Int. J. Production Economics 120 (2009) 669–688

Contents lists available at ScienceDirect

Int. J. Production Economics

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

A hybrid heuristic algorithm adopting both Boltzmann function and mutation operator for manufacturing cell formation problems

Tai-Hsi Wu [a,*], Chin-Chih Chang [b], Jinn-Yi Yeh [c]

[a] Department of Business Administration, National Taipei University, 151, University Road, San Shia, Taipei 237, Taiwan
[b] Department of Information Management, Jen-Teh Junior College of Medicine, Nursing and Management, 79-9, Shijou Li, Houlong, Miaoli 356, Taiwan
[c] Department of Information Management, National Chiayi University, 300, Syuefu Road, Chiayi City 600, Taiwan

ARTICLE INFO

Article history:
Received 29 November 2006
Accepted 8 April 2009
Available online 8 May 2009

Keywords:
Cell formation problem
Boltzmann function
Mutation operator
Hybrid heuristic

ABSTRACT

This study proposes a hybrid heuristic algorithm employing both the Boltzmann function from simulated annealing and the mutation operator from the genetic algorithm to explore the unvisited solution region and expedite the solution searching process for the cell formation problem, so that grouping efficacy is maximized. Test problems drawn from the literature are used to test the performance of the proposed heuristic algorithm. The comparative study shows that the proposed algorithm improves the best results found in the literature for 36% of the test problems in the case when singletons solutions are allowed.

1. Introduction

Group technology groups parts with similar design characteristics or manufacturing characteristics into part families in order to make the manufacturing systems more efficient and productive. Cellular manufacturing is the implementation of group technology to the manufacturing process. By adopting cellular manufacturing, several benefits can be obtained such as reduced inventory, reduced capacity, reduced labor and overtime costs, reduced set-up times, reduced material handling cost, and faster response to internal and external changes such as machine failures, product mix, and demand changes (Wemmerlov and Hyer, 1989; Dimopoulos and Mort, 2001). The entire production system is decomposed into several production cells in cellular manufacturing. 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 parts processed within cells is maximized—i.e., parts do not have to move from one cell to the other for processing.

The process of forming manufacturing cells is one of the most important steps in cellular manufacturing. It becomes difficult to obtain optimal solutions in an acceptable amount of time, especially for large-sized problems. Extensive research has been devoted to cell formation (CF) problems, with many methods proposed for identifying machine cells and part families. Basically, these methods can be classified under the following five categories: array-based, hierarchical/non-hierarchical clustering, graph-based, mathematical programming, and heuristics/meta-heuristics. Array-based methods repeatedly rearrange the rows and columns of the machine-part matrix to obtain visible groupings of machines and parts until a satisfactory solution is found. These methods include the bond energy algorithm (BEA) by McCormick et al. (1972), the rank order clustering (ROC) algorithm by King (1980), ROC2 by King and Nakornchai (1982), the direct clustering algorithm (DCA) by Chan and Milner (1982), and the close neighbor algorithm (CAN) by Boe and Cheng (1991).

Hierarchical/non-hierarchical clustering methods use a measure of similarity or dissimilarity for the grouping of

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

0925-5273/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2009.04.015

machines or parts. Studies such as McAuley (1972), Mosier and Taube (1985b), Seifoddini and Wolfe (1987), and Yasuda and Yin (2001) are of this category. Yin and Yasuda (2006) gave an overview and discussion for similarity coefficients developed to date for use in solving the CF problem. In the graph-based approaches, the process of forming manufacturing cells starts by collecting the problem data and then converting it into a weighted graph representation in which the nodes represent machines and the arcs represent their relationships defined as the value of total part flow between machines. Some studies have considered the situation where the size of each cell and the number of cells to be formed have to be restricted, such as Rajagopalan and Batra (1975) and Harlalakis et al. (1990), while some have not, such as Vohra et al. (1990) and Wu and Salvendy (1993). Mathematical programming approaches have developed models that can incorporate complicated features of the CF problems, but usually become computationally intractable especially for large-sized problems. Kusiak (1987) presented a p-median model, while Boctor (1991) suggested a linear 0–1 formulation for the CF problem.

Due to their excellent performance in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic algorithms (GAs), simulated annealing, neural networks, and tabu search all make up another class of search methods that have been adopted to efficiently solve the CF problem and its variants with good results. Sun et al. (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 et al. (2004) maximized the parts flow within cells using a long-term tabu search-based algorithm. Aljaber et al. (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 were then decomposed into sub-graph representing machine groups and part families, respectively. Cheng et al. (1998) formulated the CF problem as a traveling salesman problem (TSP) and proposed a solution methodology based on a 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. Gonçalves and Resende (2004) offered 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 both the simulated annealing (SA) and genetic algorithm (GA) have been applied to a number of combinatorial problems with fairly good results, they have been selected in this research as the basis for developing search methods for the CF problem. SA can be viewed as a process that attempts to move from the current solution to its neighborhood solutions, resulting in better objective values. However, solutions with worse objective values are accepted with a specified probability, mainly to release them from local optima, so that they can search for the global optima. GA is developed on the basis of the evolution process that occurs in natural biology. Some individuals are selected as parents to generate offspring via the crossover operator. GA is operated through a repeat process of reproduction, evaluation, and selection until the stopping criterion is met. A mutation operator is usually applied to the individuals with certain probability to escape from local solutions and/or prevent premature convergence. It is used mainly to increase the diversity of the population and ensure an extensive search.

By accepting worse solutions from the neighborhood of the current solution through the Boltzmann function of SA, a limited degree of solution diversification can be achieved, while the mutation operator from the GA provides a higher degree of solution diversification. A proper collocation of both techniques should offer different degrees of solution diversification and thus be able to explore the unvisited solution region, exploit better solutions, escape from local optima, and expedite the solution searching process. We hence employ the Boltzmann function from SA, together with the mutation operator from the GA, in the proposed heuristic algorithm to increase the quality and efficiency of solution. Two sets of test problems with various sizes drawn from the literature are used to test the performance of the proposed heuristic algorithm: one set for the case when singletons solutions are allowed and the other set for the case when singletons solutions are not allowed. 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 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 a rearrangement of rows and columns of the matrix to create part families and machine 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. It is required that each machine in a cell should process at least one part assigned to that cell, and each part in a cell should be processed by at least one machine assigned to that cell. Although some articles in the literature did not stick to this requirements, all the solutions generated in this study must satisfy the above constraints. Fig. 1 gives a sample solution matrix for a CF problem, in which two blocks can be observed along the diagonal of the solution matrix.

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 and Rajagopalan, 1986a) and the grouping efficacy (Kumar and Chandrasekharan,

	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	1	0	0	0	1

Fig. 1. Sample solution matrix.

1990), because 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 the 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. Any 1 s outside the diagonal blocks are called "exceptional elements," and 0 s inside the diagonal blocks are called "voids."

Although grouping efficiency has been used widely, critics argue that in some cases the size of the matrix impairs its discrimination ability. To overcome this problem, Kumar and Chandrasekharan (1990) proposed another measure, the grouping efficacy Γ, which can be defined as:

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

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 the CF problem, it is used as the performance measure for the proposed hybrid algorithm in this study.

3. Proposed hybrid heuristic algorithm

For most heuristic algorithms, avoiding being trapped in local optima and preventing premature convergence are two major challenges. It is more or less the same for meta-heuristics such as the simulated annealing algorithm and the genetic algorithm.

The simulated annealing algorithm was originally proposed by Metropolis et al. (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 computed, and if the new solution results in a better objective value, then it is accepted. However, if the new solution yields a worse value, then it can still be accepted according to the 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 performed by first selecting a random number from (0, 1). If the value is less than or equal to the probability value, then the new solution is accepted; otherwise, it is rejected. By accepting worse solutions via the Boltzmann function, SA can avoid being trapped on local optima. The parameter T is gradually decreased by a cooling function as SA proceeds until the stopping condition is met.

GA is developed on the basis of the evolution process that occurs in natural biology. Some individuals are selected as parents to generate offspring via the crossover operator. All the individuals are then evaluated and selected on the basis of Darwin's concept of survival of the fittest. The process of reproduction, evaluation, and selection is repeated until the stopping criterion is met. A mutation operator is usually applied to those individuals with a certain probability to escape from local solutions and/or to prevent premature convergence. It is used mainly to increase the diversity of the population and ensure an extensive search.

The combination of both techniques should offer different degrees of solution diversification. We hence employ the Boltzmann function from SA, together with the mutation operator from the GA, in the proposed heuristic algorithm to increase the quality and efficiency of solution.

3.1. Initial solution

A large number of similarity coefficients methods (SCM) and rank order clustering (ROC) techniques have been proposed for the CF problem. In this study, due to their simplicity and easy implementation, both methods are used to generate quick initial solutions for later improvement by the proposed algorithm.

It is well known that decomposing an originally difficult problem into several sub-problems usually increases the problem-solving efficiency. Since the CF problem considers the grouping of parts and machines, an intuitive solution approach is to decompose the entire problem into two sub-problems dealing with the assignment of parts and machines, respectively. In the SCM method we design, parts assignment is determined in the first stage, followed by a proper assignment of machines in the second stage. The ROC method, however, forms the initial part families and machine cells in one procedure.

3.1.1. Similarity coefficients method

The first approach used for generating initial solutions is the SCM method. SCM usually consists of three steps: (1) form the initial machine-part incidence matrix; (2) select a similarity coefficient and compute similarity values between part pairs and construct a similarity matrix; (3) use a clustering rule to process the values in the similarity matrix and form part families.

3.1.1.1. 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 sub-problem. The similarity measure, denoted S_{ij}, is defined as $S_{ij} = a_{ij}/(a_{ij} + b_{ij} + c_{ij})$, where a_{ij} represents the number of machines processing both parts i and j; while b_{ij} is the number of

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 ; (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.

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, the higher priority its parts have for placement in the same cell. This process is repeated until all parts have been assigned to cells. Consider the sample machine-part matrix in Fig. 2(a), the corresponding similarity matrix for parts is displayed in Fig. 2(b). Assume two cells are to be formed. The largest coefficient in the matrix of Fig. 2(b) 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. Fig. 3 shows the parts assignment using the proposed greedy rule.

3.1.1.2. Machine assignment. Wu 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. In this study we adopt their approach in determining the machines assignment. The procedure is summarized as follows.

Step 1. Read the results of the parts assignment.

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

	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. 4. Initial solution matrix obtained by SCM.

The results of parts assignment shown in Fig. 3 are used to demonstrate the above procedure. After calculating the sum of the numbers of voids and exceptional elements for each machine-cell combination, it can be observed in Fig. 4 that 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 this CF problem can thus be obtained.

3.1.2. Rank order clustering method

The second approach used for generating initial solutions is the rank order clustering method. The ROC algorithm determines a binary value for each row and column, rearranges the rows and columns in descending order of their binary values, and then identifies clusters (King, 1980). The ROC algorithm is described below, in which m and n denote the number of machines and parts, respectively (Heragu, 1997).

Step 1. Assign binary weight $BW_i = 2^{n-i}$ to each row i of the machine-part incidence matrix.

Step 2. Determine the decimal equivalent DE of the binary value of each column j using the formula

$$DE_j = \sum_{i=1}^{n} 2^{n-i} a_{ij}.$$

Step 3. Rank the columns in decreasing order of their DE values. Break ties arbitrarily. Rearrange the columns based on this ranking. If no rearrangement is necessary, then stop; otherwise, go to step 4.

Step 4. For each rearranged column of the matrix, assign binary weight $BW_j = 2^{m-j}$.

Step 5. Determine the decimal equivalent of the binary value of each row i using the formula

$$DE_i = \sum_{j=1}^{m} 2^{m-j} a_{ij}.$$

Step 6. Rank the rows in decreasing order of their DE values. Break ties arbitrarily. Rearrange the rows based on this ranking. If no rearrangement is necessary, then stop; otherwise, go to step 1.

Consider the sample machine-part matrix in Fig. 2, while the processes of rearranging columns and rows using the ROC algorithm are shown in Figs. 5–7. The binary values in Fig. 7 indicate that no rearrangement is necessary. It is thus the initial solution matrix for the CF problem obtained by using the ROC algorithm, and it incidentally matches the one obtained by SCM in Fig. 4.

	P1	P2	P3	P4	P5	Binary weight
Column value	20	10	10	17	8	
M1	1	0	0	1	0	16
M2	0	1	1	0	1	8
M3	1	0	0	0	0	4
M4	0	1	1	0	0	2
M5	0	0	0	1	0	1

Fig. 5. Binary values and weights for the columns and rows of sample matrix.

	P1	P4	P2	P3	P5	Row value
Binary weight	16	8	4	2	1	
M1	1	1	0	0	0	24
M2	0	0	1	1	1	7
M3	1	0	0	0	0	16
M4	0	0	1	1	0	6
M5	0	1	0	0	0	8

Fig. 6. Rearrangement of columns of sample matrix in Fig. 5.

	P1	P4	P2	P3	P5	Binary weight
Column value	24	20	3	3	2	
M1	1	1	0	0	0	16
M3	1	0	0	0	0	8
M5	0	1	0	0	0	4
M2	0	0	1	1	1	2
M4	0	0	1	1	0	1

Fig. 7. Rearrangement of rows of matrix in Fig. 6.

3.2. Solution improvement

As mentioned in Section 3.1, after parts assignment has been determined, a simple procedure for assigning machines follows—i.e., the procedure for assigning parts actually leads the procedure for assigning machines in this study. The solution quality of parts assignment thus plays a very critical role in the success of the entire solution quality. This section introduces a type of move to search for better neighborhood solutions of the current parts assignment, in collaboration with the Boltzmann function with a constant temperature and a mutation operator for exploring more diversified solutions and escaping from local optima in order to improve the current parts assignment and move toward the optimal solutions.

3.2.1. Searching for neighborhood solutions

The neighborhood of a given solution is defined as the set of all feasible solutions reachable by a single move. This study implements an insertion-move. The insertion-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 is denoted (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:

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

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

3.2.2. Boltzmann function with constant temperature

In SA, new solutions with worse values can still be accepted according to the Boltzmann probability function, $P(\Delta E) = \exp(-\Delta E/T)$. The probability of accepting worse solutions decreases as temperature T decreases. In this study we adopt the concept of Boltzmann probability, but prefer constant probability of accepting worse solutions. This can be achieved by introducing the Boltzmann function with constant temperature. After intensive testing, it is discovered that setting the probability $P(\Delta E)$ at 0.7 gives very good results.

3.2.3. Mutation operator

Like other traditional neighborhood search techniques, the insertion-moves usually lead to better solutions smoothly and efficiently in the beginning of the solution process, but get trapped in local optima easily after a short while. The application of mutation enables the algorithm to explore unvisited regions and to generate new solutions better than currently best ones. Mutation provides a small amount of random search and helps ensure that no point in the solution space has a zero probability of being examined.

In this study the implementation of the mutation operator is similar to the traditional gene-by-gene mutation with small probability p. This means a mutation check is performed part by part on the incumbent solution of parts assignment. For each part, a random number from $(0, 1)$ is first drawn. If the value is greater than or equal to p, then the part stays in the current cell; otherwise it is moved to another cell that is randomly selected.

3.3. Proposed algorithm BMCF

This section describes the proposed hybrid algorithm BMCF 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 are used in the algorithm to detect situations such as stagnancy and being trapped in local solutions. Strategies for coping with these situations are designed as well. Before proceeding to the proposed algorithm BMCF, we introduce some notations.

NC	number of cells (cell size)
n	number of times the mutation operator is performed

k_0 initial solution of parts assignment obtained by Jaccard's similarity (k_{0_SCM}) or rank order clustering method (k_{0_ROC})
k current solution of parts assignment
k' neighborhood solution of parts assignment
k^* incumbent solution of parts assignment of current cell size
k^{**} best solution of parts 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
$E(k,m)$ grouping efficacy of parts assignment k and machines assignment m
T temperature constant
r probability of accepting worse solutions
p mutation probability
C_1 number of times the neighborhood solution fails in the Boltzmann test
C_2 number of times mutation has been performed
C_3 number of times the neighborhood solution is equal to the incumbent solution
C_4 number of times the neighborhood solution is worse than the incumbent solution

The proposed algorithm *BMCF* can be summarized as follows:

Algorithm BMCF

Step 1. Let $NC=2$, $n=200$, $T=20$, $p=0.3$, $r=0.7$.
Step 2. Generate an initial solution of parts assignment k_0. Let $k=k_0$, $k^*=k_0$, $k^{**}=k_0$.
Step 3. Generate an initial solution of machines assignment m_0 based on initial parts assignment k_0. Let $m=m_0$, $m^*=m_0$, $m^{**}=m_0$.
Step 4. Parameters initialization: let $C_1=C_2=C_3=C_4=0$.
Step 5. If $C_2<n$ and $C_3<n$ and $C_4<n$ and $E(k^*,m^*)<100$, then repeat Steps 5.1 to 5.8; otherwise, go to Step 6.
Step 5.1. If $C_1<1$, then generate a new parts assignment k' through neighborhood searching by performing an insertion-move, go to Step 5.3.
Step 5.2. If $C_1\geq 1$, then apply mutation operator to k^* and generate a new parts assignment k'. Let $C_2=C_2+1$.
Step 5.3. Read parts assignment k' from the above steps and generate corresponding machines assignment m' using the procedure in Section 3.1.2.
Step 5.4. Let $\Delta E=E(k,m)-E(k',m')$. If $\Delta E<0$, then $k=k'$, $m=m'$, $C_1=0$.
Step 5.5. If $\Delta E>0$ and $\exp(-\Delta E/T)\geq r$, then $k=k'$, $m=m'$, $C_1=0$; otherwise $C_1=C_1+1$.
Step 5.6. If $E(k',m')=E(k^*,m^*)$, then $C_3=C_3+1$, go to Step 5.
Step 5.7. If $E(k',m')>E(k^*,m^*)$, then $k^*=k'$, $m^*=m'$, $C_3=C_4=0$, go to Step 5.
Step 5.8. If $E(k',m')<E(k^*,m^*)$, then $C_4=C_4+1$, go to Step 5.
Step 6. If $E(k^*,m^*)>E(k^{**},m^{**})$, then $E(k^{**},m^{**})=E(k^*,m^*)$, $k^{**}=k^*$, $m^{**}=m^*$, $NC=NC+1$, go to Step 4; otherwise report the current $E(k^{**},m^{**})$, k^{**}, m^{**}, $NC-1$, and stop the algorithm.

Note that BMCF consists of a procedure combining both the Boltzmann function with constant temperature in Step 5.5 and a mutation operator in Step 5.2, and it is repeatedly applied until a cell size resulting in the best grouping efficacy has been found. The initial number of cells is set at 2 in Step 1 and is gradually increased by 1 at a time as long as the solution improvement is observed in Step 6. The initial solutions of parts assignment and machines assignment are generated in Steps 2 and 3, respectively. Some counters incessantly detecting the solution searching process are initialized in Step 4. For a specific cell size, the best grouping plan for parts and machines will be calculated iteratively and obtained in steps 5.1–5.8.

Counter C_1 is used to record the number of times a solution fails in the Boltzmann's test to avoid being trapped in local solutions and wasting too much computational effort. As long as the value of C_1 is 0, a new neighborhood solution is generated through the insertion-move in Step 5.1; otherwise, mutation is applied in order to generate a new solution with a higher degree of diversification in Step 5.2. If the newly generated neighborhood solution is better than the current solution, then a replacement is made in Step 5.4. If the newly generated neighborhood solution is worse than the current solution, then the Boltzmann test with constant temperature is performed in Step 5.5. Before solution replacement is made, a check is performed to ensure that each machine in a cell should process at least one part assigned to that cell, and each part in a cell should be processed by at least one machine assigned to that cell. Any neighborhood solutions fail to satisfy the above constraints would be disregarded although it may have improving objective function values.

A comparison with the incumbent solution of current cell size follows next. The newly generated neighborhood solution replaces the incumbent solution if it results in larger grouping efficacy. If both have the same values of grouping efficacy, then counter C_3, monitoring the solution stagnancy, is increased by 1. The solution process repeats as long as counters C_2, C_3, and C_4 are all less than n, and the grouping efficacy of the incumbent solution is not yet a perfect grouping. When stopping criteria are met in Step 5, the incumbent solution obtained at this point represents the best solution of the current cell size. It is possible that better solutions may result in if larger cell sizes are considered. We thus compare the incumbent solution of the current cell size to the best solution found so far in Step 6 to determine whether to increase the cell size by 1 and continue the procedure or report the best solution found and terminate BMCF. The BMCF described above is for the situation when the SCM approach is used to generate initial solutions. After very minor modification, BMCF is applicable as well when the initial solution is generated by the ROC algorithm.

For users with specific preferences in cell size, the BMCF can save a lot of run time since it will skip the process of iteratively searching for the best cell size. The savings become even more significant as the cell size increases.

4. Computational results

Test instances from the literature are used to evaluate the computational characteristics of the proposed heuristic BMCF and the results are compared with those of the algorithms reported in the literature. Some studies allow the existence of singletons (cells having less than two parts or two machines) in the solutions and some do not. In order to make the comparisons fair and meaningful, the computational results are shown and discussed separately in two sub-sections. The matrices of the test problems range from 5×7 to 40×100, and they comprise well-structured and unstructured matrices. The proposed algorithm BMCF was coded in C and implemented on a Pentium III 933 MHz personal computer with 256 MB RAM.

4.1. Solutions allowing singletons

In the situation where singletons solutions are allowed, the similarity coefficients method in Section 3.1.1 for generating initial solutions was incorporated with the BMCF to produce the final solutions and is denoted SCM-BMCF in Table 1. Twenty-five test problems from the literature (Cheng et al., 1998; Onwubulo and Mutingi, 2001) were used to examine the performance of SCM-BMCF, and the results were compared with the best results found in the literature, i.e., the TSP-GA (Cheng et al., 1998), the GA (Onwubulo and Mutingi, 2001), and the SACF (Wu et al., 2008), where singletons are allowed. Because the proposed method might have stochastic features, 10 independent runs were performed for each test instance.

Table 1 and Fig. 8 compare the test results. From Table 1, it can be observed that the results obtained by SCM-BMCF are better than or equal to those reported results except in problem #11. The grouping efficacy of problem #11 obtained by GA is 86.25, which is far greater than the values obtained by the other approaches including TSP-GA, SACF, BMCF, and Hybrid-GA: all are around 55. The reported value 86.25 hence needs further confirmation.

To be more specific, SCM-BMCF improves the best values of the grouping efficacy found in the TSP-GA, GA, and SACF methods for 9 (36%) problems (#7, #10, #12, #15, #19, #20, #21, #22, #24); while for 15 (60%) problems, SCM-BMCF obtains values of the grouping efficacy that are equal to the best results of the TSP-GA, GA, and SACF methods. Since 10 replicates were performed for each test instance, the best, average and standard deviation values of grouping efficacy are listed in Table 1 as well. The standard deviation is 0 in 13 out of 25 problems, and the largest value is never greater than 0.85. This very low standard deviation indicates that BMCF is able to consistently produce good solutions. In addition, the cell size resulting in the best grouping efficacy for each test problem is also given in Table 1 in comparison with the ones obtained by the SACF. Only an extremely short amount of run time is needed for the SCM-BMCF to acquire the best solutions. The run time ranges from 0.007 to 7.235 s depending on different problem sizes. Since the computational environment in this study (933 MHz) is same as that of SACF in Wu et al. (2008), the dominance of BMCF over SACF in the computational efficiency can be easily observed when the problem sizes become large. For problems with grouping efficacies surpassing the results of TSP-GA, GA, and SACF, the corresponding solution matrices are presented in Appendix A.

4.2. Solutions not allowing singletons

BMCF checks on solutions every time a new part assignment or a new machine assignment is generated if the singletons restriction is considered. Any new solutions violating this restriction are disregarded. Thirty-five test problems from the literature (Gonçalves and Resende, 2004) were used to examine the performance of BMCF, and the results were compared with the best results found in the literature—i.e., Hybrid-GA of Gonçalves and Resende (2004), where singletons are not allowed. In addition to SCM-BMCF, another method of generating initial solutions, the ROC method, was incorporated with the BMCF and is denoted ROC-BMCF. Very minor modification to the program code of SACF (Wu et al., 2008) was made as well to cope with the singletons restriction. Table 2 gives the computational results of these methods. As in the case where singletons are allowed, 10 independent runs were performed for each test instance due to the stochastic features that BMCF and SACF might have.

Note that in test instance #12, row 22 is missing in Table 2 of the original paper (Askin and Subramanian, 1987), it hence is a 14×23, not a 14×24 matrix. Similarly, in test instance #33, all elements in rows 3, 4, 8, 10, 12, and 24 are 0 in Fig. 5 of the original paper (King and Nakornchai, 1982), it hence is a 30×90, not a 36×90 matrix.

Table 2 shows that SCM-BMCF produced best solutions in 31 out of 35 problems, while Hybrid-GA found 29, ROC-BMCF found 26, and SACF found 18. Initial solutions generated by the SCM are better than those of the ROC method in most of the test instances and some of them are very close to or even equal to the best solutions. In addition, SCM-BMCF provides better results than the Hybrid-GA. In 25 out of 35 test problems, both methods obtain the same results, while SCM-BMCF produced better results than those of Hybrid-GA in six problems (#26, #29, #30, #31, #33, #34) and Hybrid-GA has better results than those of SCM-BMCF in four problems (#7, #14, #16, #21). SCM-BMCF performs slightly better than Hybrid-GA especially in test problems with larger sizes. Both methods give the cell sizes resulting in the best grouping efficacy for each test problem. The cell sizes from both methods are the same in most of the test instances except in problems with larger sizes, where very minor

Table 1
Performance of SCM-BMCF compared to TSP-GA, GA, and SACF in the case where singletons are allowed.

Test instances			Other approaches					Proposed approach				
No.	Source	Size	TSP-GA	GA	SACF			SCM-BMCF				
			Γ	Γ	Γ	Cell size	Average CPU time	Max	Average	Standard deviation	Cell size	Average CPU time
1	Waghodekar and Sahu (1984)	5×7	68.00	62.50	69.57	2	0.002	69.57	69.57	0.00	2	0.007
2	Seifoddini (1989)	5×18	77.36	77.36	79.59	2	0.014	79.59	79.59	0.00	2	0.020
3	Kusiak and Cho (1992)	6×8	76.92	76.92	76.92	2	0.002	76.92	76.92	0.00	2	0.007
4	Kusiak and Chow (1987)	7×11	46.88	50.00	60.87	5	0.018	60.87	60.87	0.00	5	0.043
5	Boctor (1991)	7×11	70.37	70.37	70.83	4	0.012	70.83	70.83	0.00	4	0.035
6	Chandrashekharan and Rajagopalan (1986a)	8×20	85.24	85.24	85.25	3	0.026	85.25	85.25	0.00	3	0.045
7	Chandrashekharan and Rajagopalan (1986b)	8×20	58.33	55.91	58.41	2	0.024	58.72	58.57	0.43	2	0.032
8	Mosier and Taube (1985a)	10×10	70.59	72.79	75.00	5	0.016	75.00	75.00	0.00	5	0.049
9	Chan and Milner (1982)	10×15	92.00	92.00	92.00	3	0.018	92.00	92.00	0.00	3	0.037
10	Stanfel (1985)	14×24	67.44	63.48	71.21	8	0.449	71.83	71.58	0.42	7	0.312
11	King (1980)	16×43	53.89	86.25	52.44	5	1.095	56.38	55.63	0.84	7	0.992
12	Mosier and Taube (1985b)	20×20	37.12	34.16	41.04	6	0.351	43.26	43.14	0.15	5	0.342
13	Kumar et al. (1986)	20×23	46.62	39.02	50.81	7	0.597	50.81	50.81	0.00	7	0.392
14	Carrie (1973)	20×35	75.28	66.30	78.40	5	1.214	78.40	78.02	0.83	5	0.380
15	Boe and Cheng (1991)	20×35	55.14	44.44	56.04	4	0.789	57.61	57.53	0.09	5	0.412
16	Chandrasekharan and Rajagopalan (1989)	24×40	100.00	100.00	100.00	7	1.568	100.00	100.00	0.00	7	0.732
17	Chandrasekharan and Rajagopalan (1989)	24×40	85.11	85.11	85.11	7	1.819	85.11	85.11	0.00	7	1.053
18	Chandrasekharan and Rajagopalan (1989)	24×40	73.03	73.03	73.51	7	1.512	73.51	73.51	0.00	7	0.887
19	Chandrasekharan and Rajagopalan (1989)	24×40	49.37	37.62	52.44	8	3.405	53.29	52.82	0.32	11	1.993
20	Chandrasekharan and Rajagopalan (1989)	24×40	44.67	34.76	47.13	9	5.828	48.63	48.07	0.47	11	2.338
21	Chandrasekharan and Rajagopalan (1989)	24×40	42.50	34.06	44.64	9	5.005	46.15	45.19	0.36	12	3.332
22	Kumar and Vannelli (1987)	30×41	53.80	40.96	62.42	13	9.626	62.59	61.58	0.85	14	4.513
23	Stanfel (1985)	30×50	56.61	48.28	60.12	13	15.440	60.12	59.61	0.35	13	4.123
24	Stanfel (1985)	30×50	45.93	37.55	50.51	11	17.591	50.83	50.33	0.27	14	5.468
25	Chandrasekharan and Rajagopalan (1989)	40×100	84.03	83.90	84.03	10	106.934	84.03	84.03	0.00	10	7.235

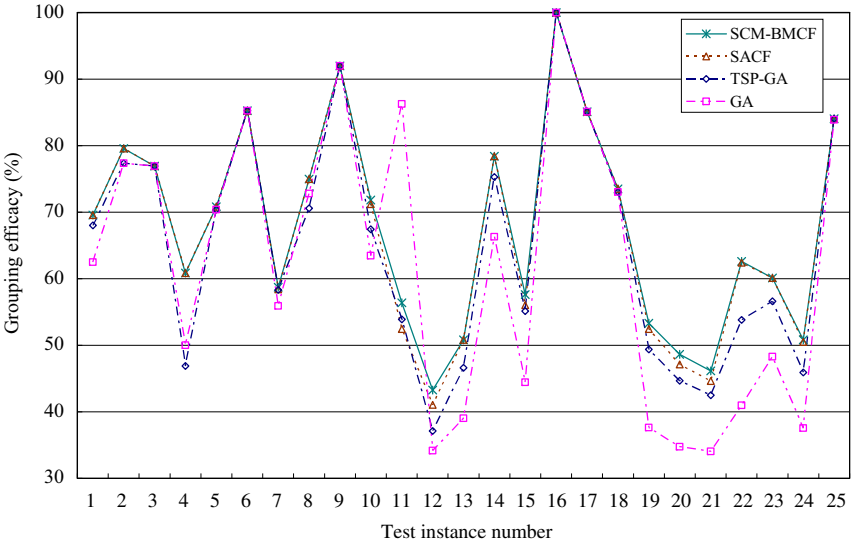


Fig. 8. Results of SCM-BMCF, SACF, TSP-GA, and GA on 25 test instances.

differences can be seen. Fig. 9 displays the solution results of the SACF, Hybrid-GA, and the proposed SCM-BMCF and ROC-BMCF methods, respectively.

Similar to the situation where singletons are allowed, SACF consumed much more runtime than the BMCF methods, especially in large-sized test problems. No significant difference is observed between SCM-BMCF and ROC-BMCF. Despite the computational environment in this study (933 MHz) being lower than that of Hybrid-GA (1.333 GHz), both SCM-BMCF and ROC-BMCF, however, take only a small fraction of the run time of Hybrid-GA to obtain comparatively favorable solutions. The superiority can be easily observed. For problems with grouping efficacies surpassing the results of Hybrid-GA, the corresponding solution matrices are presented in Appendix B.

In addition to the computational results that appear in Table 7 of their article, block-diagonal solution matrices for each test problem are provided as well in Gonçalves and Resende (2004). The grouping efficacies calculated from these block-diagonal matrices are also given in the column headed "appendix" of Table 2. In spite of this, the above comparison is made on the basis of the values reported in Table 7 of their article.

4.3. Further analysis

The excellent solution efficiency of the proposed BMCF method may be attributed to the proper combinations of four stopping criteria ($C_2 \geq n$, $C_3 \geq n$, $C_4 \geq n$, perfect grouping) in Step 5 of BMCF. BMCF terminates as long as one stopping criterion is met. This setting considers the trade-off between an improvement in solution qualities and additional, but possibly unnecessary, run time. The following experiment analyzes the setting of the stopping criteria for BMCF. After preliminary filtering, six alternatives of stopping the implementation of BMCF were selected:

1. Combinations of stopping criteria 1, 2, 3, and 4;
2. Combinations of stopping criteria 1, 3, and 4;
3. Combinations of stopping criteria 1, 2, and 4;
4. Combinations of stopping criteria 2 and 4;
5. Stopping criterion 3 only;
6. Stopping criterion 1 only.

The computational experiment was conducted by running the 35 test instances (10 replicates for each instance) in Section 4.2 using the above six alternatives as stopping criteria/criterion and recording both the average run time and average values of grouping for each test instance in Table 3. By observing the data in Table 3, it can be concluded that criterion 3 ($C_4 \geq n$) is the most active stopping criterion, and any alternative without adopting criterion 3 would be computationally inefficient. SC 1+2+4 and SC 1 are thus eliminated for further comparisons. No significant difference in the run time can be observed among the other four alternatives. Similarly, no significant difference among the other four alternatives can be observed when comparing the solution effectiveness. It is suggested in this study that SC 1+2+3+4 is used as the stopping criteria since this combination should provide more "ways out" to account for various problem characteristics than any single criterion that is considered.

5. Conclusions

This research has proposed a hybrid heuristic algorithm, BMCF, employing both the Boltzmann function from SA and the mutation operator from the GA, to explore the unvisited solution region and expedite the solution searching process for the cell formation problem. Due to their simplicity and easy implementation, the similarity coefficients method (SCM) and the rank order clustering (ROC) method have been used to generate quick initial solutions for later improvement by the BMCF. In the solution improvement stage, the insertion-move has been introduced for searching out better neighborhood solutions, in collaboration with the Boltzmann function with

Table 2
Comparisons of Performance of SACF, ROC-BMCF, and SCM-BMCF in the case where singletons are not allowed.

Test instances			SACF (10 replicates)			Hybrid-GA (10 replicates)				ROC-BMCF (10 replicates)				SCM-BMCF (10 replicates)			
No.	Source	Size	Max	Cell size	CPU time	Max	Appendix	Cell size	CPU time	Initial	Max	Cell size	CPU time	Initial	Max	Cell size	CPU time
1	King and Nakornchai (1982)	5 × 7	73.68	2	0.00	73.68	73.68	2	0.53	73.68	73.68	2	0.00	70.00	73.68	2	0.00
2	Waghodekar and Sahu (1984)	5 × 7	62.50	2	0.00	62.50	62.50	2	0.47	56.52	62.50	2	0.00	56.52	62.50	2	0.00
3	Seifoddini (1989)	5 × 18	79.59	2	0.01	79.59	79.59	2	0.85	53.45	79.59	2	0.01	79.59	79.59	2	0.01
4	Kusiak and Cho (1992)	6 × 8	76.92	2	0.00	76.92	76.92	2	0.66	53.33	76.92	2	0.02	53.33	76.92	2	0.02
5	Kusiak and Chow (1987)	7 × 11	53.13	3	0.00	53.13	53.13	3	1.09	38.24	53.13	3	0.02	47.06	53.13	3	0.01
6	Boctor (1991)	7 × 11	70.37	3	0.02	70.37	70.37	3	1.35	62.07	70.37	3	0.02	65.52	70.37	3	0.02
7	Seifoddini and Wolfe (1986)	8 × 12	68.29	3	0.01	68.30	68.30	3	1.44	31.25	68.29	3	0.05	59.52	68.29	3	0.03
8	Chandrashekharan and Rajagopalan (1986a)	8 × 20	85.25	3	0.03	85.25	85.25	3	1.68	50.70	85.25	3	0.05	85.25	85.25	3	0.04
9	Chandrashekharan and Rajagopalan (1986b)	8 × 20	58.41	2	0.03	58.72	58.72	2	1.68	54.05	58.72	2	0.05	42.50	58.72	2	0.04
10	Mosier and Taube (1985a)	10 × 10	70.59	3	0.02	70.59	70.59	3	1.82	48.72	70.59	3	0.03	70.59	70.59	3	0.02
11	Chan and Milner (1982)	10 × 15	92.00	3	0.03	92.00	92.00	3	2.19	83.02	92.00	3	0.05	92.00	92.00	3	0.03
12	Askin and Subramanian (1987)	14 × 23	67.61	6	0.45	69.86	69.86	5	6.05	22.55	69.86	5	0.27	40.23	69.86	5	0.27
13	Stanfel (1985)	14 × 24	69.33	5	0.32	69.33	69.33	5	6.24	35.63	69.33	5	0.30	67.11	69.33	5	0.25
14	McCormick et al. (1972)	14 × 24	51.96	6	0.47	52.58	51.96	6	7.85	37.04	51.96	6	0.48	39.81	51.96	6	0.45
15	Srinivasan et al. (1990)	16 × 30	67.83	4	0.38	67.83	67.83	4	10.24	53.66	67.83	4	0.28	60.00	67.83	4	0.20
16	King (1980)	16 × 43	52.44	6	2.49	54.86	54.29	5	13.92	17.58	54.60	6	1.32	39.52	54.60	6	0.93
17	Carrie (1973)	18 × 24	53.64	6	0.73	54.46	54.46	6	11.34	29.06	54.46	6	0.41	33.63	54.46	6	0.46
18	Mosier and Taube (1985b)	20 × 20	42.34	5	0.36	42.96	42.96	5	10.89	32.61	42.96	5	0.38	31.06	42.96	5	0.32
19	Kumar et al. (1986)	20 × 23	49.65	5	0.66	49.65	49.65	5	12.03	31.79	49.65	5	0.33	37.93	49.65	5	0.28
20	Carrie (1973)	20 × 35	76.14	4	1.57	76.22	76.22	4	15.61	28.67	76.22	5	0.67	34.62	76.22	5	0.62
21	Boe and Cheng (1991)	20 × 35	56.04	4	1.48	58.07	56.68	5	16.38	23.00	57.61	5	0.61	20.81	57.61	5	0.67
22	Chandrasekharan and Rajagopalan (1989)	24 × 40	100.00	7	2.81	100.00	100.00	7	19.26	100.00	100.00	7	0.95	100.00	100.00	7	0.62
23	Chandrasekharan and Rajagopalan (1989)	24 × 40	85.11	7	3.07	85.11	85.11	7	25.28	61.01	85.11	7	1.41	64.97	85.11	7	1.07
24	Chandrasekharan and Rajagopalan (1989)	24 × 40	73.51	7	1.90	73.51	73.51	7	26.82	39.23	73.51	7	1.14	73.51	73.51	7	0.73
25	Chandrasekharan and Rajagopalan (1989)	24 × 40	51.88	9	5.83	51.97	51.85	9	26.48	27.12	51.97	10	2.40	35.93	51.97	10	2.36
26	Chandrasekharan and Rajagopalan (1989)	24 × 40	44.44	10	9.85	47.06	46.50	9	25.97	35.63	46.98	11	3.28	23.73	47.33	11	3.38
27	Chandrasekharan and Rajagopalan (1989)	24 × 40	44.24	9	8.49	44.87	44.85	9	26.06	30.51	44.85	9	2.51	35.93	44.87	10	2.85
28	McCormick et al. (1972)	27 × 27	53.19	3	0.42	54.27	54.27	4	25.90	49.22	54.27	4	0.35	32.84	54.27	4	0.48
29	Carrie (1973)	28 × 46	40.97	6	5.70	44.62	43.85	9	43.78	22.07	44.90	10	4.39	36.12	45.31	10	3.82
30	Kumar and Vannelli (1987)	30 × 41	58.58	10	13.48	58.48	56.05	11	43.00	31.82	59.52	10	3.53	32.22	59.52	10	3.25
31	Stanfel (1985)	30 × 50	60.00	12	22.43	59.66	59.77	12	52.45	26.54	60.00	12	4.79	54.75	60.00	12	5.29
32	Stanfel (1985)	30 × 50	50.51	11	34.65	50.51	50.51	11	48.97	26.09	50.25	11	5.06	32.13	50.51	11	6.09
33	King and Nakornchai (1982)	30 × 90	39.95	11	300.31	42.64	42.64	9	81.46	12.66	43.84	10	4.21	10.48	44.59	12	9.84
34	McCormick et al. (1972)	37 × 53	58.23	2	1.79	56.42	56.14	2	87.66	52.15	59.04	3	0.83	48.57	59.04	3	0.92
35	Chandrasekharan and Rajagopalan (1989)	40 × 100	84.03	10	250.75	84.03	84.03	10	152.13	39.66	84.03	10	10.47	54.13	84.03	10	8.52

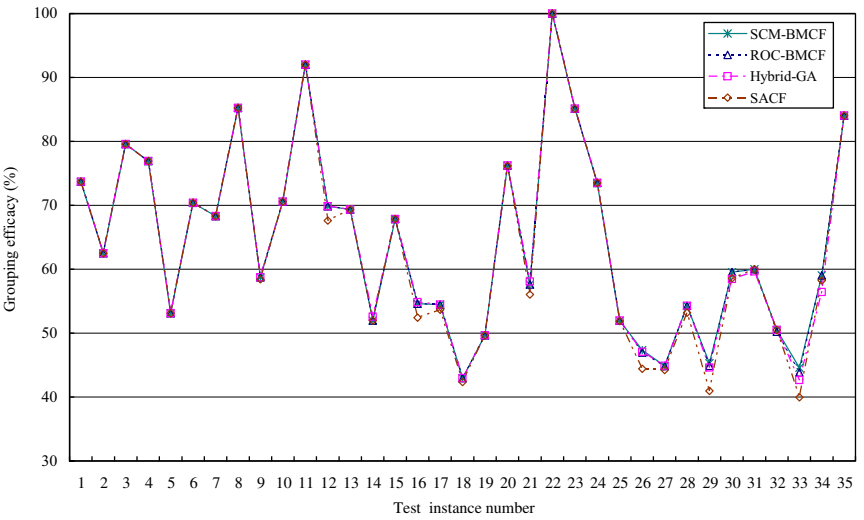


Fig. 9. Results of SCM-BMCF, ROC-BMCF, Hybrid-GA, and SACF on 35 test instances.

Table 3
Computational experiments of six alternatives of stopping criteria/criterion.

No.	SC 1+2+3+4		SC 1+3+4		SC 1+2+4		SC 3+4		SC 3		SC 1	
	Γ	Time	Γ	Time	Γ	Time	Γ	Time	Γ	Time	Γ	Time
1	73.68	0.004	73.68	0.003	73.68	0.016	73.68	0.003	73.68	0.004	73.68	0.019
2	62.17	0.004	62.34	0.005	62.50	0.017	62.34	0.005	62.34	0.005	62.50	0.011
3	79.59	0.010	79.59	0.006	79.59	0.043	79.59	0.006	79.59	0.009	79.59	0.052
4	76.92	0.010	76.92	0.014	76.92	0.026	76.92	0.013	76.92	0.011	76.92	0.020
5	53.13	0.017	53.13	0.015	53.13	0.059	53.13	0.014	53.13	0.017	53.13	0.066
6	70.37	0.019	70.37	0.019	70.37	0.087	70.37	0.024	70.37	0.014	70.37	0.117
7	68.29	0.034	68.29	0.035	68.29	0.176	68.29	0.030	68.29	0.029	68.29	0.192
8	85.25	0.042	85.25	0.043	85.25	0.268	85.25	0.042	85.25	0.041	85.25	0.373
9	58.62	0.041	58.62	0.041	58.68	0.183	58.59	0.041	58.56	0.049	58.59	0.206
10	70.59	0.022	70.59	0.027	70.59	0.115	70.59	0.028	70.59	0.040	70.59	0.157
11	92.00	0.032	92.00	0.041	92.00	0.245	92.00	0.037	92.00	0.051	92.00	0.281
12	67.88	0.276	67.79	0.262	68.83	1.436	68.31	0.224	67.79	0.257	68.83	1.840
13	69.33	0.247	69.33	0.248	69.33	1.505	69.33	0.235	69.33	0.243	69.33	1.985
14	51.67	0.397	51.86	0.513	51.96	2.533	51.91	0.554	51.58	0.471	51.96	2.833
15	67.83	0.203	67.83	0.218	67.83	1.934	67.83	0.219	67.83	0.297	67.83	3.422
16	53.69	0.950	53.69	0.934	53.91	5.572	53.69	0.654	53.69	0.982	53.91	6.837
17	54.27	0.515	54.46	0.437	54.40	2.775	54.20	0.558	53.89	0.466	54.46	3.095
18	42.70	0.372	42.90	0.288	42.93	2.914	42.70	0.401	42.69	0.331	42.95	3.050
19	49.65	0.309	49.65	0.309	49.65	3.450	49.65	0.259	49.65	0.364	49.65	3.195
20	76.22	0.71	76.19	0.642	76.22	3.875	75.89	0.642	75.44	0.673	76.22	5.624
21	57.28	0.703	56.70	0.560	57.61	5.683	57.06	0.543	56.51	0.553	57.61	6.666
22	100.00	0.626	100.00	0.701	100.00	5.162	100.00	0.716	100.00	1.255	100.00	12.459
23	85.11	1.167	85.11	0.905	85.11	10.609	85.11	0.957	85.11	1.143	85.11	18.880
24	73.51	0.729	73.51	0.934	73.51	7.286	73.51	1.012	73.51	1.050	73.51	12.773
25	50.76	2.523	51.48	2.129	51.69	16.868	51.20	1.980	51.12	2.230	51.80	19.145
26	46.51	2.731	45.77	2.916	46.65	22.529	46.49	3.338	46.08	2.899	46.84	20.925
27	44.14	3.217	43.76	2.382	44.44	16.827	43.73	2.710	44.02	3.127	44.44	17.123
28	53.76	0.479	53.98	0.315	54.00	6.119	53.42	0.398	53.81	0.459	54.21	7.826
29	44.55	3.120	44.61	2.543	44.69	17.572	44.32	2.984	44.62	2.308	45.13	13.162
30	57.99	3.913	57.81	3.973	59.21	24.879	57.12	2.866	57.55	3.772	59.29	24.850
31	59.91	5.216	59.93	5.006	60.00	41.220	59.93	4.149	59.97	4.928	60.00	48.509
32	49.23	4.966	49.55	4.711	50.40	42.084	49.65	6.335	49.16	5.454	50.38	32.592
33	38.45	9.920	34.48	5.643	42.76	66.982	38.72	9.892	37.89	7.422	41.37	67.283
34	58.54	1.210	58.61	0.692	59.04	9.972	58.56	0.873	58.3	0.963	59.04	11.437
35	84.03	10.139	84.03	9.703	84.03	133.683	84.03	10.143	84.03	8.174	84.03	199.877

constant temperature and a mutation operator for exploring more diversified solutions and escaping from local optima so as to improve the current parts assignment and move toward the optimal solutions. Proper combinations of four stopping criteria have been set to search for solutions with very good qualities without having to spend unnecessary computational efforts.

In the situation where singletons are allowed, computational results obtained from running a set of 25 test instances from the literature have shown that SCM-BMCF improves the values of the grouping efficacy for 9 (36%) problems, while for 15 (60%) problems, SCM-BMCF obtains values of the grouping efficacy that are equal to the best results found in the literature. As to the situation where singletons are not allowed, computational results obtained from running a set of 35 test instances from the literature have shown that SCM-BMCF outperforms ROC-BMCF and SACF. In addition, when compared to the best results found in the literature—i.e., Hybrid-GA—both methods obtain the same results in 25 out of 35 test problems, while SCM-BMCF has produced better results than those of Hybrid-GA in six problems and Hybrid-GA has better results than those of SCM-BMCF in four problems. SCM-BMCF performs slightly better than Hybrid-GA especially in test problems with larger sizes. Apart from the performance in solution effectiveness, the superiority of both SCM-BMCF and ROC-BMCF in solution efficiency over other approaches from the literature can be easily observed.

Acknowledgement

The authors are grateful to Professor José Fernando Gonçalves for his kind comments on some of the test instances.

Appendix A

See Tables A1–A9

Appendix B

See Tables B1–B6

Table A1
#7.

Machine:8 Part:20

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

Cell:2
Grouping Efficacy(%):58.7156
Total Operations:91
Exceptional Elements:27
Voids:18

Table A2
#10.

Machine:14 Part:24

	5	9	10	12	14	15	16	22	1	2	17	19	20	23	11	13	24	7	8	18	3	4	21	6
6	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0
14	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Cell:7
Grouping Efficacy(%):71.831
Total Operations:61
Exceptional Elements:10
Voids:10

Table A3
#12.

Machine:20 Part:20

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

Cell:5
Grouping Efficacy(%):43.2624
Total Operations:111
Exceptional Elements:50
Voids:30

Table A4
#15.

Machine:20 Part:35

	1	3	5	15	17	20	25	29	2	7	10	12	13	24	27	31	4	6	9	11	21	28	33	18	30	32	34	35	8	14	16	19	22	23	26
3	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1
8	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
17	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	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	0
16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1
5	0	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	1	0	0	0	1	1
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0
9	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	1	0	1	1	1	0	1	0	1	0
10	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	0	0	1	1	1	1	1	0	1
20	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	0	0	1	1	0	1	0	1	1

Cell:5
Grouping Efficacy:57.6087
Total Operations:148
Exceptional Elements:41
Voids:38

Table A5
#19.

Machine:24 Part:40

	2	12	15	23	34	8	11	19	21	28	37	38	25	3	32	1	16	17	10	13	14	22	35	36	4	5	18	26	27	30	6	29	39	40	7	20	31	24	9	33
3	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	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	1	1	1	1	1	0	1	1	0	0	0	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	1	0	0	0
4	0	0	0	0	0	1	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	0	0	0	0
16	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	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	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	0	0	0	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	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	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	0	1	0	0	0	0	0	0	0	0	0	0
2	0	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	0	0	0	0	0	0	0	0	0	0	1
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	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	1	0	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
9	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	0	0	0	0	1	1	1	1	0	0	0	0	0	1
17	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	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
10	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	0	1	0	0	1	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Cell:11
Grouping Efficacy(%):53.2895
Total Operations:131
Exceptional Elements:50
Voids:21

Table A6
#20.

Machine:24 Part:40

	2	12	15	23	34	11	25	28	9	20	24	13	14	27	36	7	17	31	6	29	33	39	40	3	32	4	5	18	26	30	38	8	19	21	37	1	16	10	22	35
3	1	1	1	1	1	0	0	0	0	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	0	0	0
20	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	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	1	1	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	1	1	0	0	0	1	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
14	0	0	0	0	0	1	1	0	0	0	0	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	1	0	0	0
10	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0
19	0	0	0	0	0	0	0	0	1	0	0	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	0	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	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
9	0	1	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	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	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
24	1	0	0	0	0	1	0	0	0	0	0	0	0	0	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
6	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	0	1	1	1	1	0	0	0	0	0	0	1	0	0
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0
18	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	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	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
22	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	0	0	0	1	0	0	0	0	0	1	1	0	0	0
5	0	1	0	0	0	0	0	0	0	0	0	0	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	1
11	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	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Cell:11
Grouping Efficacy(%):48.6301
Total Operations:131
Exceptional Elements:60
Voids:15

Table A7
#21.

Machine:24 Part:40

	4	18	27	7	17	20	29	31	19	36	37	39	8	11	28	15	23	24	34	3	12	21	13	14	40	5	26	30	38	1	16	2	25	32	6	9	33	10	22	35
12	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	0	0	0	0	0	0	0	0	0	0	1	0
18	1	1	1	0	0	0	0	0	0	0	0	0	0	1	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
17	0	0	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	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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
13	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
23	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	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	0	1	0	1	0
19	0	0	1	0	0	0	0	0	0	1	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	1	0	0	0	1
6	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	1	1	1	0	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
22	0	0	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	1	0	1	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
20	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
24	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	0	0	0	0	0	1	0	1	0	0	0	0	0	0
1	0	0	0	0	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	0	0	0	0	1	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0
21	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	0	0	1	0	0	0	0	0	1	1	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Cell:12
Grouping Efficacy(%):46.1538
Total Operations:131
Exceptional Elements:65
Voids:12

Table A8
#22.

Machine:30 Part:41

	8	1	3	21	22	30	10	32	33	41	5	18	26	37	2	12	23	31	39	7	14	15	25	28	35	6	13	4	16	27	34	36	38	17	9	29	11	19	20	24	40
27	1	0	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	0	0	0	0	0	0	0	0	0	0	0	0
28	1	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	0	0	0	0	0	0	0	1	0	0	0	0	0
19	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	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	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
30	0	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	0	0	0	0	0	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	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
2	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	1	1	1	0	0	0	0	0	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	1
14	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	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	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
22	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	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	1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	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	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
9	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	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	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	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	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
17	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	0	1	1	0	1	0	0	0	0	0	0	0	0	0
18	0	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	1	1	1	0	0	0	0	0	0	0	0	0	0
26	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	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
13	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	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
24	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	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	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
8	1	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	0	0	0	0	0	0	1	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Cell:14
Grouping Efficacy(%):62.5899
Total Operations:127
Exceptional Elements:40
Voids:12

Table A9

#24

Machine:30 Part:50

	28	30	50	23	24	25	13	15	34	20	38	10	14	18	40	1	4	16	33	42	45	2	3	11	5	6	7	12	26	27	29	19	31	32	44	48	17	22	9	21	43	37	46	47	8	36	41	49	35	39
7	1	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	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	1	1	1	0	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	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
19	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	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	1	0	0	0	0	0	0
12	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0
29	0	0	0	1	1	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	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
22	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	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
24	0	0	0	0	0	0	1	1	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	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
26	0	0	0	0	0	0	1	0	1	1	0	1	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	0	0	0	0	0	0	0	0	0	0	0	1	0
27	0	0	0	0	0	0	1	1	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	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	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	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	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	0	0	0
3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	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	1	0	0	1	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	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	1	0	0	0	0
5	0	0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	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	0	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
20	0	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	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1	0	0	0	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	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
6	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	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	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	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0
10	0	0	0	1	1	0	0	0	0	0	0	0	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	0	1	1	1	0	0	0	0	0	1
9	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	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0
11	0	0	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	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0
16	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	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
21	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	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	1	1
23	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1
30	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	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

Cell:14
Grouping Efficacy(%):50.8287
Total Operations:167
Exceptional Elements:75
Voids:14

Table B1

#26

Machine:24 Part:40

	25	28	40	8	11	19	21	37	39	2	12	15	23	31	34	13	14	27	36	1	16	35	6	7	29	20	24	4	5	18	26	30	38	9	17	33	3	32	10	22
7	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	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	0	1	0	0	0	0	0	0	0
16	0	0	0	1	0	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	1	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	1	1	0	0	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	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	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	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	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0
12	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	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
6	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	0	0	0	1	1	1	1	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
18	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0
23	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
11	0	0	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	0	0	0	0	0	0	0	0	1	1

Cell:11
Grouping Efficacy(%):47.3333
Total Operations:131
Exceptional Elements:60
Voids:19

Table B2

#29

Machine:28 Part:46

	4	38	14	17	19	20	21	22	23	25	28	30	34	35	46	11	32	33	39	40	41	42	43	44	45	13	26	27	29	8	9	10	36	37	1	2	3	5	6	7	12	15	16	24	18	31
1	1	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	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0
2	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	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
23	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	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1
4	0	0	1	0	1	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	1	0	0	0	0	0	0	0	0	0	1	0	1	1
5	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
18	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
25	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	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
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
22	0	0	0	0	0	0	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	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	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	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	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	1	0	0	0	0	1	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	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	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	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	1	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
9	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0
10	1	0	0	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	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0
15	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	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0
26	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
28	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Cell:10
Grouping Efficacy(%):45.3061
Total Operations:211
Exceptional Elements:100
Voids:34

Table B3

#30

Machine:30 Part:41

	10	11	32	33	41	1	3	13	21	22	30	4	18	37	2	12	20	23	31	39	40	8	9	29	7	15	24	25	35	14	16	27	34	36	5	17	19	38	6	26	28
1	1	0	1	1	1	0	0	0	0	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	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	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	0
9	0	0	0	0	0	0	1	1	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	0	0	0	0	0	0
19	0	0	0	0	0	1	1	0	1	1	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	0	0	0	0
20	0	0	0	0	0	1	1	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	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	1	1	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	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
25	0	0	0	0	0	0	0	0	0	0	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	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
22	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	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	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	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	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
27	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	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	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
17	0	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	0	0	1	1	0	1	0	0	0	0	0	0	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	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
26	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	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
5	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	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	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	0	0
13	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	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
24	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	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
6	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	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
16	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	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Cell:10
Grouping Efficacy(%):59.5238
Total Operations:127
Exceptional Elements:27
Voids:41

Table B4

#31

Machine:30 Part:50

	29	31	33	38	19	21	22	23	26	27	20	24	25	5	7	13	14	15	16	39	42	44	48	40	41	45	49	50	9	12	4	6	8	10	11	2	3	17	28	30	32	34	35	36	37	1	18	43	46	47
19	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	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
21	0	1	1	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	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
23	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	1	1	0	1	1	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	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	0	0	1	1	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	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	1	0	0	1	1	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	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0	0	0	0	0	0
30	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
25	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	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	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
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	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	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
3	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	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	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	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2	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	0	0	1	0	0	0	0	0	0	1	1	1	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
9	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	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0
20	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	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0	0	0	0	1	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	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	1	0	0	0
26	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	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
28	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Cell:12
Grouping Efficacy(%):60
Total Operations:153
Exceptional Elements:48
Voids:22

Table B5

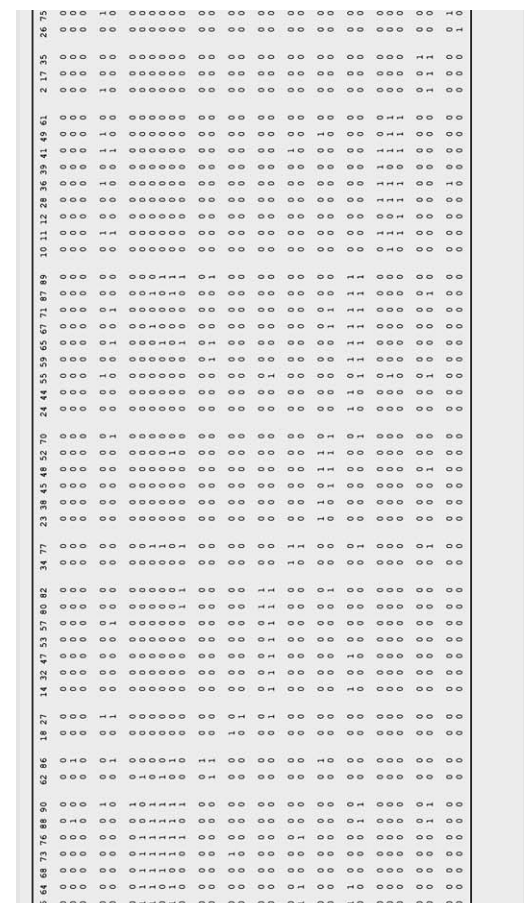
#34

Machine:37 Part:53

	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	46	49	50	1	18	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	20	21	22	23	24	25	41	42	43	44	45	47	48	51	52	53
1	1	1	1	1	1	1	1	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	1
7	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	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	0	0	0	0	0	0	1	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	1	1	1	1	1	1	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1	1	1	1	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	0	0	0	0	0	0	0	0	0	0	0	0	0	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	1	1	0	0	1	0	1	0	1	1
28	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
30	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1
32	1	1	1	1	1	1	1	1	1	1	1	1	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	0	0	0	0	0	0	0	1	0	0	0	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
25	0	0	1	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	1	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	0	0
29	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	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
36	0	0	0	0	0	0	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	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0

Cell:3
Grouping Efficacy(%):59.0406
Total Operations:977
Exceptional Elements:337
Voids:107

Table B6
#33



Machine:30 Part:90

| | 20 | 30 | 81 | 83 | 85 | 1 | 3 | 4 | 5 | 6 | 7 | 9 | 13 | 16 | 19 | 25 | 29 | 31 | 33 | 37 | 42 | 43 | 46 | 50 | 54 | 58 | 60 | 63 | 66 | 69 | 72 | 74 | 78 | 79 | 84 | 8 | 15 | 21 | 22 | 40 | 51 | 56 | 64 | 68 | 73 | 76 | 88 | 90 | 62 | 86 | 18 | 27 | 14 | 32 | 47 | 53 | 57 | 80 | 82 | 34 | 77 | 23 | 38 | 45 | 48 | 52 | 70 | 24 | 44 | 55 | 59 | 65 | 67 | 71 | 87 | 89 | 10 | 11 | 12 | 28 | 36 | 39 | 41 | 49 | 61 | 2 | 17 | 35 | 26 | 75 |
|---|---|---

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.
Seifoddini, H., Wolfe, P.M., 1986. Application of the similarity coefficient method in group technology. IIE Transactions 18 (3), 266–270.
Seifoddini, H., Wolfe, P.M., 1987. Selection of a threshold value based on material handling cost in the machine-component grouping. IIE Transactions 19, 266–270.
Srinivasan, G., Narendran, T., Mahadevan, B., 1990. An assignment model for part-families problem in group technology. International Journal of Production Research 28, 145–152.
Stanfel, L., 1985. Machine clustering for economic production. Engineering Costs and Production Economics 9, 73–81.
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, N., Salvendy, G., 1993. Modified network approach for the design of cellular manufacturing systems. International Journal of Production Research 31, 1409–1421.
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, T.-H., Chang, C.-C., Chung, S.-H., 2008. A simulated annealing algorithm for manufacturing cell formation problems. Expert Systems with Applications 34, 1609–1617.
Yasuda, K., Yin, Y., 2001. A dissimilarity measure for solving the cell formation problem in cellular manufacturing. Computers and Industrial Engineering 39, 1–17.
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.