipei 237, Taiwan. Tel.: +886 2 86746558; fax: +886 2 86715912.
E-mail addresses: taiwu@dyu.edu.tw, aggie94kimo@yahoo.com.tw (T.-H. Wu).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.012

process plans for parts and manufacturing factors such as production volume and cell size. Their process of forming manufacturing cells starts by collecting the above problem data and then converting it into a weighted graph representation in which the nodes and arcs represent machines and their relationships defined as the value of total part flow between machines, respectively. Similar representations have been used by Rajagopalan and Batra (1975), Harlalakis, Nagi, and Proth (1990), Vohra, Chen, Chang, and Chen (1990), and Wu and Salvendy (1993) with different problem concerns. Some of them (Rajagopalan & Batra, 1975; Harlalakis et al., 1990) consider the situation where the size of each cell and the number of cells to be formed has to be restricted, while some of them (Vohra et al., 1990; Wu & Salvendy, 1993) did not.

Due to their excellent performance in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic algorithms, simulated annealing, neural networks and tabu search make up another class of search methods that has been adopted to efficiently solve the CF problem and its variants with good results obtained. Sun, Lin, and Batta (1995) presented a short-term tabu search-based algorithm for solving the CF problem with the objective of minimizing the intercellular parts flows, while Wu, Low, and Wu (2004) maximizes the parts flow within cells using long-term tabu search-based algorithm. Aljaber, Baek, and Chen (1997) proposed a tabu search approach to deal with this problem by modeling it as a shortest spanning path problem with respect to both parts and machines. The resulting spanning paths for parts and machines are then decomposed into subgraphs representing machine groups and part families, respectively. Cheng, Gupta, Lee, and Wong (1998) formulated the CF problem as a traveling salesman problem (TSP) and proposed a solution methodology based on genetic algorithm, while Dimopoulos and Mort (2001) presented a hierarchical clustering approach based on genetic programming. Onwubulo and Mutingi (2001) developed a genetic algorithm, which accounts for inter-cellular movements and the cell-load variation. Conçalves and Resende (2004) presented a hybrid algorithm combining a local search and a genetic algorithm with very promising results reported.

The purpose of this study is to develop a procedure that is efficient and effective for obtaining machine–part groupings when the manufacturing system is represented by a 0–1 machine–part incidence matrix. Since simulated annealing (SA) has been applied to a number of combinatorial problems with fairly good results obtained, it was, hence, selected in this research as the basis for developing search methods for the CF problem. SA can be viewed as a process which attempts to move from the current solution to its neighborhood solutions resulting in better objective values. However, for solutions with worse objective values, they are accepted with a specified probability mainly to escape from the local optima in its search for the global optima. A set of 25 test problems with various sizes drawn from the literature is used to test the performance of the proposed heuristic algorithm. The corresponding results are compared to several well-known algorithms published.

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

2. Cell formation problem

Cell formation in a given 0–1 machine–part incidence matrix involves rearrangement of rows and columns of the matrix to create part families and machines cells. In this research, we attempt to determine a rearrangement so that the inter-cellular movement can be minimized and the utilization of the machines within a cell can be maximized. Two matrices shown in Fig. 1 are used to illustrate the concept. Fig. 1a is an initial matrix where no blocks can be observed directly. After rearrangement of rows and columns, two blocks can be obtained along the diagonal of the solution matrix in Fig. 1b.

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, 1986a) and the grouping efficacy (Kumar & Chandrasekharan, 1990) due to they are easy to implement. Grouping efficiency η is defined as follows:

$$\eta = q\eta_1 + (1 - q)\eta_2$$

where η_1 is the ratio of the number of 1's in the diagonal blocks to the total number of elements in the diagonal blocks of the final matrix, η_2 is number of 0's in the off-diagonal blocks to the total number of elements in the off-diagonal blocks of the final matrix, and q is a weight factor. For those 1's outside the diagonal blocks, they are called "exceptional elements"; while those 0's inside the diagonal blocks are called "voids".

a

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

b

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

Fig. 1. Rearrangement of rows and columns of matrix to create cells: (a) initial matrix and (b) matrix after rearrangement.

Although grouping efficiency has been used widely, it was argued for its low discriminating capability in some cases affected by the size of the matrix. To overcome this problem, Kumar and Chandrasekharan (1990) proposed another measure, the grouping efficacy Γ, and can be defined as:

$$\Gamma = \frac{e - e_0}{e + e_\mathrm{v}}$$

where e is the total number of 1's in the matrix; e_0 is the total number of exceptional elements; and e_v is the total number of voids. Grouping efficacy ranges from 1 to 0, with 1 being the perfect grouping. As grouping efficacy has been widely accepted in recent studies regarding CF problem, it is used as the performance measure for the proposed SA algorithm in this study.

3. Simulated annealing approach

Simulated annealing (SA) algorithm was originally proposed by Metropolis, Rosenbluth, and Teller (1953) to simulate the annealing process. SA starts with a high temperature. After generating an initial solution, it attempts to move from the current solution to one of its neighborhood solutions. The changes in the objective function values (ΔE) are computed. If the new solution results in better objective value, it is accepted. However, if the new solution yields worse value, it can still be accepted according to the probability function, $P(\Delta E) = \exp(-\Delta E/k_\mathrm{B}T)$, where k_B is Boltzmann's constant and T is the current temperature. This check is performed by first selecting a random number from $(0,1)$. If the value is less than or equal to the probability value, the new configuration is accepted; otherwise, it is rejected. By accepting worse solutions, SA can avoid being trapped on local optima. SA repeats this process L times at each temperature to reach the thermal equilibrium, where L is a control parameter, usually called the Markov chain length. The parameter T is gradually decreased by a cooling function as SA proceeds until the stopping condition is met.

3.1. Initial solution

Since the CF problem considers the grouping of parts and machines, an intuitive solution approach is to decompose the entire problem into two subproblems dealing with parts assignment and machines assignment, respectively. When parts assignment is firstly determined, followed by a proper assignment of machines, generation of an initial solution is hence completed.

3.1.1. Parts assignment

As in McAuley's research (McAuley, 1972), Jaccard's similarity measure is used to evaluate similarity between parts as an important index for assigning parts to cells in this subproblem. The similarity measure, denoted S_{ij}, is defined as: $S_{ij} = \frac{a_{ij}}{a_{ij}+b_{ij}+c_{ij}}$, where a_{ij} represents the number of machines processing both parts i and j; while b_{ij} is the number of machines processing part i but not part j, and c_{ij} is the number of machines processing part j but not part i. After calculating the similarity matrix for each pair of parts, we are now able to generate the initial parts assignment by using the following greedy rule: the higher similarity measure a pair of parts has, with the higher priority they should be placed in the same cell. This process is repeated until all parts have been assigned to cells.

An example is used to illustrate this process. Consider a sample machine–part matrix in Fig. 2a, the corresponding similarity matrix for parts is displayed in Fig. 2b. Suppose that there are two cells to be formed. The largest coefficient in the matrix of Fig. 2b is 1, indicating that parts 2 and 3 must be assigned to the same cell, say cell 1. We proceed with the second largest coefficient in the matrix, 0.5, appearing in pairs $(2,5)$ and $(3,5)$. Part 5 is thus assigned to cell 1 with parts 2 and 3. The remaining coefficient in the matrix is 0.33 in pair $(1,4)$. Since these two parts do not have any relationship with any parts in cell 1, together they should be assigned to the next cell, cell 2, as shown in Fig. 3. In the case when three cells are to be formed, randomly select one part from the pair with the least coefficient in the matrix and assign it to cell 3; the rest arrangement remains the same.

3.1.2. Machines assignment

Since the number of voids and exceptional elements are major components comprising the formula of grouping efficacy, procedures considering these two elements should very possibly generate solutions with good values of grouping efficacy for the CF problem. This provides the motivation for our research to design the machines assignment procedure below:

a

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

b

	P1	P2	P3	P4	P5
P1	-	0	0	0.33	0
P2		-	1	0	0.5
P3			-	0	0.5
P4				-	0
P5					-

Fig. 2. Sample machine–part matrix and corresponding similarity matrix for parts: (a) machine–part matrix and (b) similarity matrix for parts.

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

Fig. 3. Assignment of parts.

Step 1. Read the results of parts assignment.
Step 2. For each machine, find the cell to which the machine assignment will result in the least sum of number of exceptional elements and voids. If a tie happens, assign the machine to a cell with the least number of voids.
Step 3. Repeat *Step 2* until all machines have been assigned to cells.

Results of parts assignment shown in Fig. 3 is used to demonstrate the machines assignment procedure. Fig. 4 gives the sum of number of voids and exceptional elements for each machine–cell combination. Machines are then assigned to cells that result in the least sum. As a result, machines 2 and 4 are assigned to cell 1, while machines 1, 3, and 5 are assigned to cell 2. The initial solution matrix for the CF problem can thus be obtained and shown in Fig. 5.

3.2. Solution improvement

As mentioned in Section 3.1, when parts assignment is determined, machines assignment process follows. The solution quality of parts assignment, thus, plays a very critical factor in the success of entire solution quality. This section introduces strategies for searching better neighborhood solutions to improve the current solution and move toward the optimal solutions.

The neighborhood of a given solution is defined as a set of all feasible solutions that can be reached by a single move/transition. In this study, two types of moves are implemented interactively: (1) a single-move, and (2) an exchange-move. The single-move is an operation that moves a part j from its current cell i (source cell) to a new cell i' (destination cell). The new move made is denoted (i',j). For the single-move, a move that results in the most improvement in the objective function value from the current solution is selected. That is,

$$M_1(i1',j1)=\max\{\text{obj}^{(i',j)}-\text{obj}^{\text{current}},\ \forall i'\in I,\ i'\neq i,\ \forall j\in J\}$$

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

The exchange-move is an operation consists of two dependent single-moves. If a part j is moved from its source cell i to destination cell i' (first single-move), then one part j' ($j'\in J_{i'}=\{\textit{parts}\text{ assigned to cell }i'\}$) from the destination cell i' of the first move has to be moved to the source cell i of the first move (second single-move) in exchange. Two moves, (i',j) and (i,j'), are generated. For the exchange-move, the pair of moves resulting in the most improvement in the objective function value from the current solution is selected. That is,

$$M_2\{(i2',j2),(i2,j2')\}=\max\{\text{obj}^{(i',j),(i,j')}-\text{obj}^{\text{current}},\ \forall i'\in I,\ i'\neq i,\ \forall j\in J,\ \forall j'\in J_{i'}\}$$

	Cell 1	Cell 2
	v+e	v+e
M1	3+2	$\boxed{0+0}$
M2	$\boxed{0+0}$	2+3
M3	3+1	$\boxed{1+0}$
M4	$\boxed{1+0}$	2+2
M5	3+1	$\boxed{2+0}$

Fig. 4. Sum of voids and exceptional elements for each machine–cell combination.

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

Fig. 5. Initial solution matrix.

3.3. SA algorithm for CF problem

This section describes the proposed algorithm SACF in detail. It is evident that the number of cells to be formed will affect the grouping solutions obtained. In our algorithm, the number of cells resulting in the best grouping efficacy is generated automatically. However, the flexibility is preserved for users to specify the number of cells they prefer. In addition, several counters and indicators are used in the algorithm to speed up the solution searching process and/or escape from the local optima. Before we proceed to the algorithm, some notations are introduced first.

S	current solution
S^c	neighborhood solution
S^*	best solution found in current number of cells
S^{**}	best solution found so far
T_0	initial temperature
T_f	final temperature
α	cooling rate
L	Markov chain length
k	iteration number
C	initial number of cells
C^*	optimal number of cells
D	length of period for evoking exchange-move

The proposed algorithm SACF can be summarized as follows.

Algorithm SACF

Step 1. Generate an initial solution S by using parts assignment and machines assignment procedures in Section 3.1. Set $C=2$, $S^{**}=S^*=S$, $C^*=C$.

```
Step 2. Initialize SA and other parameters: $T_0$, $T_f$, $\alpha$, $L$, $D$, $counter = 0$, $counter\_MC = 0$, $counter\_trapped = 0$, $counter\_stagnant = 0$.
Step 3. If $counter\_MC < L$ and $counter\_trapped < L/2$, then repeat Steps 3.1 to 3.7:
    Step 3.1. Generate a new parts assignment plan through neighborhood searching by performing single-move.
    Step 3.2. Perform exchange-move move if counter equals to a multiple of $D$.
    Step 3.3. Read parts assignment from above steps and generate corresponding machines assignment using procedure in Section 3.1.2 and thus the neighborhood solution $S^c$.
    Step 3.4. If $f(S^c) > f(S^*)$, then $S^* = S^c$, $S = S^c$, $counter\_stagnant = 0$, $counter\_MC = counter\_MC + 1$, go to Step 3.
    Step 3.5. If $f(S^c) = f(S^*)$, then $S = S^c$, $counter\_stagnant = counter\_stagnant + 1$, $counter\_MC = counter\_MC + 1$, go to Step 3.
    Step 3.6. Compute $\Delta = f(S^c) - f(S)$. Select a random variable $X \sim U(0,1)$. If $e^{\frac{\Delta}{T}} > X$, set $S = S^c$, $counter\_trapped = 0$; otherwise, $counter\_trapped = counter\_trapped + 1$.
    Step 3.7. $counter\_MC = counter\_MC + 1$, go to Step 3.
Step 4. If $T_k \leqslant T_f$ or $counter\_stagnant \geqslant check$, go to Step 5; otherwise, $T_{k+1} = T_k \times \alpha$, $counter\_MC = 0$, $counter = counter + 1$, go to Step 3.
Step 5. If $f(S^*) > f(S^{**})$, then $f(S^{**}) = f(S^*)$, $S^{**} = S^*$, $C^* = C$, $C = C + 1$, go to Step 2; otherwise report the current $f(S^{**})$, $S^{**}$, $C^*$, and stop the algorithm.
```

Note that SACF consists of an SA procedure that is repeatedly applied until a cell size resulting in the best grouping efficacy has been found. Initial number of cells is set at 2 in *Step 1*, and is gradually increasing by 1 at a time as long as solution improvement is observed in *Step 5*. All algorithmic parameters and counters are initialized in *Step 2*. In addition to the Markov chain length, normally used to assure the thermal equilibrium is reached in each temperature, another counter recording the number of times a solu-

Table 1
Levels for each parameter

Parameter	Level 1	Level 2	Level 3
T_0	10	30	50
α	0.7	0.8	0.9
L	10	30	70
D	6	12	18

Table 2
Results of experimental analysis on all parameter combinations

T_0	α	L	D					
			6		12		18	
			Ratio	CPU time (s)	Ratio	CPU time (s)	Ratio	CPU time (s)
10	0.7	10	1.064	5.077	1.065	5.206	1.061	3.945
		30	1.066	6.489	1.066	5.476	1.065	5.250
		70	1.065	10.106	1.064	8.253	1.066	7.894
	0.8	10	1.067	6.868	1.066	4.726	1.066	4.650
		30	1.066	8.421	1.063	5.748	1.066	4.767
		70	1.066	17.373	1.066	8.711	1.066	8.166
	0.9	10	1.067	6.691	1.065	5.136	1.063	5.386
		30	1.065	9.902	1.067	6.244	1.067	6.382
		70	1.064	10.012	1.065	8.635	1.065	7.728
30	0.7	10	1.066	6.471	1.066	4.523	1.065	4.528
		30	1.067	7.758	1.065	5.693	1.066	5.052
		70	1.068	12.410	1.066	8.000	1.065	11.124
	0.8	10	1.064	5.272	1.066	4.641	1.060	4.377
		30	1.066	8.237	1.067	5.478	1.062	5.438
		70	1.067	13.025	1.068	9.169	1.064	10.446
	0.9	10	1.066	5.834	1.065	4.873	1.066	5.384
		30	1.065	7.220	1.067	7.215	1.067	6.939
		70	1.065	14.258	1.067	10.446	1.064	8.817
50	0.7	10	1.065	5.557	**1.068**	5.442	1.059	4.118
		30	1.067	7.678	1.067	6.782	1.067	6.756
		70	1.063	9.672	1.068	11.191	1.065	7.553
	0.8	10	1.064	6.770	1.060	4.189	1.063	4.614
		30	1.066	7.389	1.066	5.974	1.068	6.182
		70	1.065	10.321	1.066	9.325	1.067	10.428
	0.9	10	1.065	8.607	1.065	5.038	1.061	4.830
		30	1.068	8.936	1.065	6.284	1.062	5.385
		70	1.067	13.849	1.065	8.651	1.063	8.933

tion fails in the Boltzmann's probability test is used in *Step 3* to avoid being trapped in local solutions and causing too much computational effort wasted in certain temperatures. Two types of moves, the single and exchange-move, namely, are utilized interactively in the proposed algorithm to guide the solution searching. Both moves play different roles in the process of solution improvement. From our experience of intensive testing, it is observed that single-move usually leads to better solutions smoothly and efficiently, but only up to certain point. Frequent use of exchange-move, however, brings too much disturbance to solution searching with too much computational effort spent. We hence use single-move as a primary tool for finding better neighborhood solution in *Step 3.1*, but employ exchange-move to add some disturbance to current solution every certain period, D, in *Step 3.2* to increase the probability of finding more "diversified" solutions to bring the searching process to a new and unexplored solution space. SACF also records the number of times when neighborhood solutions become stagnant. When this number reaches a pre-specify constant *check* in *Step 4*, a solution check is performed by comparing the grouping efficacy of current cell size to the best solution found so far in *Step 5* to determine whether to increase the cell size by 1 and continue the procedure or report the best solution found and terminate SACF. After intensive testing, the value of *check* is set at 4 in this study.

For users having their preferences in cell size, the proposed algorithm can save lots of run time since it will skip the process of iteratively searching for the cell size resulting in the best grouping efficacy. The savings in run time become even more significant as the cell size increases.

4. Computational results

In this section, 25 test problems from the literature are used to evaluate the computational characteristics of the proposed heuristic SACF, and the results are compared with those of algorithms reported in the literature, i.e., the ZODIAC (Chandrasekharan & Rajagopalan, 1987), the TSP-GA (Cheng et al., 1998), and the GA (Onwubulo & Mutingi, 2001). The matrices of the test problems range from 5×7 to 40×100 and comprise well-structured and unstructured matrices. The proposed algorithm SACF was coded in *C* and implemented on a Pentium III 933 MHz personal computer with 256 MB RAM.

4.1. Parameter settings

As widely known, settings of SA parameters critically affect the solution efficiency and effectiveness. An experiment regarding the setting of five parameters appeared in SACF is firstly conducted.

The five SA parameters appeared in SACF: T_0, T_f, α, L, and D represents the starting temperature, final temperature, cooling rate, Markov chain length, and length of period for evoking exchange-move, respectively. Except T_f was set at 0.002, three levels are chosen and given in Table 1 for each parameter.

Six test problems (#2, #7, #10, #12, #21, #24) representing various problem sizes are selected and used in the experimental analysis of parameters. Due to the stochastic features of the proposed method might have, five independent runs were performed on each parameter combination for each test instance. The ratios of our results to the best of ZODIAC, TSP-GA, and GA, are calculated and the average ratios are given in Table 2. From Table 2, it can be observed that all parameter combinations perform quite well and do not differ much in terms of solution quality (more than 6% improvement over the best results of ZODIAC, TSP-GA, and GA). Since the parameter combination ($T_0 = 50$, $\alpha = 0.7$, $L = 10$, $D = 12$) produces the best improvement in a relatively efficient manner, we decide to use it as the suggested parameter setting for use in the next section.

4.2. Results

The parameter setting obtained in Section 4.1 is used by *SACF* to run all the test instances in this section. Fig. 6 and

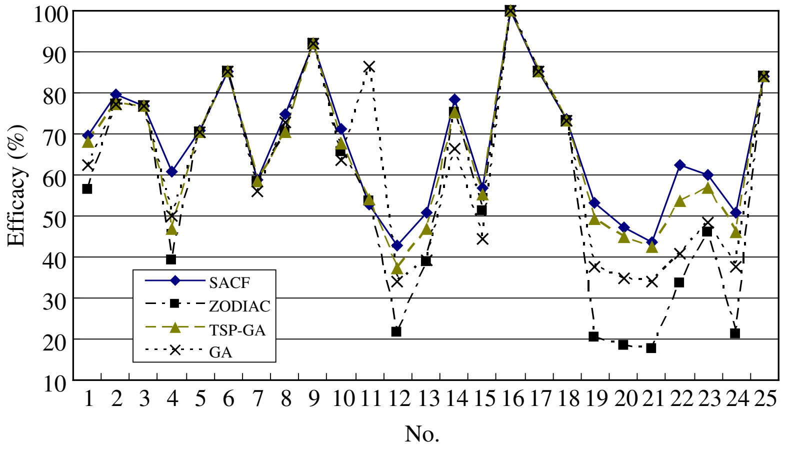

Fig. 6. Results of SACF, ZODIAC, TSP-GA, and GA on 25 test instances.

Table 3
Performance of SACF compared to ZODIAC, TSP-GA, and GA

No.	Test instances	Size	Other approaches			Best of the three approaches	Proposed approach SACF				
	Source		Grouping efficacy (%)				Grouping efficacy (%)			Cell size	CPU time (s)
			ZODIAC	TSP-GA	GA		Max.	Avg.	Std.		
1	Waghodekar and Sahu (1984)	5 × 7	56.52	68.00	62.50	68.00	69.57	69.57	0.00	2	0.002
2	Seifoddini (1989)	5 × 18	77.36	77.36	77.36	77.36	79.59	79.59	0.00	2	0.014
3	Kusiak and Cho (1992)	6 × 8	76.92	76.92	76.92	76.92	76.92	76.92	0.00	2	0.002
4	Kusiak and Chow (1987)	7 × 11	39.14	46.88	50.00	50.00	60.87	58.84	1.01	5	0.018
5	Boctor (1991)	7 × 11	70.37	70.37	70.37	70.37	70.83	69.13	1.39	4	0.012
6	Chandrashekharan and Rajagopalan (1986a)	8 × 20	85.24	85.24	85.24	85.24	85.25	85.25	0.00	3	0.026
7	Chandrashekharan and Rajagopalan (1986b)	8 × 20	58.33	58.33	55.91	58.33	58.41	58.41	0.00	2	0.024
8	Mosier and Taube (1985a)	10 × 10	70.59	70.59	72.79	72.79	75.00	71.49	2.87	5	0.016
9	Chan and Milner (1982)	10 × 15	92.00	92.00	92.00	92.00	92.00	92.00	0.00	3	0.018
10	Stanfel (1985)	14 × 24	65.55	67.44	63.48	67.44	71.21	69.38	1.59	8	0.449
11	King (1980)	16 × 43	53.76	53.89	86.25	86.25	52.44	52.44	0.00	5	1.095
12	Mosier and Taube (1985b)	20 × 20	21.63	37.12	34.16	37.12	41.04	41.02	0.04	6	0.351
13	Kumar et al. (1986)	20 × 23	38.66	46.62	39.02	46.62	50.81	47.05	1.99	7	0.597
14	Carrie (1973)	20 × 35	75.14	75.28	66.30	75.28	78.40	77.78	1.23	5	1.214
15	Boe and Cheng (1991)	20 × 35	51.13	55.14	44.44	55.14	56.04	56.04	0.00	4	0.789
16	Chandrasekharan and Rajagopalan (1989)	24 × 40	100.00	100.00	100.00	100.00	100.00	100.00	0.00	7	1.568
17	Chandrasekharan and Rajagopalan (1989)	24 × 40	85.11	85.11	85.11	85.11	85.11	85.11	0.00	7	1.819
18	Chandrasekharan and Rajagopalan (1989)	24 × 40	73.03	73.03	73.03	73.03	73.51	73.51	0.00	7	1.512
19	Chandrasekharan and Rajagopalan (1989)	24 × 40	20.42	49.37	37.62	49.37	52.44	52.44	0.00	8	3.405
20	Chandrasekharan and Rajagopalan (1989)	24 × 40	18.23	44.67	34.76	44.67	47.13	45.59	1.13	9	5.828
21	Chandrasekharan and Rajagopalan (1989)	24 × 40	17.61	42.50	34.06	42.50	44.64	43.81	0.60	9	5.005
22	Kumar and Vannelli (1987)	30 × 41	33.46	53.80	40.96	53.80	62.42	61.08	1.41	13	9.626
23	Stanfel (1985)	30 × 50	46.06	56.61	48.28	56.61	60.12	59.88	0.21	13	15.440
24	Stanfel (1985)	30 × 50	21.11	45.93	37.55	45.93	50.51	49.60	0.68	11	17.591
25	Chandrasekharan and Rajagopalan (1989)	40 × 100	83.92	84.03	83.90	84.03	84.03	84.03	0.00	10	106.934

Table 3 show the computational results of SACF and published results in the literature including ZODIAC, TSP-GA, and GA for the 25 test instances. According to Table 3, results obtained by SACF are better than or equal to those reported results except in problem #11. To be more specific, SACF obtains for 6 (24%) problems values of the grouping efficacy that are equal to the best results found in ZODIAC, TSP-GA, and GA methods and improves the values of the grouping efficacy for the rest 18 (72%) problems. In 3 (12%) problems, i.e., problem #4, #12, and #22, the percentage improvement is higher than 10%, with the highest being 21.74%, appeared in problem #4. Since five replicates are performed for each test instance, the best, average and standard deviation values of grouping efficacy are listed in Table 3 as well. The standard deviation is 0, in 13 out of 25 problems, and the largest value is never greater than 2.87. The very low standard deviation indicates that SACF is not only able to produce good solutions but also can be considered as a robust heuristic algorithm. In addition, the cell size resulting in the best grouping efficacy for each test problem is also given in Table 3. As to run time data, it ranges from 0.002 s to 106.934 s depending on different problem sizes.

5. Concluding remarks

A simple yet effective approach for the cell formation problem, *SACF*, has been proposed in this research. Considerable efforts have been devoted to the design of (1) parts assignment procedure in which part families are formed through the construction of similarity matrix of parts, and (2) machine assignment procedure, in which the number of voids and exceptional elements, major components comprising the formula of grouping efficacy, are explicitly considered. We believe this explicit consideration of number of voids and exceptional elements in the machine assignment procedure has directed SACF to converge to solutions with good values of grouping efficacy. In the solution improvement stage, two types of moves, the single and exchange-move, namely, have been utilized interactively and collocate properly in the proposed algorithm to guide the solution searching. In addition, several counters and indicators have been used in the algorithm to speed up the solution searching process and/or escape from the local optima.

Computational results obtained from running a set 25 test instances from the literature have shown that SACF improves the best values of the grouping efficacy found in ZODIAC, TSP-GA, and GA methods for 18 (72%) problems, and obtains for 6 (24%) problems values of the grouping efficacy that are equal to the best results found in ZODIAC, TSP-GA, and GA. In 3 (12%) problems, the percentage improvement is higher than 10%, with the highest being 21.74%.

Although SACF is able to find the number of cells that can result in the best grouping efficacy, the process of iteratively searching for the cell size, however, consumes too much run time. Developing more effective methods for finding proper cell sizes may thus be regarded as a future research.

References

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

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

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

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

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

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

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

Chandrasekharan, M. P., & Rajagopalan, R. (1987). ZODIAC – An algorithm for concurrent formation of part families and machine cells. *International Journal of Production Research, 25*(6), 835–850.

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

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

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

Dimopoulos, C., & Mort, N. (2001). A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. *International Journal of Production Research, 39*, 1–19.

Harlalakis, J., Nagi, R., & Proth, J. M. (1990). An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research, 28*, 185–198.

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

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.

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

Kumar, K. R., & Vannelli, A. (1987). Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research, 25*(12), 1715–1728.

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

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

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

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

Moon, C., & Kim, J. (1999). Genetic algorithm for maximizing the parts flow within cells in manufacturing cell design. *Computers and Industrial Engineering, 36*, 379–389.

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

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

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

Rajagopalan, R., & Batra, J. (1975). Design of cellular production systems-a graph theoretical approach. *International Journal of Production Research, 13*, 256–268.

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

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

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

Vohra, T., Chen, D., Chang, J., & Chen, H. (1990). A network approach to cell formation in cellular manufacturing. *International Journal of Production Research, 28*, 2075–2084.

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

Wemmerlov, U., & Hyer, N. L. (1989). Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research, 27*(9), 1511–1530.

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.

Wu, N., & Salvendy, G. (1993). Modified network approach for the design of cellular manufacturing systems. *International Journal of Production Research, 31*, 1409–1421.