

# Two DEA Based Ant Algorithms for CMS Problems

Hossein Ali Akbarpour, Fatemeh Dadkhah

**Abstract**—This paper considers a multi criteria cell formation problem in Cellular Manufacturing System (CMS). Minimizing the number of voids and exceptional elements in cells simultaneously are two proposed objective functions. This problem is an Np-hard problem according to the literature, and therefore, we can't find the optimal solution by an exact method. In this paper we developed two ant algorithms, Ant Colony Optimization (ACO) and Max-Min Ant System (MMAS), based on Data Envelopment Analysis (DEA). Both of them try to find the efficient solutions based on efficiency concept in DEA. Each artificial ant is considered as a Decision Making Unit (DMU). For each DMU we considered two inputs, the values of objective functions, and one output, the value of one for all of them. In order to evaluate performance of proposed methods we provided an experimental design with some empirical problem in three different sizes, small, medium and large. We defined three different criteria that show which algorithm has the best performance.

**Keywords**—Ant algorithm, Cellular manufacturing system, Data envelopment analysis, Efficiency

## I. INTRODUCTION

THE cellular manufacturing system is an application of group technology, which is viewed as an efficient manufacturing philosophy. As known, the CMS is the best suited for a batch-flow production system in which many different products, having relatively low annual volumes, are produced intermittently in small lot sizes. Besides, this system can be adapted to a versatile market and rapid development of technology. With the rapid development of technology and short life cycles of new products in the current competitive market, the CMS approach has attracted the attention of many researchers and practitioners because of its practical utility. The most important problems in the design of CMS in the literature are the cell formation and its efficiency measurement procedures. Besides, there are few researches on the efficiency measurement of cell formation [1].

The DEA has been very sparingly applied to justification a number of analyses and operations decisions related to the advanced manufacturing system and technologies. Ertay proposed a framework based on the multi-criteria decision making for analyzing a firm's investment justification problem in a normal and high mold production technology to cope with the competition in the global market [1]. Ruiz Torres and Lopez considered problem of scheduling jobs on parallel machines in multi criteria environment [2].

H. A. Akbarpour is with the Department of Engineering, Marvdasht Branch, Islamic Azad University, Marvdasht, Iran (e-mail: akbarpour.ha@gmail.com).

F. Dadkhah is with the Department of Mathematic, Marvdasht Branch, Islamic Azad University, Marvdasht, Iran (e-mail: dadkhah\_f@miau.ac.ir).

They decided to minimize the makespan and the number of tardy jobs, simultaneously. To achieve this aim, they focused on simulated annealing algorithm and developed four different methods based on different initial solutions derived on benchmark. For evaluating the performance of proposed algorithms and identifying the most efficient algorithm, they used FDH formulation of DEA.

Mahdavi and his colleagues [3] presented a new mathematical model for cell formation in cellular manufacturing system based on cell utilization concept. The objective of the model is minimizing the number of voids of each cell to achieve the higher performance of cell utilization. Also they [4] proposed cell formation problem in cellular manufacturing and presented a model with non linear constraints and integer variables. The objective of the model is to minimize the number of voids and exceptional elements. Their model was Np\_hard and cannot be solved for real sized problems efficiently; they developed a genetic algorithm to solve it. In proposed algorithm they introduced a new chromosome scheme to assign the parts and machines to cells.

The following of paper organized in four different sections. In section II a DEA background and its different features are presented. Also we provided a numerical example to better illustration of the proposed algorithm. General structure of ant algorithms with developed methods is presented in section III. An experimental design including data generation, parameters setting and computational results is provided in section IV. The conclusion of paper and some future works are presented in section V.

## II. DATA ENVELOPMENT ANALYSIS

General structure of DEA has been introduced by Farrel in 1954 for the first time. Based on this article some researchers worked on this new concept and developed two models; CCR by Charnes, Cooper and Rhodes in 1978 [5] and BCC by Banker, Charnes and Cooper in 1984 [6]. Now there are other models such as FDH, BCC-CCR and CCR-BCC. But the BCC and CCR models are the basic models in DEA. The DEA is a linear programming based method which evaluates relative efficiency of Decision Making Units (DMUs). It can include multiple outputs and inputs without a priori weights and without requiring explicit specification of functional forms between inputs and outputs. It computes a scalar measure of efficiency and determines efficient levels of inputs and outputs for each DMU under evaluation which has a range of zero to one [7]. In fact, the DEA solves a linear programming to evaluate efficiency score of different decision making units relatively. Each DMU can have some inputs and outputs with

different weights. In case of one input and one output one can divide value of output by value of input for evaluating the efficiency score of an especial DMU. But, in real management problems usually there are many different parameters either with or without specific weights which effect the determination of efficiency score of a DMU. In this case one should challenge with a decision making problem. In this research, in order to triumph on formed decision making problem, we used BCC input oriented model of DEA technique. We considered each individual of population in proposed GA (each chromosome) as a DMU. As mentioned, in DEA each DMU can have input and output one or more. In proposed GA we supposed that each DMU has two inputs which are makespan and cumulative tardiness. Also we supposed that all DMUs have identical outputs, all of them give us processed jobs. In order to employ the DEA technique in GA, we provided a numerical example and illustrated efficiency score, efficient frontier and ranking of DMUs.

*Example:* Suppose that five DMUs A, B, C, D and E with identical outputs and two different inputs achieved from five various chromosomes, are as Table 1. Table 2 shows the efficiency score of each DMU according to BCC model. Also we provided the efficient frontier in Fig. 1.

In order for ranking the DMUs, first, we divided the makespan of each DMU by the tardiness of it and then, sort them based on the minimum distance to 1, e.g., rank of C is better than rank of A.

TABLE I  
INPUTS AND OUTPUT OF DMUS

DMU	A	B	C	D	E
Input1	2	5	3	4	6
Input2	5	2	2	4	1
Output	1	1	1	1	1

TABLE II  
EFFICIENCY SCORE OF DMUS

DMU	A	B	C	D	E
Efficiency	1	0.82	1	0.73	1

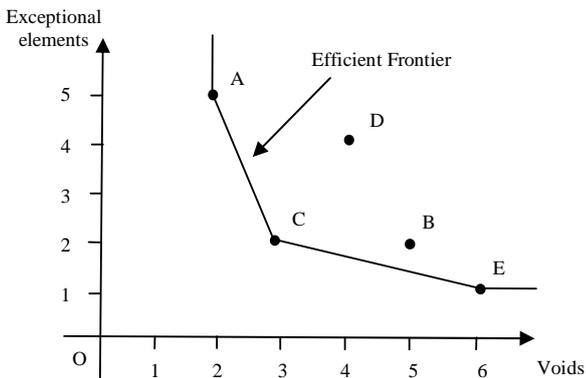


Fig. 1 BCC input oriented efficient frontier generated from observed data

### III. PROPOSED ALGORITHMS

#### A. General structure of an ant algorithm

Meta-heuristics such as genetic algorithms, simulated annealing and tabu search are used to solve flow shop scheduling problems. In recent times, attempts are being made to solve combinatorial optimization problems by making use of artificial ant algorithms. Ant algorithms were first proposed by Dorigo and colleagues as a multi-agent approach to difficult combinatorial optimization problems like the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP). Ant algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest [8]. In this paper we developed two ant algorithms for the CMS problem. In both of them we used the solution scheme which defined in [4].

#### B. ACO Algorithm

The first algorithm which we have developed it in this research works based on ACO. At first we present the principal structure of that. It has three steps as following.

##### Step1. Initialization

- a) Parameters setting
- b) Initial generation
- c) Pheromone initialization

##### Step2. Doing while stopping criteria is not met

- a) Applying for all artificial ants
  - i) Constructing a complete solution using semi probability selection rule
  - ii) Improving the solution by applying a local search
  - iii) Local pheromone updating
- b) Global pheromone updating

##### Step3. Printing the best achieved solution

In step1 of proposed ACO algorithm, after adjusting effective parameters, for each artificial ant a complete randomly solution generated and based on the best solution initialized the pheromone trial. The step2 is the master loop that iterated till stopping criteria is satisfied. This step contains two paces which the first pace applied for all artificial ants. In this pace, for each ant a solution is constructed by use of the semi probability selection rule as shown in (1). For this aim the random of  $q$  generated from a uniform distribution  $U(0, 1)$ .

$$q \leq q_0 : j = \arg \max_{j \in N_i^k} ([\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta)$$

$$q > q_0 : p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, j \in N_i^k, \eta_{ij} = 1/d_{ij} \quad (1)$$

In order to apply the local search we investigate two methods. In the first method presented in Fig. 2, sequence of cells is inverted. If it causes to better solution that is replaced the former. Otherwise the second method presented in Fig. 3, which is pair-wise exchange with cyclic exchange applied. If achieved solution is dominant on the former exchange them.

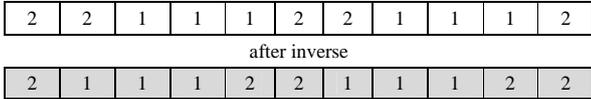


Fig. 2 Inverse Local Search

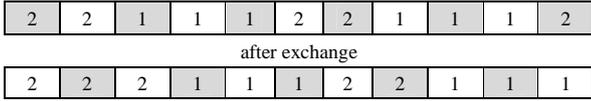


Fig. 3 Pair-wise Exchange with Cyclic Exchange

The local pheromone trail updating applied after that each artificial ant produced a solution by (2).

$$\tau_{ij}(t) = (1 - \rho') \cdot \tau_{ij}(t) + \rho' \cdot \tau_0 \quad (2)$$

The general updating of pheromone trail carried out based on the best solution by (3).

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}^{best}(t), \forall (i, j) \in best \quad (3)$$

### C. MMAS Algorithm

The second algorithm of this paper has proposed MMAS. We provided the structure of that step by step.

#### Step1. Initialization

- a) Parameters setting
- b) Initial generation
- c) Pheromone initialization

#### Step2. Doing while stopping criteria is not met

- a) Applying for all artificial ants
  - i) Constructing a complete tour using probability selection rule
  - ii) Improving the solution by applying a local search
- b) Global pheromone updating

#### Step3. Printing the best achieved solution

In step1 of proposed MMAS algorithm, after adjusting effective parameters, for each artificial ant a complete randomly solution generated. In MMAS all routes get maximum amount of pheromone trail. The step2 is the master loop that iterated till stopping criteria is satisfied. This step contains two paces which the first pace applied for all artificial ants. In this pace, for each ant a solution is constructed by use of the probability selection rule as shown in (4).

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, j \in N_i^k, \eta_{ij} = 1/d_{ij} \quad (4)$$

In order to apply the local search for improving each solution we applied two methods as well as the ACO algorithm which represented in Fig. 2 and Fig. 3.

The general updating of pheromone trail is carried out by (3). Of course there is a lower bound and upper bound which update in each iteration by (5).

$$\tau_{min} \leq \tau_{ij}(t) \leq \tau_{max}, \tau_{max}^* = 1/(\rho \cdot f(S^*)) \quad (5)$$

$$\tau_{min}^* = q \times \tau_{max}^*, q \in U(0,1)$$

The  $f(S^*)$  achieved from summation of two objective functions, because both of them have same worthiness.

## IV. EXPERIMENTAL DESIGN

### A. Data Generation

We generated experimental data to evaluate and compare the performance of methods that developed based on ant algorithm in this research. To generate experimental problem we considered three sets based on different size of problem, correspond to carry out researches, consist of small, medium and large size. In each size two important factors are available; the first factor is number of machines and the number of parts is the second. We considered 5, 10 and 20 for the former and 10, 20 and 50 for the latter, in small, medium and large size, respectively. In each size we defined a random variable with probability of 0.4, so that it determines each part requires to what machines to process, and to determine of this dependency generated a random number from a uniform distribution  $U(0, 1)$ , if this is less than 0.6 then the part requires to that specific machine. This variable helps to generation of more different problems such that we generated 15 different problems by variation of this variable, in each size. So we generated 45 different problems in three variety sizes which will run 10 different replicates for all of them. We summarized this subject in Table III.

TABLE III  
CONSIDERED LEVELS FOR FACTORS

	Small	Medium	Large
Number of machines	5	10	20
Number of parts	10	20	50
Skipping probability	0.4	0.4	0.4

### B. Parameters Setting

Performance of each algorithm is affected by some various parameters, significantly. If these values aren't selected correctly, appropriate results won't obtain. In order to select the parameters that result solutions with high quality, we considered problems in three different sizes that described before and selected some problems as a sample in each size. Sample size is 6 for small, medium and large problems. In this paper we considered some of the important factors with different levels for proposed algorithm. These factors and their levels are shown in Table IV.

We run proposed algorithm ten independent replicates, by combination of different factors represented in Table IV and selection the best combination according to results of them. Minimizing two considered objective functions simultaneously is our measurement. However, the CPU time is an important

TABLE IV  
PROPOSED ALGORITHM FACTOR LEVEL

Factor	Levels
Pheromone evaporation ( $\rho$ )	0.01, 0.02, 0.03
Tunable parameter ( $q0$ )	0.1, 0.3, 0.5
Number of iteration ( $maxiter$ )	25, 50, 75 (s) 50, 100, 150 (m) 150, 200, 300 (l)
Number of initial population ( $antnum$ )	5, 10, 20 (s) 10, 20, 25 (m) 20, 50, 60 (l)

criterion to realize the best factor values. After tuning all parameters except  $maxiter$ , we fixed the best obtained parameter values and found the best value of  $maxiter$ . The obtained values for every factor in all three different sizes are shown in Table 5.

C. Computational Results

TABLE V  
BEST VALUES FOR PROPOSED ALGORITHMS PARAMETERS

Size	$\rho$	$q0$	$maxiter$	$antnum$
Small	0.01	0.3	50	10
Medium	0.01	0.3	100	20
Large	0.01	0.3	300	50

TABLE VI  
RAS CRITERION COMPARISON

Size	ACO		MMAS	
	Mean	St. Dev	Mean	St. Dev
Small	1.0927	0.6662	1.1333	0.7211
Medium	4.571	1.588	4.752	1.631
Large	16.035	2.055	16.189	2.152

TABLE VII  
MID CRITERION COMPARISON

Size	ACO		MMAS	
	Mean	St. Dev	Mean	St. Dev
Small	7128	10486	9595	14451
Medium	32699	12781	30454	14695
Large	209909	52805	186375	42522

TABLE VIII  
CPU TIME CRITERION COMPARISON

Size	ACO		MMAS	
	Mean	St. Dev	Mean	St. Dev
Small	1.9417	0.3751	1.9959	0.4111
Medium	12.340	1.589	13.022	1.616
Large	111.03	14.44	118.75	16.00

The results of proposed algorithm performance to solve considered problem is presented in this section. Both of two scenarios are coded in *MATLAB 7.1* and are carried out 10 independent runs. Every run records all the non repeated Pareto optimal solutions. Scenarios run on a PC with a Pentium IV 3.0 GHz processor with 512 MB of RAM and Windows Xp professional operating system.

In order to measure the performance of presented algorithm, we considered three criteria as MID, RAS and CPU time. The CPU time is a known criterion; therefore, we explained the two others as (1) and (2).

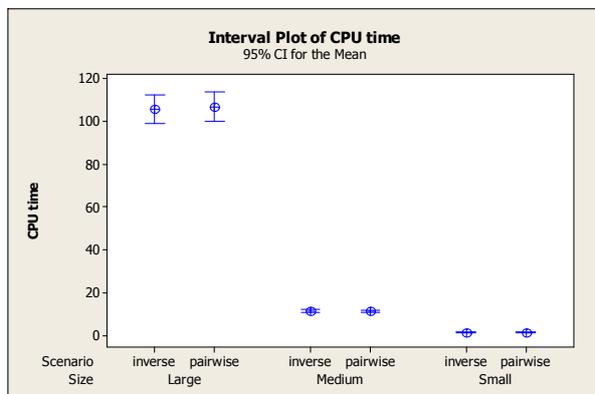
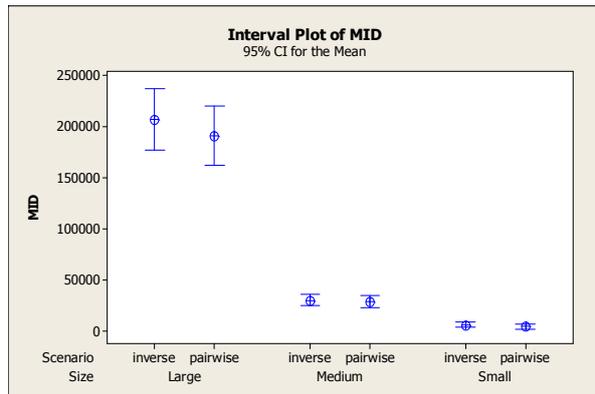
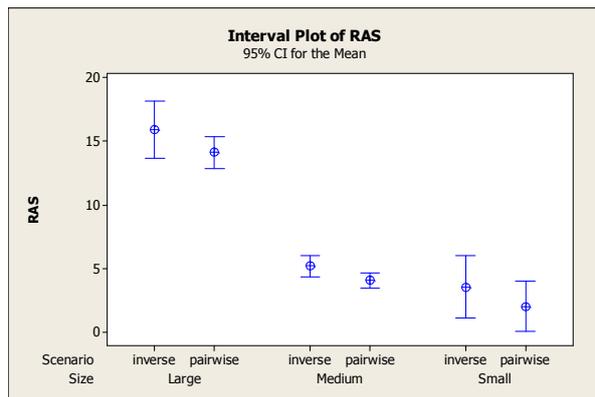


Fig. 4 Interval plot of RAS, MID and CPU time criteria in three sizes

$$MID = \frac{\sum_{i=1}^n c_i}{n} \quad \ni \quad c_i = \sqrt{f_{1i}^2 + f_{2i}^2} \quad (6)$$

$$RAS = \frac{\sum_{i=1}^n \left( \frac{f_{i1} - F_i}{F_i} \right) + \left( \frac{f_{i2} - F_i}{F_i} \right)}{n}, \quad F_i = \min \{f_{1i}, f_{2i}\} \quad (7)$$

According to (6) and (7), the smallest value of MID or RAS criterion shows the best performance. In order to evaluate the performance of proposed algorithm we selected non dominated solutions which positioned on efficient frontier with efficiency score of 1. Then we calculated the MID and the RAS of all these solutions using (6) and (7).

Since the performance of meta-heuristic depends on used parameters, intensely, the elimination of Pareto archive set number is an important advantage which is performed by DEA in proposed algorithm. This feature causes not to need to determine number of optimal solutions for evaluating of algorithm performance.

We compared two different scenarios performance by two criteria introduced in (6) and (7) and provided the results in Tables VI and VII. The results show that there are no significantly differences between two algorithms. Also we provided the interval plot of proposed criteria in Minitab 16 statistical software as shown in Fig. 4 in order to more explanation the performance.

#### V.CONCLUSION

In this research we proposed a multi criteria CMS problem with the aim of minimizing exceptional elements and voids. We employed the DEA technique to develop two robust ant algorithms. We considered both objective functions as inputs for each DMU with the same output of one. Algorithms implemented in reasonable time. Two proposed algorithms implement in a reasonable time and it seems that both of them are usable in industries. As a future work we can employ the fuzzy logic in proposed algorithms and compare its performance with DEA.

#### REFERENCES

- [1] T. Ertay, and D. Ruan, "Data envelopment analysis based decision model for optimal operator allocation in CMS," *European Journal of Operational Research*, 2005, 164, 800–810.
- [2] A. J. Ruiz-Torres, and F. J. López, "Using the FDH formulation of DEA to evaluate a multi-criteria problem in parallel machine scheduling," *Computers & Industrial Engineering*, 2004, 47, 107–121.
- [3] I. Mahdavi, B. F. Javadi, K. Alipour and J. Slomp, "Designing a new mathematical model for cellular manufacturing system based on cell utilization," *Applied Mathematics and Computation*, 2007, 190, 662–670.
- [4] I. Mahdavi, M. M. Paydar, M. Solimanpur and A. Heidarzade, "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing," *Expert Systems with Applications*, 2009, 36, 6598–6604.
- [5] A. Charnes, W.W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European Journal of Operation Research*, 1978, 429–444.
- [6] R. D. Banker, A. Charnes, and W.W. Cooper, "Some models for estimating technical and scale inefficiencies in data envelopment analysis," *Management Science*, 1984, 30, 1078–1092.
- [7] M. R. Alirezaee and M. Afsharian, "A complete ranking of DMUs using restrictions in DEA models," *Applied Mathematics and Computation*, 2007, 189, 1550–1559.
- [8] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," *Artificial Life*, MIT Press, 1999.