

# Heuristic Approaches for Cell Formation in Cellular Manufacturing

Shahram Saeedi<sup>1</sup>, Maghsud Solimanpur<sup>2</sup>, Iraj Mahdavi<sup>1</sup>, Nikbakhsh Javadian<sup>1</sup>

<sup>1</sup>Department of Industrial Eng., Mazandaran University of Science and Technology, Babol, Iran; <sup>2</sup>Faculty of Engineering, Urmia University, Urmia, Iran.

Email: shahram.saeedi@gmail.com, irajarash@rediffmail.com, nijavadian@ustmb.ac.ir, m.solimanpur@urmia.ac.ir

Received April 22<sup>nd</sup>, 2010; revised May 16<sup>th</sup>, 2010; accepted May 22<sup>nd</sup>, 2010.

## ABSTRACT

*Cellular Manufacturing System (CMS) is an application of Group Technology (GT) that allows decomposing a manufacturing system into subsystems. Grouping the machines and parts in a cellular manufacturing system, based on similarities is known as cell formation problem (CFP) which is an NP-hard problem. In this paper, a mathematical model is proposed for CFP and is solved using the Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Simulated Annealing (SA) meta-heuristic methods and the results are compared. The computational results show that the GA method is more effective in solving the model.*

**Keywords:** Cell Formation Problem, Ant Colony Optimization, Genetic Algorithm, Simulated Annealing, Sequence Data, Production Volume

## 1. Introduction

Cellular Manufacturing System (CMS) is an application of the Group Technology (GT) philosophy that allows decomposing a manufacturing system into subsystems which makes its management easier than the entire manufacturing system. It has been shown that CMS is an accepted solution to the problem of productivity in batch production which includes a large portion of world manufacturing [1]. The main idea in CMS is the principle of “Similar things should be done similarly” which means the similar manufacturing processes should be identified and grouped in dedicated manufacturing cells.

Manufacturing systems employing CMS can improve the productivity to a large extent. It has been found that CMS can increase the productivity of manufacturing system by three major factors [2]:

- improvement in quality of the work-force,
- increase in the availability of capital,
- improvement in the production technology.

Based on the simulation results performed by Morris and Tersine [3], the superiority of CM over batch production is significant especially when the setup/operation ratio is high, demand is stable, one-way intercellular flows and considerable materials handling are concerned.

Due to different solution approaches, different grouping solutions may be proposed for a certain problem.

Therefore there should be some criteria to compare these solutions and choose the best one. There are several objectives to measure the effectiveness of CMS such as:

- Minimum number of intercellular/intracellular moves,
- Greatest proportion of part operations performed within a single cell,
- Maximum machine utilization,
- Minimal total costs by reducing set-up times, and WIP (Work-in-Process),
- Minimal capital investment,
- Minimum number of voids in the cells.

With respect to the benefits mentioned above, CMS has attracted the attention of researchers for the last decades. Some researches related to the work presented in this paper are reviewed in the following.

Burbidge [4] defines group technology as: “an approach to the organization of work in which the organizational units are relatively independent groups, each responsible for the production of a given family of products”. In this approach, the main goal is to form manufacturing groups in which, some machines are located in dedicated cells associated with some similar parts based on a machine-part incidence matrix. In each cell, some operations are done on the parts by machines, so that the main objective is to maximize the intra-cell operations, and to minimize the number of inter-cell movements

(exceptional elements). It is shown that the machine-part cell formation (MPCF) is a NP-hard problem [5]. Therefore, it takes a long time to obtain an optimal solution for medium-sized problems while it is computationally intractable for large-sized problems. Thus, development and application of heuristic techniques has attained the interest of researchers in this area.

Joines *et al.* [6] offered a classification of the techniques available for manufacturing cell formation. Individual techniques are aggregated into methodological groups including array-based clustering, hierarchical clustering, non-hierarchical clustering, graph theoretic approach, artificial intelligence, mathematical programming, and heuristic approaches.

**Table 1** provides a review of the researches related to the current work in terms of the solution approach or problem perspective. The works pointed out in this table suffer from at least one of the following drawbacks:

1) Intercellular movements have been calculated regardless of production volume though it is directly affected by this parameter.

2) Sequence of operations has only been taken into account in the calculation of similarity between the parts. However, this parameter directly affects the number of movements of parts between the cells.

3) In a large number of researches, the total number of “ones” fell out of diagonal blocks is considered as a measure of the number of intercellular movements between the cells. However, this value is seriously dictated by the sequence through which parts are processed. Suppose a certain operation of a part is processed out of the associated cell. If this is the first or the last operation of the part, a single intercellular movement takes place whereas it is counted twice in otherwise. The mathematical model attempted in this paper provides a formula to calculate the intercellular movements in this way.

In this paper, a mathematical model is proposed for solving the cell formation problem, and the model is solved using Genetic Algorithm (GA), Simulated Annealing (SA) and Ant Colony Optimization (ACO). Performance of these methods is compared using two examples selected from the literature. The comparison shows

**Table 1. Summary of literature review**

Reference	Applied Methodology	Sequence of operation	Production Volume	Exceptional Elements (Voids)	Intercellular Movements
Islir [7]	Ant algorithm	No	No	No	No
Prabhakaran <i>et al.</i> [8]	Ant algorithm	Yes	Yes	No	Yes
Mak <i>et al.</i> [9]	Ant algorithm	Yes	No	No	No
Spiliopoulos and Sofianopoulou [10]	Ant algorithm	Yes	No	No	Yes
Kesen <i>et al.</i> [11]	Ant algorithm	Yes	No	No	No
Satolgu and Suresh [12]	Goal Programming	No	No	No	No
Kao and Fu [13]	Clustering Algorithm	No	No	No	No
Pandian and Mahapatra [14]	Neural Networks	Yes	No	Yes	Yes
Mahdavi <i>et al.</i> [15]	Genetic Algorithm	Yes	No	Yes	No
Mahdavi and Shirazi [16]	Heuristic Algorithm	Yes	No	Yes	No
Arkat <i>et al.</i> [17]	Simulated Annealing	No	Yes	No	No
Ahi <i>et al.</i> [18]	TOPSIS	Yes	No	Yes	No
Wang <i>et al.</i> [19]	Scatter Search	Yes	Yes	No	No
Murugunandam <i>et al.</i> [20]	GA + Tabu Search	Yes	Yes	No	No

the effectiveness of GA method.

## 2. Problem Formulations

### 2.1 Notations

$C, M, P$ : Total number of Cells, Machines, and Parts.

$i, j, c$ : Index of machines, parts and cells respectively.

$D_j$ : Demand for part  $j$ .

$L_c$ : Minimum number of machines which should be assigned to cell  $c$ .

$y_{jc}$ : Boolean decision variable, which is 1 if part  $j$  is assigned cell  $c$ , and 0 otherwise.

$x_{ic}$ : Boolean decision variable, which is 1 if machine  $i$  is assigned to cell  $c$ , and 0 otherwise.

$\alpha_{ij}$ : Boolean parameter, which is 1 if part  $j$  needs machine  $i$  for completion, and 0 otherwise.

$\beta_{ij}$ : Boolean parameter, which is 1 if machine  $i$  is the first or the last machine needed for part  $j$ , and 0 otherwise.

### 2.2 Mathematical Formulation

The objective function of the proposed model is to minimize the total number of intercellular movements ( $f_1$ ) and total number of voids ( $f_2$ ) which can be formulated as below:

$$f_1 = \sum_{c=1}^C \sum_{i=1}^M \sum_{j=1}^P D_j \cdot \alpha_{ij} \cdot (2 - \beta_{ij}) \cdot y_{jc} \cdot (1 - x_{ic}); \quad (1)$$

$$f_2 = \sum_{c=1}^C \sum_{i=1}^M \sum_{j=1}^P D_j \cdot (x_{ic} \cdot y_{jc} - \alpha_{ij} \cdot x_{ic} \cdot y_{jc}); \quad (2)$$

$$\text{Minimize } f = f_1 + f_2$$

Subject to:

$$y_{jc} \leq \sum_{i=1}^M x_{ic}; \quad \forall j, c \quad (3)$$

$$\sum_{c=1}^C x_{ic} = 1; \quad \forall i \quad (4)$$

$$\sum_{c=1}^C y_{jc} = 1; \quad \forall j \quad (5)$$

$$\sum_{i=1}^M x_{ic} \geq L_c; \quad \forall c \quad (6)$$

$$x_{ic}, y_{jc} \in \{0, 1\}; \quad (7)$$

Constraint (3) implies that assignment of a part to a cell is subject to the presence of at least one machine in that cell. Constraints (4) and (5) ensure that any part or machine is assigned to only one cell. Constraint (6) maintains the size of the cells and guarantees that at least a predefined minimum number of parts will be assigned to each cell.

## 3. The ACO Algorithm

Ant colony optimization was first developed by Dorigo

*et al.* [21] based on the behavior of real ants. Real ants which live in colonies leave the nest to find food and come back again at every time. Based on observations, these ants always choose the shortest path to reach the food. As soon as this shortest path is found by some ants, the subsequent ants follow the same path. In fact there is a complicated communication system controlling the movement of ants. The secret of this communication is based on a substance, called *pheromone*. Real ants lay a substance known as pheromone on the ground when they pass through a path. This substance is smelled by other ants which leads them to follow the path traveled by prior ants. The more ants pass on a path, the more pheromone is put on that path. Since the shorter path is traveled faster, the density of pheromone on this path increases faster than other paths. Therefore, a majority of ants intend to travel on the shorter path after a given time. This is the underlying mechanism of ACO which is implemented to solve CFP in the following subsections.

### 3.1 Solution Representation and Evaluation

Suppose there are  $M$  machines and  $P$  parts to be clustered into  $C$  cells. The relation of machines and parts, which shows machine requirement of parts is normally represented by a matrix named. In this paper, the machine-part incidence matrix indicates the production process data as well. Specifically, an entry  $a_{ij} = k$  in the matrix, means that operation  $k$  of part  $j$  needs machine  $i$  for completion. **Table 2** shows an example including fifteen machines and twenty-five parts.

In the proposed algorithm, a solution is represented with a string of length  $M + P$ . The first  $M$  characters of the string show the cell number of the machines, and the rest are used for the parts. For example, for  $M = 5$ ,  $P = 4$ , and  $C = 2$ , a solution can be represented as "212112211" which means assign of machine 1 to 5 to cells 2,1,2,1 and 1 respectively and assignment of parts 1 to 4 to cells 2,2,1 and 1 respectively.

The generated solution may be infeasible, that is, the selected machine and part, are assigned to a wrong cell. In this case, the solution will be deleted, otherwise the goodness (or efficiency) of the solution will be computed.

### 3.2 Goodness Measurement

One of the most important steps in heuristic techniques is the evaluation of the obtained solutions. In this step, the *goodness* (or *fitness*) of the solution is calculated, and based on the result, the solution may be deleted, kept, or marked as good. The GA technique always keeps a population of feasible best fitted chromosomes (obtained solutions) and tries to achieve better solutions by mating the parents. Similarly, the ACO model keeps a list of best solutions ever found called *elite list*. When a new solution

**Table 2.** 15 × 25 machine-part matrix and demand for parts (*Dj*) of example 1

Parts	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	18	19	20	21	22	23	24	25	
<i>Dj</i>	59	95	30	46	13	34	12	63	57	74	98	5	93	75	22	24	61	100	26	56	19	67	97	24	47	
<b>M1</b>		1				1		1					1	3	3	1		1								
<b>M2</b>	2			2	2	4	2			5				2			2		8			2				
<b>M3</b>									6			6							6		1			6	5	
<b>M4</b>			2					5	3	2	2	2							2	1	2		2	2	2	
<b>M5</b>	3			3	3		3										3					3		7		
<b>M6</b>		3	4			3		2					3		2	3		4								
<b>M7</b>	1			1			1							1			1					1				
<b>M8</b>							7		2	4	5	4							4	4	4		4	4		
<b>M9</b>	4			4	4		4							4								4	5			
<b>M10</b>		4			1	5		4							4	4		3								
<b>M11</b>									5	6	4	5							5	5	5		6	5	4	
<b>M12</b>			1						1	1	1	1							1	2			1	1	1	
<b>M13</b>	5			5			5							5			4									
<b>M14</b>	6		3						4	3	3	3							3	3	3		3	3	3	
<b>M15</b>		2				2	6	3					2		1	2		2	7							

is obtained, the goodness (fitness) function is applied, and based on the result, we decide to add the solution to the elite list, or omit the solution and generate another one.

In this model, considering the objective function defined in Sub-section 2.3, the number of intercell movements and the number of voids in cells is to be minimized. So, the goodness function is defined as below:

$$goodness = f = \frac{1}{f_1 + f_2 + 1} \tag{8}$$

During the ACO, SA and GA iterations, the goodness of each solution is calculated using Equation (8). The constant value “1” is added to prevent division by zero.

### 3.3 The Ant Algorithm

Descriptive procedure of the proposed algorithm for solving the attempted mathematical model is as follows:

Begin

1. Initialize
2. Generate a feasible random solution, and add

it to the elite list.

3. Evaluate the efficiency (goodness) of the solution
  4. Repeat
    - a. Generate another random solution, based on pheromone trails.
    - b. Evaluate goodness, if better than the worst solution in the elite list, add it to elite list and delete the worst solution from list, update pheromone trails.
    - c. Evaporate pheromone
    - d. Alter solutions periodically
  5. Until stopping condition is met
- End

This model uses a  $\mathbf{P} = [P_{ck}]_{(C) \times (M+P)}$  pheromone matrix in which, *C*, *M*, and *P*, are the number of cells, machines and parts, respectively. The initial value of  $P_{ck}$  is 1. So, to generate a random feasible solution in step 3, there is no need to use the pheromone matrix.

## 4. Genetic Algorithm

In this method, an initial population of solutions (chromosomes) is generated randomly while subsequent populations are generated by choosing good parents and mating them. The mating may cause to worse, better, or even infeasible solutions. By keeping better solutions in population and omitting bad ones, the algorithm converges stage by stage and after a number of iterations, the local or global optimum will be found.

### 4.1 The GA Algorithm

Begin

1. Generate initial population containing N chromosomes.
  2. Compute the fitness of chromosomes in current population.
  3. Generate the next population
    - a. Choose two best parents randomly from the current population.
    - b. Mate the parents and generate two children (Crossover operator).
    - c. Apply the mutation operator.
    - d. Compute the fitness of children.
  4. Repeat step 3 until termination condition is met.
- End

In the proposed algorithm, the size of initial population is 1000 and the mating candidate parents are chosen by roulette wheel method. Chromosomes are represented as described in Subsection 3.1. The probability of crossing over and mutation is considered as 0.8 and 0.2 respectively.

## 5. Simulated Annealing

The Simulated Annealing (SA) algorithm is derived from metallurgy and thermodynamics which incorporated a temperature parameter into the minimization parameter. A high temperature expands the search space, and a lower temperature restricts the exploration. The procedure starts from a high temperature and ends at a low temperature. At each temperature, a number of iterations are done.

Some heuristic algorithms like Hill Climbing technique, may found the *Local Optimum* instead of the *Global optimum* because the movements leading to a new point worse than the current point are not allowed. SA algorithm allows non-improving movements to be taken in the hope of escaping the local optimum with a probability depending on the procedure temperature and the amount of the badness of the solution.

### 5.1 The SA Algorithm

Using the same representation described for solutions in Subsection 3.1, the SA algorithm can be written as follows:

Begin

1. Initialize Temp.
  2. Find a feasible solution, called x.
  3. Compute its goodness  $f(x)$ .
  4. Repeat until frozen
    - a. Do 1000 times
      - i.  $y = \text{FindNeighbour}(x)$ .
      - ii.  $\Delta = f(x) - f(y)$ .
      - iii. if  $\Delta > 0$  then  $x = y$ ; Accept y).
      - iv. else if  $U(0,1) < e^{-(\Delta)/\text{Temp}}$  then  $x = y$  (Accept y).
      - v. else reject y.
    - b. End Do.
  5.  $\text{Temp} = \text{Temp} * 0.95$ .
  6. The Solution will be best so far.
- End

The Algorithm starts at the temperature of 5000 with a feasible solution. The neighbor of a solution is obtained by making some changes in the solution (some parts or machines are randomly moved from one cell to other one).

## 6. Examples

The proposed algorithms are applied to solve two benchmark problems ( $15 \times 25$  and  $20 \times 35$  sizes) available in the literature. The algorithms are implemented in C Language and are executed on a Pentium IV PC.

### Example 1

The first example consists of fifteen machines and twenty-five parts which are grouped in three cells. The machine-part incidence matrix of the problem with is given in **Table 2**. The table also indicates demand of each part generated by a discrete uniform distribution in [1..100]. The solution obtained by the proposed ACO, SA and GA algorithms for this problem is shown in **Table 3**. The obtained values for  $f_1, f_2$  and  $f$  are 863, 803 and 1666 respectively in all three methods.

### Example 2

The second example is adopted from Boe and Cheng [22] with 35 parts, 20 machines and four cells. **Table 4** and **Table 5** show the machine-cell incidence matrix and the solution obtained by ACO, GA and SA algorithms. Since the demand for parts are not included in [22], these values are randomly generated using a discrete uniform distribution in [1..100]. The final obtained value of the objective function is 4562 in all three methods.

**Figures 1** and **2** show the number of iterations and convergence speed of different algorithms. In these figures, the horizontal axis shows the number of iterations (number of generations in GA, number of ants in ACO, and iterations of SA) and the vertical axis shows the value of the objective function.

**Table 3. The solution obtained by the proposed ACO, SA, and GA algorithms for example 1**

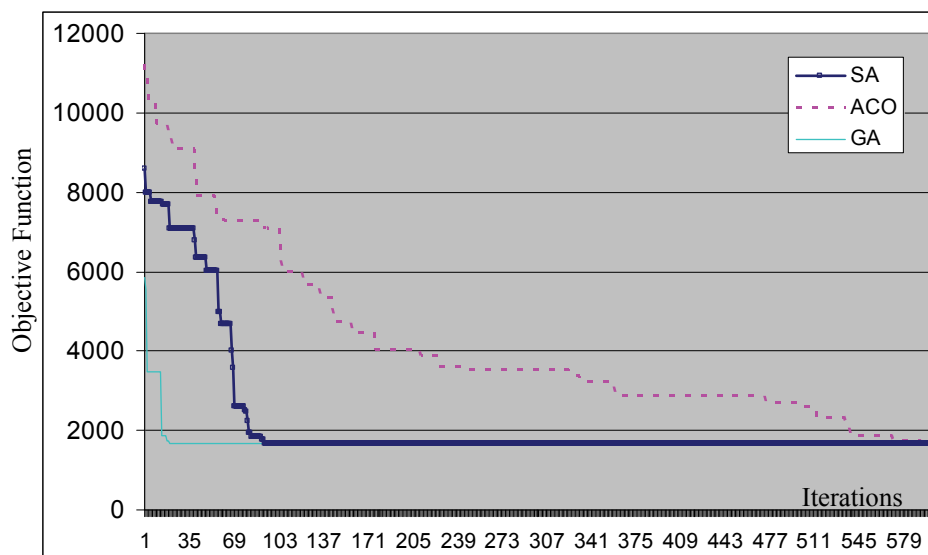
	1	4	5	7	14	17	22	2	6	8	13	15	16	18	3	9	10	11	12	19	20	21	23	24	25	
<b>M7</b>	1	1		1	1	1	1																			
<b>M2</b>	2	2	2	2	2	2	2		4								5			8						
<b>M5</b>	3	3	3	3		3	3																		7	
<b>M13</b>	4	4	4	4	4		4																	5		
<b>M9</b>	5	5		5	5	4																				
<b>M1</b>					3			1	1	1	1	3	1	1												
<b>M15</b>				6				2	2	3	2	1	2	2						7						
<b>M6</b>								3	3	2	3	2	3	4	4											
<b>M10</b>		1						4	5	4		4	4	3												
<b>M12</b>																1	1	1	1	1	1	2		1	1	1
<b>M4</b>									5							2	3	2	2	2	2	1	2	2	2	2
<b>M14</b>	6															3	4	3	3	3	3	3	3	3	3	3
<b>M8</b>				7												2	4	5	4	4	4	4	4	4	4	4
<b>M11</b>																5	6	4	5	5	5	5	5	6	5	4
<b>M3</b>																6			6	6		1		6	5	

**Table 4. The machine-cell incidence matrix of example 2**

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

**Table 5. The solution obtained by the proposed ACO, SA, and GA algorithms for example 2**

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



**Figure 1. Convergence speed of GA, ACO and SA for example**

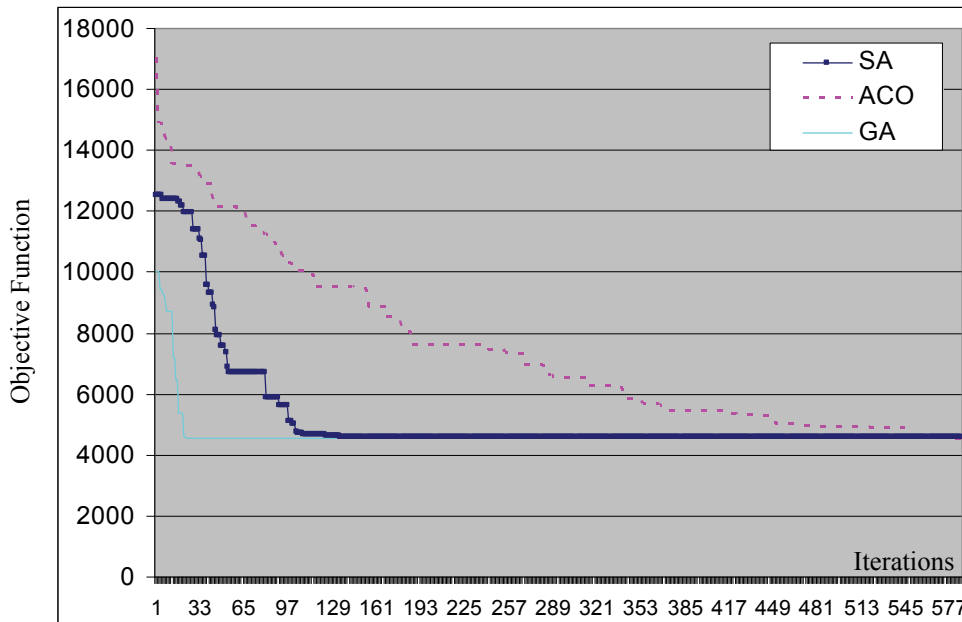


Figure 2. Convergence speed of GA, ACO and SA for example 2

Table 6. Computational time of GA, ACO and SA

Meta-Heuristic Method	Example 1	Example 2
GA	10	14
SA	37	48
ACO	251	263

The computational time (in seconds) of different algorithms for examples 1 and 2 are shown in Table 6.

7. Conclusions

This paper discusses that the sequence of operations and the production volume are two major factors to be considered in the design of CMS. Despite this fact, it has not been taken into account in a majority of researches available in the literature. To capture this fact, a new model for solving cell formation problem in CMS is proposed. Due to the NP-hardness of the formulated problem, three solution approaches based on ACO, GA and SA are used to solve the model. The objective function of the model is to minimize the total number of intercellular movements and the number of voids. The total number of cells is defined as a constant parameter in the algorithm.

The computational results show that the proposed algorithms are effective in minimizing the total number of

voids and intercellular movements.

As shown in Figures 1 and 2, the GA algorithm has obtained the optimum value faster than other techniques.

The attempted mathematical model can be further extended by considering alternate process plans for each part, machine redundancy, processing time of each operation, etc.

REFERENCES

- [1] C. Zhao and Z. Wu, "A Genetic Algorithm for Cell Formation with Multiple Routes and Multiple Objectives," *International Journal Production Research*, Vol. 38, No. 2, 2000, pp. 385-395.
- [2] C. C. Gallagher and W. A. Knight, "Group Technology Production Methods in Manufacturing," Knight/Ellis Horwood Limited, 1986.
- [3] J. S. Morris and R. J. Tersine, "A Simulation Analysis of Factors Influencing the Attractiveness of Group Technol-



- ogy and Cellular Layouts,” *Management Science*, Vol. 36, No. 12, 1990, pp. 1567-1578.
- [4] J. L. Burbidge, “Group-Technology in Engineering Industry,” Mechanical Engineering Publication Ltd., UK, 1979.
- [5] A. Ballakur and H. J. Steudel, “A within Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems,” *International Journal of Production Research*, Vol. 25, No. 5, 1987, pp. 639-655.
- [6] J. A. Joines, R. E. King and C. T. Culbreth, “A Comprehensive Review of Production-Oriented Manufacturing Cell Formation Technique,” *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 3, No. 3-4, 1996, pp. 225-265.
- [7] A. Islier, “Group Technology by Ant System Algorithm,” *International Journal of Production Research*, Vol. 43, No. 5, 2005, pp. 913-932.
- [8] G. Prabhakaran, A. Muruganandam and P. Asokan, “Machine Cell Formation for Cellular Manufacturing Systems Using an Ant Colony System Approach,” *International Journal of Advanced Manufacturing Technology*, Vol. 25, 2005, pp. 1013-1019.
- [9] K. L. Mak, P. Peng, X. X. Wang and T. L. Lau, “An Ant Colony Optimization Algorithm for Scheduling Virtual Manufacturing Systems,” *International Journal of Computer Integrated Manufacturing*, Vol. 20, No. 6, 2007, pp. 524-537.
- [10] K. Spiliopoulos and S. Sofianpoulou, “An Efficient Ant Colony Optimization System for the Manufacturing Cells Formation Problem,” *International Journal of Advanced Manufacturing Technology*, Vol. 36, No. 5-6, 2008, pp. 589-597.
- [11] S. E. Kesen, M. D. Toksari, Z. Gungor and E. Guner, “Analyzing the Behaviors of Virtual Cells (Vcs) and Traditional Manufacturing Systems: Ant Colony Optimization (ACO)-Based Metamodels,” *Computers and Operations Research*, Vol. 36, No. 7, 2009, pp. 2275-2285.
- [12] S. I. Satoglu and N. C. Surech, “A Goal Programming Approach for Design of Hybrid Cellular Manufacturing System in Dual Resource Constrained Environments,” *Computers and Industrial Engineering*, Vol. 56, No. 2, 2009, pp. 560-575.
- [13] Y. Kao and S. C. Fu, “An Ant-Based Clustering Algorithm for Manufacturing Cell Design,” *International Journal of Advanced Manufacturing Technology*, Vol. 28, 2006, pp. 1182-1189.
- [14] R. S. Pandian and S. S. Mahapatra, “Manufacturing Cell Formation with Production Data Using Neural Networks,” *Computers and Industrial Engineering*, Vol. 56, No. 4, May 2009, pp. 1340-1347.
- [15] I. Mahdavi, M. M. Paydar, M. Solimanpur and A. Heidarzadeh, “Genetic Algorithm Approach for Solving a Cell Formation Problem in Cellular Manufacturing,” *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 6598-6604.
- [16] I. Mahdavi, B. Shirazi and M. M. Paydar, “A Flow Matrix-Based Heuristic Algorithm for Cell Formation and Layout Design in Cellular Manufacturing System,” *International Journal of Advanced Manufacturing Technology*, Vol. 39, No. 9-10, 2008, pp. 943-953.
- [17] J. Arkat, M. Saidi and B. Abbasi, “Applying Simulated Annealing to Cellular Manufacturing System Design,” *International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 5-6, 2007, pp. 531-536.
- [18] A. Ahi, M. B. Aryanezhad, B. Ashtiani and A. Makui, “A Novel Approach to Determine Cell Formation, Intercellular Machine Layout and Cell Layout in the CMS Problem Based on TOPSIS Method,” *Computers and Operations Research*, Vol. 36, 2009, pp. 1478-1496.
- [19] X. Wang, J. Tang and K. Yung, “Optimization of the Multi Objective Dynamic Cell Formation Problem Using a Scatter Search Approach,” *International Journal of Advanced Manufacturing Technology*, Vol. 44, No. 3-4, 2009, pp. 318-329.
- [20] A. Muruganandam, G. Prabhakaran, P. Asokan and V. Baskaran, “A Memetic Algorithm Approach to the Cell Formation Problem,” *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 9-10, 2005, pp. 988-997.
- [21] M. Dorigo and G. D. Caro, “The Ant Colony Optimization Meta-Heuristic,” McGraw-Hill, New York, 1999.
- [22] W. J. Boe and C. Cheng, “A Close Neighbour Algorithm for Designing Cellular Manufacturing Systems,” *International Journal of Production Research*, Vol. 29, No. 10, 1991, pp. 2097-2116.