An Ant Colony Optimization Approach for the Machine-Part Cell Formation Problem

Mehdi Hosseinabadi Farahani*
Department of Industrial Engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran

Leila Hosseini
Department of Industrial Engineering, University of Kurdistan, Pasdaran Boulevard, Sanandaj, Iran
E-mail: leila.hosseini@uok.ac.ir

Received 11 December 2010
Accepted 30 April 2011

Abstract

In this paper, the problem of grouping machines and parts into cells (machine-part cell formation problem) is considered with the objective of minimizing grouping efficacy. An ant colony optimization algorithm is developed to solve such problem. In the proposed algorithm, solutions are constructed in a new manner, two heuristic algorithms are used to determine part families and machine cells and the similarity between parts is used as heuristic information. The proposed algorithm is experimented on 35 test problems from the literature which shows its advantage over existing algorithms. The algorithm improves the best known values of the grouping efficacy for 5 problems.

Keywords: Cellular manufacturing; Machine-part cell formation; Ant colony optimization; Heuristics; Grouping efficacy.

1. Introduction

Cellular manufacturing (CM) as an application of group technology is concerned with the formation and operation of manufacturing cells in which a set of part families are processed using machine cells. One of the most important problems encountered in designing CM system is cell formation (CF), which deals with identifying machine cells and part families.[1]

Many solution methods have been developed to solve the machine-part cell formation (MPCF) problem in which a given machine-part incidence matrix is modified to obtain machine cells and part families with the objective of minimizing inter-cellular movements and maximizing machines' utilization.

Similarity and dissimilarity coefficient based methods apply a measure based on the relationships between machines or parts and use an algorithm to group machines and parts into cells. Single linkage clustering (SLINK)[2], complete linkage clustering (CLINK)[3], and average linkage clustering (ALINK)[4] are some of these algorithms. Yasuda and Yin have proposed[5] an approach based on the calculation of an Average Voids Value (AVV) that exploits a dissimilarity coefficient to group machines.

Array based methods are based on rearrangement of rows and columns of the machine-part incidence matrix to obtain a block diagonal structure. Rank order clustering algorithm, the most popular array based method, has been developed by King[6] and been improved by King and Nakornchai[7], Chan and Milner[8] and Chandrasekharan and Rajagopalan[9].

Nonhierarchical clustering algorithms presented firstly by Chandrasekharan and Rajagopalan[10] have been improved in ZODIAC[11] and GRAFICS[12] which

[*] Corresponding author. Tel./fax: +98-871-6660073; *Email: mehdi.hosseinabadi@uok.ac.ir*

Published by Atlantis Press

are relatively efficient among cluster analysis algorithms.

Graph partitioning approaches consider machines and parts as nodes and the processing of parts as arcs. In order to identify machine cells and part families, these methods aim at obtaining disconnected subgraphs from the machine-part graph. Kumar et al. have considered[13] the MPCF problem as a graph decomposition problem with fixed number of groups. Vannelli and Kumar have shown[14] that the minimal bottle-neck cell is equivalent to minimal cut-node problem and used a heuristic to find minimal bottle-neck cells and to determine machines to be duplicated. Askin et al. have solved[15] a Hamiltonian path problem to determine machine cells and part families, and Ng has used[16] the minimum spanning tree approach to solve the problem. Oliveira et al have presented[17] a bipartite graph modeling with a graph clustering algorithm for determining machine cells and part families in the CM systems.

Some studies have been devoted to solve the MPCF problem as a mathematical programming optimization problem. P-median model has been suggested by Kusiak[18] and Won[19]. Integer programming models have been proposed by Choobineh[20] and Gunasingh and Lashkari[21]. Shtub[22] and Srinivasan et al.[23] have modeled the problem as an assignment problem.

The CF problem is non-deterministic polynomial (NP) complete problem[24] and consequently traditional optimization methods are incompetent to solve the large scale problems optimally within a reasonable amount of time. Therefore, metaheuristic methods have been used widely for solving the CF problem. Cheng et al. have formulated[25] the problem as a traveling salesman problem (TSP) and developed a GA to solve it. Onwubolu and Mutingi have considered[26] minimization of cell-load variation as well as intercellular movements, and developed a GA to solve the problem. Wu et al. have presented[27] a tabu search method that uses long-term memory mechanism as well as dynamic tabu tenure. Goncalves and Resende[24] have combined GA with a local search heuristic to obtain product-machine groups. Muruganandam et al. have developed[28] a memetic algorithm with the objective of minimizing total number of moves as well as minimizing cell load variation. Stawowy has developed[29] a non-specialized and non-hybridized evolutionary strategy for the problem. James et al. have combined[30] grouping GA with the local search heuristic proposed by Goncalves and Resende[24] to determine part families and machine cells. Wu et al. have proposed[31] a simulated annealing (SA) algorithm. Tariq et al. have proposed[32] a GA combined with a local search heuristic. Wu et al. have presented[33] an approach based on SA that exploits mutation operator from GA. Mahdavi et al. have presented[34] a model for the problem with the objective of minimizing exceptional elements and number of voids in cells, and developed a GA to solve the model for real sized problems. Yang and Yang have proposed[35] a neural learning algorithm based on modified adaptive resonance theory to solve the MPCF problem. Spiliopoulos and Sofianopoulou have provided[36] an ant colony optimization approach to form part families and machine groups. Hung et al have proposed[37] a novel procedure based on a fuzzy relational data clustering algorithm for solving the manufacturing cells design problem.

In this paper, an ant colony optimization (ACO) algorithm is presented to solve the MPCF problem. In the proposed ACO algorithm, in contrast to most of previously developed metaheuristics for the MPCF problem, the solution is encoded by a permutation of all parts which is more relevant for ACO implementation. Two heuristic procedures are used for determining part families and machine cells based on the sequence of parts constructed by each ant. Heuristic information is defined effectively based on similarity between parts. Computational experiments on 35 test problems from the literature demonstrate the effectiveness and robustness of the algorithm.

The remainder of this paper is organized as follows: in section 2, problem definition and performance measure are described. In section 3, the ACO algorithm for the MPCF problem is presented. In section 4, computational experiments are conducted and reported. Section 5 concludes the paper.

2. Machine-part cell formation problem

Machine-part cell formation problem can be defined as a block diagonalization problem by using a 0-1 machine-part incidence matrix A. The element A_{ij} equals to 1 if part j (1,..., n) needs to be processed by machine i (1,..., m) and 0 otherwise. Fig. 1 shows an example of machine-part incidence matrix in which rows and columns of the matrix state for machines and parts respectively.

	Part			
Machine	1	2	3	4
1	1		1	
2	1		1	
3		1		1
4	1		1	
5	1		1	
6		1		1

Fig. 1. Initial machine-part incidence matrix

The block diagonalization problem aims at minimizing the number of 1s in the off-diagonal blocks as well as the number of 0s in the diagonal blocks by rearranging rows and columns of matrix A. In the ideal situation the solution is a perfect block diagonal matrix that the number of 1s in the off-diagonal blocks and the number of 0s in the diagonal blocks are equal to zero (See Fig. 2). This implies the minimum intercellular movements and the maximum utilization of machines.

	Part			
Machine	2	4	1	3
3	1	1		
6	1	1		
5			1	1
4			1	1
1			1	1
2			1	1

Fig. 2. Perfect block diagonal matrix

The quality of a solution to this problem is determined by a specified performance measure. One of the most prevalent measures that have been used commonly in the literature is grouping efficacy (GE).[38]

$$GE = \frac{e - e_0}{e + e_v}, \tag{1}$$

where e is the total number of 1s, e_0 is the number of exceptional elements, i.e., 1s in the off-diagonal blocks, and e_v is the number of voids, i.e., 0s in the diagonal blocks.

3. The proposed ACO algorithm

ACO as a metaheuristic method has been widely used for solving the combinatorial optimization problems (e.g. see Refs. 39 and 40). ACO is based on the foraging behavior of real ants to find the shortest path from the nest to the food. Ants are social insects which live in colonies. They prefer the benefit of their colony to their individual benefits. Since real ants have not visual ability, communicate among each other using a chemical substance called pheromone deposited on their paths. An ant that selects the shorter path will get to the food and back more quickly than one that selects the longer path. Hence, shorter paths have higher amount of pheromone and will be chosen by following ants with higher probability.

In the proposed algorithm, each artificial ant probabilistically constructs an order of parts using pheromone trails and heuristic information defined by similarity between parts. A heuristic procedure is applied to determine part families based on the order of parts constructed by each ant. Machines are then assigned to cells with respect to the parts assignment using a heuristic algorithm. The best found solution of any iteration is improved by means of local search algorithm. Furthermore, the pheromone trails are modified during the execution of the algorithm through local and global updating rules, and also limited between lower and upper bounds. The global structure of the proposed algorithm is presented as follows:

```
Step 1.  Set parameters and initialize pheromones trails.
Step 2.  While the stop condition is not met, do the
         following:
    2.1. For each artificial ant, do:
         2.1.1. Construct a solution and apply local
                updating rule to update pheromone
                trails.
         2.1.2. Assign parts to cells.
         2.1.3. Assign machines to cells based on parts
                assignment.
    2.2. Implement local search.
    2.3. Update the pheromone trails by applying
         global update rule, while updating the
         pheromone trails, also limit them.
Step 3.  Return the best solution found.
```

3.1. *Solution Construction*

A set of artificial ants is created. Each ant starts with a part selected randomly and successively appends a part that has not been already selected, to the partial solution until a feasible solution is constructed (i.e., all parts are selected). Both heuristic information and pheromone intensity are used to build solutions by the artificial ants. Choosing part j to be appended to the partial solution after part i is based on ant colony system as follows[41]:

First, q is generated according to the uniform distribution U[0,1]. If $q \leq q_0$, where q_0 is a parameter between 0 and 1 determining the relative preference of exploitation to exploration, then,

$$j = \arg\max_{v \in V} \{[\tau_t(i,v)]^{\alpha}[\eta(i,v)]^{\beta}\}, \tag{2}$$

otherwise, part j is appended to the partial solution according to the following selection probability:

$$p(i,j) = \frac{[\tau_t(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum_{v \in V}[\tau_t(i,v)]^{\alpha}[\eta(i,v)]^{\beta}}, \tag{3}$$

where V is the set of all non-appended parts, $\tau_t(i,v)$ is the pheromone trail between part i and part v at iteration t of the algorithm, α is a parameter which determines relative importance of the pheromone trail. $\eta(i,v)$ is the desirability of setting part v after part i in a partial solution based on heuristic information and β is a parameter which determines relative importance of the heuristic information. Heuristic information is calculated based on Jaccard's similarity coefficient which is defined for parts i and v as follow[42]:

$$\eta(i,v) = \frac{x}{x+y+z}, \tag{4}$$

where x is the number of machines visited by both parts, y is the number of machines visited by part i not part v and z is the number of machines visited by part v not part i.

3.2. *Local pheromone update*

Local pheromone update is performed after each solution is constructed by an ant in order to prevent from premature convergence by reducing the amount of pheromone on the corresponding path and to discourage following ants from selecting the same path. In the proposed algorithm the following local pheromone update rule is applied.

$$\tau_t(i,j) = (1-\rho').\tau_t(i,j), \tag{5}$$

where ρ' is a parameter between 0 and 1.

The applied local pheromone update rule leads the searching procedure to unvisited areas by increasing diversification more than standard ant colony system. Before executing the global pheromone update, the changes resulted from local pheromone update are removed.

3.3. *Determining part-families*

After the sequence of parts is specified by an ant, a heuristic method is applied to determine part families. The global procedure of determining part families is presented below.

Step 1. Calculate the similarity coefficient between each part and its immediately following part in the solution constructed by an ant.

Step 2. Consider (C-1) lowest calculated numbers in Step1 and specify the position of associated parts (C is the number of cells).

Step 3. Determine cells based on the specified position numbers in step2. Starting from the part placed at the first position, assign parts one by one to cell 1, until the lowest specified position number in Step2. By starting from the next part, assign parts one by one to cell 2, until the second lowest specified position number in Step2 and so on.

3.4. *Determining machine-cells*

After determining part- families, we need to determine machine-cells to have a complete solution. Each machine i is assigned to a cell c in which the most grouping efficacy estimation is obtained according to the following formula:

$$GEE_{i,c} = \frac{ones1_i - onesout_{i,c}}{ones2_i + zerosin_{i,c}}, \tag{6}$$

where

$$ones1_i = ones - \sum_{k=1}^{i-1} onesout_k, \tag{7}$$

$$ones2_i = ones + \sum_{k=1}^{i-1} zerosin_k, \tag{8}$$

and $onesout_{i,c}$ is the number of 1s related to machine i in the off-diagonal blocks if it is assigned to cell c, $zerosin_{i,c}$ is the number of 0s related to machine i in the diagonal blocks if it is assigned to cell c, $ones$ is the total number of 1s in the machine-part incidence matrix, $onesout_k$ is the number of 1s related to previously assigned machine k in the off-diagonal blocks with regard to its cell and *zerosin*$_k$ is the number of 0s related to machine *k* in the diagonal blocks.

However, the machine assignment determined by the above procedure may be infeasible. In other words, it is possible that at least one cell has no machine assigned to it. In the case of infeasibility, a heuristic method is used. For each cell containing no machine, by starting from the smallest index, among machines that there is at least another machine in their corresponding cell, a machine is chosen that its assigning to the cell maximizes the grouping efficacy calculated from Eq.(1). The selected machine is then assigned to the cell.

3.5. *Local Search*

In the proposed algorithm, local search is applied to the best found solution of any iteration. The employed local search is given by Tariq et al.[32] This local search works by checking that if any changes in machine assignment or part assignment improves the GE. In particular, starting from machine 1, each machine is assigned to different cells and when an improvement occurs, the change is recorded. The same procedure is applied to parts. The cycle of local improvement is started all over again until no further improvement is recorded.

3.6. *Global pheromone update*

Global updating is performed to make the search more directed. At the end of each iteration *t*, the following global updating rule is applied to the pheromone trails:

$$\tau_{t+1}(i,j) = (1-\rho)\tau_t(i,j) + H(\rho GE^*), \quad (9)$$

where ρ ($0 \leq \rho \leq 1$) is the evaporation rate and *H* is a non-negative parameter employed to manage the change of the pheromone intensities. The parameter *H* has been set to 1 for the edges between each two parts assigned to same cell in the global best solution, i.e., the best solution found so far, and to 0 for the others; accordingly, the pheromone trails between each two parts assigned to different cells in the global best solution are only evaporated.

In the beginning of the first iteration, initial pheromone values of all paths are set 1. The pheromone values of all paths are always limited between two boundaries $\tau_{\min}$ and $\tau_{\max}$ which are calculated based on the following formula:

$$\tau_{\max} = \rho GE^*, \quad (10)$$

$$\tau_{\min} = 0.0001\tau_{\max}. \quad (11)$$

It is obvious that the above boundaries are updated only when a solution better than the global best solution is obtained.

4. Computational experiments

The proposed ACO algorithm has been coded in Visual C++ and implemented on a 2.00 GHz PC with 2 GB memory. To determine the best values of parameters, a series of pilot experiments were conducted. The following results have then been achieved: 50 artificial ants in the colony, $\rho = 0.01$, $\rho = 0.01$, $\alpha = 5$, $\beta = 0.9$ and $q_0 = 0.7$. The algorithm terminates after 1000 iterations or 30 consecutive iterations with no improvement, depending on which criterion is satisfied first.

To evaluate the performance of the algorithm, a set of 35 problems from the literature where the size of matrices ranges from 5×7 through 40×100 have been tested. The sources of data sets are demonstrated in Table 1.

The computational results of the proposed ACO algorithm are shown in Table 2, which gives the minimum, the average, the maximum, the standard deviation of the obtained grouping efficacies and the average computation time found over 10 runs for each instance.

Considering the results shown in Table 2, it is seen that the new approach, for 5 instances, finds solutions better than the best known solutions and, for 23 test problems, solutions equal to the best known ones. The average computation times are extremely low and never exceed 8.60 seconds for all test problems. Also, the standard deviation is 0 for 16 problems and at most 0.71, which indicates the high robustness of the algorithm. The trade-off between intensification and diversification is the key point to achieve good results in different runs of a search algorithm.[43] In the proposed algorithm, this trade-off is obtained by using a well-defined heuristic information and an effective local search algorithm to reinforce the intensification of the algorithm as well as defining upper and lower limits for the pheromone values and using a strategy in the local pheromone update to increase the diversification of the algorithm.

Table 1. Data sources

No.	Problem Source	Size
1	King and Nakornchai[7]	5×7
2	Waghodekar and Sahu[46]	5×7
3	Seifoddini[47]	5×18
4	Kusiak and Cho[48]	6×8
5	Kusiak and Chow[49]	7×11
6	Boctor [50]	7×11
7	Seifoddini and Wolfe [51]	8×12
8	Chandrasekharan and Rajagopalan[9]	8×20
9	Chandrasekharan and Rajagopalan[10]	8×20
10	Mosier and Taube[52]	10×10
11	Chan and Milner[8]	10×15
12	Askin and Subramanian[53]	14×24
13	Stanfel[54]	14×24
14	McCormick et al.[55]	16×24
15	Srinivasan et al.[23]	16×30
16	King[6]	16×43
17	Carrie[56]	18×24
18	Mosier and Taube[57]	20×20
19	Kumar et al.[13]	20×23
20	Carrie[56]	20×35
21	Boe and Cheng[58]	20×35
22	Chandrasekharan and Rajagopalan[59]	24×40
23	Chandrasekharan and Rajagopalan[59]	24×40
24	Chandrasekharan and Rajagopalan[59]	24×40
25	Chandrasekharan and Rajagopalan[59]	24×40
26	Chandrasekharan and Rajagopalan[59]	24×40
27	Chandrasekharan and Rajagopalan[59]	24×40
28	McCormick et al.[55]	27×27
29	Carrie[56]	28×46
30	Kumar and Vannelli[60]	30×41
31	Stanfel[54]	30×50
32	Stanfel[54]	30×50
33	King and Nakornchai[7]	36×90
34	McCormick et al.[55]	37×53
35	Chandrasekharan and Rajagopalan[11]	40×100

The results are compared to some well-known methods reported in the literature, including ZODIAC[11], GRAFICS[12],MST[44], GATSP[25], GP[45], GA[26], EA[24], ES[29], HGGA[30], HGA[32], GA*[34], SA[31] and HHA[33].

ZODIAC, GRAFICS, MST, and EA do not allow Singletons, i.e., cells containing only one machine or only one part, which can degrade solution quality. HHA has reported solutions in the both cases.

The best grouping efficacies obtained by the algorithms that allow singletons reported in the literature are shown in Table 2. As seen, the proposed algorithm outperforms GATSP, ES and HGA for all problem instances.

In GP, 17 problems have been considered; for 14 problems solutions found by the new approach are better and for 2 problems are equal. Since GP has reported only three significant digits, possibly the rounding error causes the slight better solution for problem 25. In GA, 25 problems have been considered; the new approach obtains better solutions for 20 problems and equal solutions for 4 problems. For problem 16, GA has reported a solution which is inconsistent with that of the other approaches.

In comparison with HGGA, the proposed approach finds better solutions for 7 problems and the same solutions for 21 test problems. From the detailed results reported by HGGA, it seems that, for problems 1 and 16, the used data is inconsistent with those used in this study, and for problems 27 and 30, the grouping efficacy values is not consistent with the reported solutions. In GA*, 22 problems have been considered; for 4 problems solutions found by the new approach are better and for 15 problems are equal.

In SA and HHA, 25 problems have been considered. In comparison with SA, better solutions have been achieved by the new approach for 10 problems while for 14 problems, solutions are equal. In comparison with HHA, the new approach achieves better solutions for 5 problems and the same solutions for 18 test problems. For problem 31, SA and HHA outperform our approach, but, from the data reported by HHA, the used data is clearly different.

In order to have a fair comparison, the proposed algorithm has also been run 10 replicates on each 35 instances by considering the constraint of having no singleton. The results are shown in Table 3.

The best grouping efficacies obtained by the algorithms that do not allow singletons, reported in the literature are also shown in Table 3. It can be observed from Table 3 that the proposed algorithm finds the best overall solutions for 6 instances and solutions equal to the best known ones for 20 test problems.

Table 2. Results in the case where singletons are allowed.

											New Approach					
No.	**Size (m×n)**	**GATSP**	**GP**	**GA**	**ES**	**HGGA**	**SA**	**HGA**	**GA***	**HHA**	**No. of Cells**	**Min.**	**Avg.**	**Max.**	**St. dev**	**Avg. Time (s)**
1	5×7				73.68	82.35		73.68			2	73.68	73.68	73.68	0.00	0.00
2	5×7	68		62.5	60.87	69.57	69.57	69.57	69.57	69.57	2	69.57	69.57	69.57	0.00	0.00
3	5×18	77.36		77.36	79.59	79.59	79.59	79.59	79.59	79.59	2	79.59	79.59	79.59	0.00	0.02
4	6×8	76.92		76.92		76.92	76.92	76.92	76.92	76.92	2	76.92	76.92	76.92	0.00	0.00
5	7×11	46.88		50	53.13	60.87	60.87	58.62	60.87	60.87	5	60.87	60.87	60.87	0.00	0.04
6	7×11	70.37		70.37	70.37	70.83	70.83	70.37	70.83	70.83	4	70.83	70.83	70.83	0.00	0.02
7	8×12				68.29	69.44		68.3			4	69.44	69.44	69.44	0.00	0.03
8	8×20	85.24	85.2	85.24	85.25	85.25	85.25	85.25	85.25	85.25	3	85.25	85.25	85.25	0.00	0.04
9	8×20	58.33	58.7	55.91	58.72	58.72	58.41	58.72	58.72	58.72	2	58.72	58.72	58.72	0.00	0.11
10	10×10	70.59		72.79	70.59	75	75	70.59	75	75	5	75.00	75.00	75.00	0.00	0.01
11	10×15	92	92	92	92	92	92	92	92	92	3	92.00	92.00	92.00	0.00	0.01
12	14×24				69.86	72.06		70.83			7	72.06	72.06	72.06	0.00	0.03
13	14×24	67.44	71.8	63.48	69.33	71.83	71.21	70.51	71.83	71.83	7	71.43	71.51	71.83	0.17	0.35
14	16×24				51.96	52.75		51.96			8	52.22	52.89	**53.26**	0.36	0.44
15	16×30				67.83	68.99		67.83			6	67.44	68.55	68.99	0.71	0.39
16	16×43	53.89		86.25	54.86	57.53	52.44	54.86	56.13	56.38	8	55.13	55.59	56.55	0.50	1.13
17	18×24				54.46	57.73		54.95			9	56.07	57.15	57.73	0.59	0.33
18	20×20	37.12		34.16	42.96	43.18	41.04	43.45	42.94	43.26	5	42.66	43.12	43.45	0.23	0.38
19	20×23	46.62	49	39.02	49.65	50.81	50.81	49.65		50.81	7	50.00	50.52	50.81	0.27	0.64
20	20×35	75.28	76.7	66.3	76.14	77.91	78.4	76.14	77.9	78.4	5	78.40	78.40	78.40	0.00	0.31
21	20×35	55.14	56.8	44.44	58.06	57.98	56.04	58.38		57.61	5	57.53	58.27	58.38	0.27	0.53
22	24×40	100	100	100	100	100	100	100	100	100	7	100.00	100.00	100.00	0.00	0.10
23	24×40	85.11	85.1	85.11	85.11	85.11	85.11	85.11	85.11	85.11	7	85.11	85.11	85.11	0.00	0.11
24	24×40	73.03	73.5	73.03	73.51	73.51	73.51	73.51	73.51	73.51	7	73.51	73.51	73.51	0.00	0.12
25	24×40	49.37	53.3	37.62	51.88	53.29	52.44	52.5	52.87	53.29	11	52.38	52.87	53.29	0.28	3.11
26	24×40	44.67	47.9	34.76	46.95	48.95	47.13	46.84	48.95	48.63	12	48.00	48.41	48.61	0.21	4.47
27	24×40	42.5	43.7	34.06	44.85	47.26	44.64	44.85	47.26	46.15	12	45.21	45.73	46.26	0.37	3.55
28	27×27				54.27	54.02		54.31			6	54.52	54.66	**54.77**	0.12	0.32
29	28×46				44.78	46.91		46.43			10	45.88	46.54	**47.06**	0.44	4.48
30	30×41	53.8	60.7	40.96	58.72	63.31	62.42	60.74		62.59	15	61.15	61.72	**62.77**	0.49	6.68
31	30×50	56.61	59.4	48.28	59.66	59.77	60.12	59.66	60.12	60.12	13	57.80	58.70	59.66	0.63	4.36
32	30×50	45.93	50	37.55	50.51	50.83	50.51	50.51	50.83	50.83	14	50.00	50.17	50.83	0.27	5.11
33	36×90				43.35	46.35		44.67			17	44.27	44.82	45.30	0.31	8.60
34	37×53				56.42	60.64		59.6			3	60.08	60.33	**61.00**	0.30	3.56
35	40×100	84.03	84	83.9	84.03	84.03	84.03	84.03	84.03	84.03	10	84.03	84.03	84.03	0.00	1.09

Table 3. Results in the case where singletons are not allowed.

							New Approach					
No.	**Size (m×n)**	**ZODIAC**	**GRAFICS**	**MST**	**EA**	**HHA**	**No. of Cells**	**Min.**	**Avg.**	**Max.**	**St. dev**	**Avg. Time (s)**
1	5×7	73.68	73.68		73.68	73.68	2	73.68	73.68	73.68	0.00	0.01
2	5×7	56.52	60.87		62.5	62.50	2	62.50	62.50	62.50	0.00	0.01
3	5×18				79.59	79.59	2	79.59	79.59	79.59	0.00	0.02
4	6×8				76.92	76.92	2	76.92	76.92	76.92	0.00	0.00
5	7×11	39.13	53.12		53.13	53.13	3	53.13	53.13	53.13	0.00	0.01
6	7×11				70.37	70.37	3	70.37	70.37	70.37	0.00	0.01
7	8×12	68.3	68.3		68.3	68.29	3	68.29	68.29	68.29	0.00	0.01
8	8×20	85.24	85.24	85.24	85.25	85.25	3	85.25	85.25	85.25	0.00	0.02
9	8×20	58.33	58.13	58.72	58.72	58.72	2	58.33	58.55	58.72	0.18	0.07
10	10×10	70.59	70.59	70.59	70.59	70.59	3	70.59	70.59	70.59	0.00	0.01
11	10×15	92	92		92	92	3	92.00	92.00	92.00	0.00	0.02
12	14×24	64.36	64.36	64.36	69.86	69.86	5	66.25	66.25	66.25	0.00	0.05
13	14×24	65.55	65.55		69.33	69.33	5	69.33	69.33	69.33	0.00	0.03
14	16×24	32.09	45.52	48.7	52.58	51.96	6	50.56	51.59	52.27	0.75	1.10
15	16×30	67.83	67.83	67.83	67.83	67.83	4	67.83	67.83	67.83	0.00	0.07
16	16×43	53.76	54.39	54.44	54.86	54.60	6	53.57	54.27	**55.21**	0.53	0.82
17	18×24	41.84	48.91	44.2	54.46	54.46	6	53.15	54.12	54.46	0.50	0.49
18	20×20	21.63	38.26		42.96	42.96	5	41.91	42.51	42.96	0.38	0.45
19	20×23	38.66	49.36	43.01	49.65	49.65	5	49.30	49.55	49.65	0.17	0.21
20	20×35	75.14	75.14	75.14	76.22	76.22	5	75.61	75.80	**76.54**	0.39	0.44
21	20×35				58.07	57.61	5	56.48	57.54	**58.15**	0.54	1.09
22	24×40	100	100	100	100	100	7	100.00	100.00	100.00	0.00	0.10
23	24×40	85.1	85.1	85.11	85.11	85.11	7	85.11	85.11	85.11	0.00	0.11
24	24×40	37.85	73.51	73.51	73.51	73.51	7	73.51	73.51	73.51	0.00	0.11
25	24×40	20.42	43.27	51.81	51.97	51.97	9	49.04	50.09	51.87	0.87	2.82
26	24×40	18.23	44.51	44.72	47.06	47.33	11	44.74	45.77	46.67	0.50	4.23
27	24×40	17.61	41.67	44.17	44.87	44.87	10	41.25	42.67	43.75	0.78	4.06
28	27×27	52.14	47.37	51	54.27	54.27	4	53.99	54.20	**54.31**	0.11	0.63
29	28×46	33.01	32.86	40	44.62	45.31	10	44.40	44.83	**45.56**	0.38	4.35
30	30×41	33.46	55.43	55.29	58.48	59.52	10	57.14	57.79	58.19	0.40	3.58
31	30×50	46.06	56.32	58.7	59.66	60	12	55.08	56.12	58.43	1.14	4.43
32	30×50	21.11	47.96	46.3	50.51	50.51	11	48.52	49.70	50.51	0.58	6.99
33	36×90	32.73	39.41	40.05	42.12	44.59	12	39.39	41.05	42.34	0.87	10.76
34	37×53	52.21	52.21		56.42	59.04	3	59.16	59.60	**60.56**	0.38	4.96
35	40×100	83.92	83.92	83.66	84.03	84.03	10	84.03	84.03	84.03	0.00	3.19

5. Conclusion

In this paper, an ant colony optimization algorithm is proposed for the machine-part cell formation problem. In the developed algorithm, solutions are constructed by a new procedure and Jaccard similarity coefficient is used as heuristic information. The algorithm not only improves the best known values of the grouping efficacy for 5 out of 35 problems from the literature but also shows an excellent performance in computation time. According to the very low diversity among solutions found in different runs, the robustness of the algorithm is remarkable.

In the proposed algorithm, in order to have a fair comparison with other methods, the number of cells for each test problem has been considered to be a predetermined number. Developing methods for determining machine cells and part families where the number of cells is not predetermined may cause improvement in the grouping efficacy. Furthermore, applying the proposed ACO algorithm with some modifications to other grouping and clustering problems is an attractive aspect for future research.

References

1. H. M. Selim, R. G. Askin and A. J. Vakharia, Cell formation in group technology: review, evaluation and directions for future research, *Comput. Ind. Eng.* **34**(1) (1998) 3–20.
2. J. McAuley, Machine grouping for efficient production. *Prod. Eng.* **51**(2) (1972) 53–57.
3. H. C. Romesburg, *Cluster analysis for Researchers* (Lifetime Learning Publications, Belmont, CA, 1984).
4. H. K. Seifoddini, Incorporation of the production volume in machine cells formation in group technology applications, in *Proc. 9th Int. Conf.* Production Research, (Cincinnati, Ohio, 1987), pp. 2348–2356.
5. K. Yasuda and Y. Yin, A dissimilarity measure for solving the cell formation problem in cellular manufacturing, *Comput. Ind. Eng.* **47** (2004) 247–273.
6. J. R. King, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, *Int. J. Prod. Res.* **18** (1980) 213-32.
7. J. R. King and V. Nakornchai, Machine-component group formation in group technology: Review and extension, *Int. J. Prod. Res.* **20**(2) (1982) 117-133.
8. H. M. Chan and D. A. Milner, Direct clustering algorithm for group formation in cellular manufacture, *J. Manuf. Syst.* **1** (1982) 65-75.
9. M. P. Chandrasekharan and R. Rajagopalan, MODROC: an extension of rank order clustering for group technology, *Int. J. Prod. Res.* **24**(5) (1986) 1221-1264.
10. M. P. Chandrasekharan and R. Rajagopalan, An ideal seed non- hierarchical clustering algorithm for cellular manufacturing, *Int. J. Prod. Res.* **24** (2) (1986) 451–464.
11. M. P. Chandrasekharan and R. Rajagopalan, ZODIAC: an algorithm for concurrent formation of part-families and machine-cells, *Int. J. Prod. Res.* **25**(6) (1987) 835–850.
12. G. Srinivasan and T. Narendran, GRAFICS: a non-hierarchical clustering algorithm for group technology, *Int. J. Prod. Res.* **29** (1991) 463–478.
13. K. R. Kumar, A. Kusiak, and A. Vannelli, Grouping of parts and components in flexible manufacturing systems, *Eur. J. Oper. Res.* **24** (1986) 387–397.
14. A. Vannelli and K. R. Kumar, A method for finding minimal bottle-neck cells for grouping part-machine families, *J. Intell. Manuf.* **24**(2) (1986) 387-400.
15. R. G. Askin, S. H. Cresswell, J. B. Goldberg and A. J. Vakharia, A hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing, *Int. J. Prod. Res.* **29**(6) (1991) 1081-1100.
16. S. Ng, On the characterization and measure of machine cells in group technology, *Oper. Res.* **44**(5) (1996) 735–744.
17. S. Oliveira, J. F. F. Ribeiro and S. C. Seok, A spectral clustering algorithm for manufacturing cell formation, *Comput. Ind. Eng.* **57**(3) (2009) 1008–1014.
18. A. Kusiak, The generalized group technology concept, *Int. J. Prod. Res.* **25**(4) (1987) 561–569.
19. Y. Won, Two-phase approach to GT cell formation using efficient p-median formulations, *Int. J. Prod. Res.* **38**(7) (2000) 1601–1613.
20. F. Choobineh, A framework for the design of cellular manufacturing systems, *Int. J. Prod. Res.* **26**(7) (1988) 1161–1172.
21. K. R. Gunasingh and R. S. Lashkari, Machine grouping problem in cellular manufacturing systems: an integer programming approach, *Int. J. Prod. Res.* **27**(9) (1989) 1465–1473.
22. A. Shtub, Modeling group technology cell formation as a generalized assignment problem, *Int. J. Prod. Res.* **27**(5) (1989) 775–782.
23. G. Srinivasan, T. Narendran and B. Mahadevan, An assignment model for the part-families problem in group technology, *Int. J. Prod. Res.* **28** (1990) 145–152.
24. J. Goncalves, and M. Resende, An evolutionary algorithm for manufacturing cell formation, *Comput. Ind. Eng.* **47** (2004) 247–273.
25. C. H. Cheng, Y. P. Gupta, W. H. Lee and K. F. Wong, A TSP-based heuristic for forming machine groups and part families, *Int. J. Prod. Res.* **36**(5) (1998) 1325–1337.

26. G. C. Onwubolu and M. Mutingi, A genetic algorithm approach to cellular manufacturing systems, *Comput. Ind. Eng.* **39**(1-2) (2001) 125–144.
27. T. H. Wu, C. Low and W. T. Wu, A tabu search approach to the cell formation problem, *Int. J. Adv. Manuf. Tech.* **23** (2004) 916–924.
28. A. Muruganandam, G. Prabhaharan, P. Asokan and V. Baskaran, A memetic algorithm approach to the cell formation problem, *Int. J. Adv. Manuf. Tech.* **25** (2005) 988–997.
29. A. Stawowy, Evolutionary strategy for manufacturing cell design, *OMEGA* **34** (2006) 1-18.
30. T. L. James, E. C. Brown and K. B. Keeling, A hybrid grouping genetic algorithm for the cell formation problem, *Comput. Oper. Res.* **34** (2007) 2059–2079.
31. T. H. Wu, C. C. Chang, and S. H. Chung, A simulated annealing algorithm for manufacturing cell formation problems, *Expert. Syst. Appl.* **34** (2008) 1609–1617.
32. A. Tariq, I. Hussain, and A. Ghafoor, A hybrid genetic algorithm for machine-part grouping, *Comput, Ind. Eng.* **56** (2009) 347–356.
33. T. H. Wu, C. C. Chang and J. Y. Yeh, A hybrid heuristic algorithm adopting both Boltzmann function and mutation operator for manufacturing cell formation problems*, Int. J. Prod. Econ.* **120** (2009) 669–688.
34. I. Mahdavi, M. M. Paydar, M. Solimanpur and A. Heidarzade, Genetic algorithm approach for solving a cell formation problem in cellular manufacturing, *Expert. Syst. Appl.* **36** (2009) 6598–6604.
35. M. S. Yang and J. H. Yang, Machine-part cell formation in group technology using a modified ART1 method, *Eur. J. Oper. Res.* **188**(1) (2008) 140–152.
36. K. Spiliopoulos and S. Sofianopoulou, An efficient ant colony optimization system for the manufacturing cells formation problem, *Int. J. Adv. Manuf. Tech.* **36**(5-6) (2008) 589–597.
37. W. L. Hung, M. S. Yang and E. S. Lee, cell formation using fuzzy relational clustering algorithm, *Math. Comput. Model.* **53**(9-10) (2011) 1776–1787.
38. C. S. Kumar and M. P. Chandrasekharan, Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, *Int. J. Prod. Res.* **28** (1990) 233-243.
39. B. Bullnheimer, R. F. Hartl and C. Strauss, An improved ant system algorithm for the vehicle routing problem, *Ann. Oper. Res.* **89** (1999) 319–328.
40. F. Ahmadizar and F. Barzinpour, A hybrid algorithm to minimize makespan for the permutation flow shop scheduling problem, *Int. J. Comput. Int. Sys.* **3** (2010) 853-861.
41. M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE T. Evolut. Comput.* **l** (1997) 53-66.
42. P. Jaccard, Novelles recgerches sur la distribution florale, Bulletin de la Socie`te` Vaudoise des Sciences Naturelles 44 (1908) 223–270.
43. C. Blum, ACO Applied to Group Shop Scheduling: A Case Study on Intensification and Diversification, in Lecture Notes in Computer Science (Springer-Verlag, Berlin, 2002), pp. 149–180.
44. G. Srinivasan, A clustering algorithm for machine cell formation in group technology using minimum spanning trees, *Int. J. Prod. Res.* **32** (1994) 2149-2158.
45. C. Dimopoulos and N. Mort, A hierarchical clustering methodology based on genetic programming for the solution of simple cell formation, *Int. J. Prod. Res.* **39**(1) (2001) 1-19.
46. P. H. Waghodekar and S. Sahu, Machine-component cell formation in group technology: MACE, *Int. J. Prod. Res.* **22** (1984) 937-948.
47. H. Seifoddini, A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications, *Int. J. Prod. Res.* **27**(7) (1989) 1161-1165.
48. A. Kusiak and M. Cho, Similarity coefficient algorithm for solving the group technology problem, *Int. J. Prod. Res.* **30**(11) (1992) 2633-2646.
49. A. Kusiak and W. S. Chow, Efficient solving of the group technology problem, *J. Manuf. Syst.* **6**(2) (1987) 117-124.
50. F. F. Boctor, A linear formulation of the machine-part cell formation problem, *Int. J. Prod. Res.* **29**(2) (1991) 343-356.
51. H. Seifoddini and P. M. Wolfe, Application of the similarity coefficient method in group technology, *IIE Trans.* **18**(3) (1986) 271-277.
52. C. T. Mosier and L. Taube, The facets of group technology and their impact on implementation, *OMEGA* **13**(5) (1985) 381-391.
53. R. G. Askin and S. P. Subramanian, A cost-based heuristic for group technology configuration, *Int. J. of Prod. Res.* **25**(1) (1987) 101-113.
54. L. E. Stanfel, Machine clustering for economic production, *Eng. Cost. Prod. Econ.* **9** (1985) 73-81.
55. W. T. McCormick, P. J. Schweitzer and T. W. White, Problem decomposition and data reorganization by a clustering technique*, Oper. Res.***20** (1972) 993-1009.
56. A. S. Carrie, Numerical taxonomy applied to group technology and plant layout. *Int. J. Prod. Res.* **11**(4) (1973) 399-416.
57. C. T. Mosier and L. Taube, Weighted similarity measure heuristics for the group technology machine clustering problem, *OMEGA* **13**(6) (1985) 577-583.
58. W. J. Boe and C. H. Cheng, A close neighbour algorithm for designing cellular manufacturing systems, *Int. J. Prod. Res.* **29**(10) (1991) 2097-2116.
59. M. P. Chandrasekharan and R. Rajagopalan, GROUPABILITY: an analysis of the properties of

binary data matrices for group technology. *Int. J. Prod. Res.* **27**(6) (1989) 1035-1052.

60. K. R. Kumar and A. Vannelli, Strategic subcontracting for efficient disaggregated manufacturing. *Int. J. Prod. Res.* **25**(12) (1987) 1715-1728.