

SOLVING A NEW BI-OBJECTIVE MODEL FOR A CELL FORMATION PROBLEM CONSIDERING LABOR ALLOCATION BY MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

R. Tavakkoli-Moghaddam and A. Ghodratnama*

*Department of Industrial Engineering, College of Engineering, University of Tehran
P.O. Box 11155-4563, Tehran, Iran
tavakoli@ut.ac.ir, ghodratn@ut.ac.ir*

A. Makui

*Department of Industrial Engineering, Iran University of Science and Technology
Postal Code 1684613114, Tehran, Iran
amakui@iust.ac.ir*

M. Khazaei

*Department of Industrial Engineering, University of Payam Noor
Postal Code 1084668611, Tehran, Iran
mk_khazaei@yahoo.com*

*Corresponding Author

(Received: November 04, 2009 – Accepted in Revised Form: September 15, 2011)

doi: 10.5829/idosi.ije.2011.24.03a.04

Abstract Mathematical programming and artificial intelligence (AI) methods are known as the most effective and applicable procedures to form manufacturing cells in designing a cellular manufacturing system (CMS). In this paper, a bi-objective programming model is presented to consider the cell formation problem that is solved by multi-objective particle swarm optimization (MOPSO) algorithm. The model contains two conflicting objectives, namely optimal labor allocation and maximization of cell utilization. In order to verify its effectiveness of the MOPSO algorithm, the results are compared with those obtained from a well-known evolutionary procedure, called NSGA-II.

Keywords Cellular Manufacturing System, Cell Formation, Labor Allocation, Multi-Objective Particle Swarm Optimization

چکیده روش های برنامه ریزی ریاضی و هوش مصنوعی به عنوان موثرترین و کاراترین روش های تشکیل سلول در طراحی سیستم های تولید سلولی شناخته می شوند. در این مقاله، یک مدل برنامه ریزی دو هدفه برای مسأله تشکیل سلولی ارایه و سپس با روش فراابتکاری بهینه سازی گروهی ذرات چند هدفه (MOPSO) حل می شود. این مدل شامل دو هدف متضاد با نام های تخصیص نیروی انسانی بهینه و بیشینه سازی کارایی سلول است. همچنین نتایج مرتبط با الگوریتم MOPSO مزبور با نتایج حاصل از روش فرا ابتکاری تکاملی NSGA-II به منظور اثبات کارآمدی روش پیشنهادی مقایسه می شود.

1. INTRODUCTION

Group Technology (GT) is a managerial philosophy that attempts to divide products into independent groups according to their design and process similarities. A cellular manufacturing system (CMS) is the most important application of GT in industrial environments that joins the

efficiency of a flow shop system with the flexibility of job shop. Because of this reason it is known as one of the best alternatives to be employed in companies with high amount of demand and design's changes.

The formation of manufacturing cells is the most critical step in designing a CMS based on the given problem, in which some different parameters can

be considered in this phase. Joins, et al. [1] proposed a comprehensive classification of methods for a cell formation problem (CFP). These are as follows: 1) array-based, 2) hierarchical, 3) non-hierarchical, 4) heuristic, 5) graph partitioning, 6) artificial intelligence, and 7) mathematical programming methods. Selim, et al. [2] presented another classification in this area that summarizes the first four mentioned methods into one category, called "clustering analysis". This classification also contains one added branch, called "descriptive methods". It is necessary to explain that some different factors can be considered in the cell formation phase, such as the number of exceptional elements (EEs), cell loading unbalances, cell constructing costs, cell utilization, labor related issues, and the like. Thus it is so clear that the effectiveness of the designed CMS directly depends on considered factors.

In the last three decades, a numerous number of studies have been implemented to develop new and more effective methods to forming manufacturing cells in CMS. Considering the proposed procedures it is obviously proved that the artificial intelligence and mathematical programming methods are much more efficient and applicable than others.

Onwunbolu and Mutingi [3] proposed two independent mathematical models for the CFP that contain minimization of cell loading unbalances and number of exceptional elements, respectively. They employed a genetic algorithm (GA) to solve their models. Gungor and Arikan [4] presented a mathematical model based on fuzzy parametric programming procedure in order to identify part families and group machines to manufacturing cells where parameters such as EE elimination costs, part demands and machine capacities were fuzzy. Rogers and Kulkarni [5], developed a mathematical model based on fuzzy parametric programming which considered machine capacities, part demands and removing cost of exceptional elements (EEs) as fuzzy parameters. Peker and Kara [6] also applied ART neural network to solve the CFP. Some other researches in this area are available in [7-12]. Andres and Lozano [13] applied particle swarm optimization (PSO) to solve their proposed model that minimizes the number of EEs. Tavakkoli-Moghaddam, et al. [14] proposed an efficient algorithm to allocate parts to cells maximizing cell efficiency, the reported results of simulated

annealing (SA) algorithm shows the capability of this proposed algorithm in order to solve the considered mathematical model.

Most of authors considered their developed mathematical models in single form and just a few numbers of efforts have been accomplished to apply multi-objective models (MOMs) for solving the cell formation problem. Kim, et al. [15] developed a bi-objective model for solving the CFP that minimizes the inter-cell movements and cell loading unbalances simultaneously. Lei and Wu [16] presented a MOM for machine-part grouping problems with three conflicting goals, namely minimizing total cost, inter-cell and intra-cell loading unbalances. They designed a multi-objective tabu search (MOTS) to solve their proposed model. Tavakkoli-Moghaddam and Minaeian [17] proposed a comprehensive, multi-objective, mixed-integer, nonlinear programming (MINLP) model for a cell formation problem (CFP) under fuzzy and dynamic conditions aiming at: (1) minimizing the total cost which consists of the costs of inter-cell movements and subcontracting parts as well as the cost of purchasing, operation, maintenance and reconfiguration of machines; (2) maximizing the preference level of the decision making (DM); and (3) balancing intra-cell workload. Dimopoulos [18] developed a MOM for the manufacturing cell design and applied a well-known evolutionary algorithm, called NSGA II. Wu, et al. [19] presented a bi-objective model for simultaneous considering of cell formation and layout problems as two important steps in designing the CMS. Bajestani, et al. [20] considered the cell formation problem under the dynamic condition via developing a multi-objective formulation and solved it by a proposed scatter search (SS) algorithm. Tavakkoli-Moghaddam, et al. [21] presented a novel, multi-objective mixed-integer nonlinear programming (MINLP) model for a cell formation problem (CFP) with alternative process routes, in which the considered objectives are: (1) minimizing the total cost consisting of; inter-cell movements, purchasing, operation, and maintenance; 2) maximizing the utilization of machines in the system; and 3) minimizing the deviation levels between the cell utilization.

By reviewing the studies in designing a CMS published in the recent decades, it can be found out that the most of researchers in this area, have focused on technical aspects and only a few

numbers of studies have been carried out to consider the related labor issues. Cesani and Steudel [22] presented a vast study of labor flexibility in cellular manufacturing systems characterized by intra-cell operator's mobility. They proposed a classification scheme to evaluate labor assignment and a framework for comparing its strategies. Suresh and Slomp [23] presented a hierarchical procedure to design a CMS considering labor assignment. Suer [24] developed a three-phase methodology to design manufacturing cells in labor intensive industries where the number of operators is more than the number of machines. Bidanda, et al. [25] implemented a comprehensive review of human related issues in CMS and classified them into eight groups based on the evaluation of a diverse range of published articles.

According to above mentioned description, only a few numbers of researchers have been developed multi-objective models to design a CMS considering human resource requirements. Slomp, et al. [26] developed a goal programming approach to design virtual manufacturing cells under the dynamic conditions considering labor allocation. Furthermore, Slomp, et al. [27] also analyzed the importance of human cross training in the CMS. Suer, et al. [28] analyzed applying of different fuzzy operators by developing a fuzzy bi-objective model in a labor-intensive cellular environment. Satoglu and Suresh [29] applied a goal programming approach to design a hybrid cellular manufacturing system (HCM) considering human resource requirements. Their proposed procedure contained three phases, namely Pareto analysis of demand volumes, machine grouping, and labor allocation.

In this paper, we present a bi-objective model with two conflicting objectives in order to form manufacturing cells in a CMS. The proposed model optimizes the assignment of labors into cells and maximizes cell utilization simultaneously. Since the proposed model is a NP-Hard problem, a multi-objective particle swarm optimization (MOPSO) algorithm is designed to solve it. Results show that the proposed MOPSO is more robust and effective than the well known NSGAI.

2. MULTI-OBJECTIVE OPTIMIZATION

For most of real world problems, very often that is

hard to define all the aspects of a given problem in terms of a single objective. Using a multi-objective form of modeling is much better idea where the considering criteria are in conflict with each other [30].

2.1. Basic Concepts A general form of a multi-objective problem (MOP) can be formulated as follows:

$$\text{Min/Max } \vec{F}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]$$

s.t.

$$g_i(\vec{x}) \geq 0 \quad ; \quad i=1, \dots, m$$

$$h_j(\vec{x}) = 0 \quad ; \quad j=1, \dots, p$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $\vec{f}_i : \mathfrak{R}^n \rightarrow \mathfrak{R}$; ($i=1, 2, \dots, k$) are the objective functions and $g_i, h_j : \mathfrak{R}^n \rightarrow \mathfrak{R}$; ($i=1, 2, \dots, m, j=1, 2, \dots, p$), are the constraint functions of the considered problem.

A few definitions are introduced to describe the basic principles in multi-objective formulation.

Definition 1 (Dominance concept): for a given MOP, vector $\vec{x} = [x_1, x_2, \dots, x_n]^T$ dominates vector $\vec{x}' = [x'_1, x'_2, \dots, x'_n]^T$ when always $f_i(\vec{x}) < f_i(\vec{x}')$ for $i=1, 2, \dots, k$.

Definition 2 (Pareto optimality): vector $\vec{x}^* \in X$ is called Pareto optimal if and only if there is no vector $\vec{x}' \in X$ which dominates \vec{x}^* . It means:

$$\forall x \in X \quad \exists \quad \vec{x}' \in X \quad ; \quad \vec{F}(\vec{x}') < \vec{F}(\vec{x}^*)$$

Definition 3 (Pareto optimal set): The Pareto Optimal Set P^* is defined by:

$$P^* = \{\vec{x} \in X \quad \exists \quad \vec{x}' \in X \quad ; \quad \vec{F}(\vec{x}') \leq \vec{F}(\vec{x})\}$$

Definition 4 (Pareto optimal set): The Pareto Front PF^* is defined by:

$$PF^* = \{\vec{u} = \vec{F}(\vec{x}) \quad \vec{x} \in P^*\}$$

2.2. Evolutionary Algorithm (EA) Classical

methods of solving MOPs (e.g., the weighted sum of objective functions, goal programming, lexicographic procedure and the like) use a point-by-point approach, in which one solution at each iteration is modified to a different solution. Thus the outcome of using a classical optimization method is a single optimized solution [30]. While evolutionary algorithms (EAs) use a population of solutions at each iteration, instead of a single one. In other words, the use of EAs to solve problems of this nature has been motivated mainly because of the population-based nature of EAs that allows the generation of several elements of the Pareto optimal set in a single run [31].

3. PROBLEM FORMULATION

In this section, we formulate the proposed bi-objective model that contains two independent goals conflicting with each other. First objective was first presented by Satoglu and Suresh [29] to minimize labor related costs as well as over assignment of them into cells. The second one tries to form manufacturing cells via maximizing the cell utilization value. Tables 1 and 2 show the input parameters and decision variables, respectively.

3.1. Assumptions A number of assumptions of the proposed model are presented as follows:

1. The demand of all products is constant (i.e., the problem is considered under the static conditions).
2. Part families and machine groups are determined simultaneously.
3. Machine capacity is not considered.
4. Machine duplication is not allowed.
5. All machines are available during the planning horizon.
6. Work sharing is not allowed for any workers.

3.2. Indices

i index of parts; $i=1,2,\dots,N$
 j index of machines; $j=1,2,\dots,M$
 k index of cells; $k=1,2,\dots,C$
 l index of labors; $l=1,2,\dots,L$

3.3. Input Parameters

$$P_{lj} = \begin{cases} 1 & \text{if worker } l \text{ is capable to work on machine } j \\ & \text{or will be capable to be cross - trained} \\ & \text{working on at the cost of } \varphi_{lj} \\ 0 & \text{Otherwise} \end{cases}$$

$$\varphi_{lj} = \begin{cases} 1 & \text{Cost of cross - training worker } l \\ & \text{to work on machine } j \\ 0 & \text{Otherwise} \end{cases}$$

H_l, F_l = Hiring, firing cost for worker l

$$a_{ij} = \begin{cases} 1 & \text{if part } i \text{ is processed by machine } j \\ 0 & \text{Otherwise} \end{cases}$$

LB_m, UB_m = Upper bound and lower bound of the machines number in each opened cell.

LB_l, UB_l = Upper bound and lower bound of the labor's number in each opened cell.

$UB_{LMachine}$ = Maximum number of machines that an operator can be served.

C_{max} = Maximum number of cells allowed to be opened.

Ω = A large constant scalar.

3.4. Decision Variables

$$h_l = \begin{cases} 1 & \text{if worker } l \text{ is employed} \\ 0 & \text{Otherwise} \end{cases}$$

$$Z_{lk} = \begin{cases} 1 & \text{if worker } l \text{ is assigned to cell } k \\ 0 & \text{Otherwise} \end{cases}$$

$$x_{ik} = \begin{cases} 1 & \text{if part } i \text{ is assigned to cell } k \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{jk} = \begin{cases} 1 & \text{if machine } j \text{ is assigned to cell } k \\ 0 & \text{Otherwise} \end{cases}$$

$$\delta_k = \begin{cases} 1 & \text{if cell } k \text{ is constructed} \\ 0 & \text{Otherwise} \end{cases}$$

$$P_{lj} = \begin{cases} 1 & \text{if the capability is } P_{lj} \text{ utilized (at cost } \varphi_{lj}) \\ 0 & \text{Otherwise} \end{cases}$$

D_l = Deviation variable showing that the number of cells worker L is assigned over the target.

By using the introduced symbols and parameters, the proposed mathematical bi-objective model is formulated as follows:

$$\text{Min } w_1[\sum_l D_l] + w_2[\sum_l H_l h_l + F_l(1 - h_l)] + w_3[\sum_l \sum_i p_{ij} p_{lj}] \quad (1)$$

$$\text{Max } \sum_i^N \sum_j^M \sum_k^{C_{Max}} a_{ij} x_{ik} y_{jk} \quad (2)$$

s.t.

$$\sum_k Z_{lk} = h_l(1 + D_l) \quad \forall l \quad (3)$$

$$\sum_j p_{lj} \leq h_l \Omega \quad \forall l \quad (4)$$

$$p_{lj} \leq P_{lj} \quad \forall l, j \quad (5)$$

$$Z_{lk} \leq \sum_j p_{lj} y_{jk} \quad \forall l, k \quad (6)$$

$$\sum_j p_{lj} = 1 \quad \forall l \quad (7)$$

$$\sum_j p_{lj} \leq UB_{LMachine} \quad \forall l \quad (9)$$

$$LB_L \leq \sum_l h_l \leq UB_L \quad (10)$$

$$\sum_k x_{ik} = 1 \quad \forall i \quad (11)$$

$$\sum_k y_{jk} = 1 \quad \forall j \quad (12)$$

$$x_{ik} \leq \delta_k \quad \forall i, k \quad (13)$$

$$\delta_k LB_M \leq \sum_l h_l \leq \delta_k UB_M \quad \forall k \quad (14)$$

$$\sum_k \delta_k \leq C_{Max} \quad (15)$$

$$p_{lj}, h_l, Z_{lk}, x_{ik}, y_{jk}, \delta_k \in \{0,1\}; \quad (16)$$

$$D_l = \{0, \dots, C_{Max} - 1\} \quad (17)$$

The first objective function of the proposed bi-objective model consists of three conflicting terms

with weights of w_1 through w_3 as their determined preferences by the decision maker. The first term attempts to minimize over assignments of labors through manufacturing cells, where D_l is the positive deviation variable for this goal and also is related to Constraint 3. The second one tries to minimize the total hiring and firing costs based on h_l that mainly is a binary integer variable. Finally, the last one minimizes the total cross-training cost of all employed labors based on p_{lj} variables and the corresponding cross-training costs of ϕ_{lj} .

The first nine constraints of the model are related to the labor assignment objective and the others are defined for the cell formation phase. Constraint (3) controls over assignments of employed workers and includes deviation variable of D_l . Constraints (4) ensure that the unemployed labors should not be assigned to any machines. Constraint (5) ensures that no labor must be assigned to cells which have not opened. Constraint (6) ensures that capabilities made available (p_{lj}) are limited to the potential capabilities (P_{lj}). Constraint (7) guarantees that an employed labor can be assigned to the specific manufacturing cell when the labor is allocated to at least one machine of that special cell. Constraint (8) determines that each machine can be serviced by only one labor. According to the limited human capacity, employed labors is not able to service more than $UB_{LMachine}$ machines as considered in Constraint (9). Constraint (10) says that the number of employed operators in the whole system must not exceed the desired interval. Constraints (11) and (12) determine that each part and machine must be allocated to only one cell, respectively. Constraint (13) guarantees that if a manufacturing cell has not opened, no part must be assigned to it. Constraint (14) controls the upper and lower bounds of the number of assigned machines in each opened cell. Finally according to Constraint (15), it is not allowed to open more than the predefined number of cells, called C_{max} .

4. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population-based algorithm that was first presented by Kenedy and Eberhart [32] by analyzing the natural behavior of birds or fish to find food into the flocks or schools. Originally, since this is a kind of swarm

intelligence techniques, each feasible solution is supposed as a particle. Each particle has a specific position and velocity and moves (flies) through the search space according to its best last experimented position and the best known position of the other members. This is similar to behavior of people in making decisions where they consider their own best past experience and the best experience of how the other people around them have performed [33].

4.1. Main Computations Main equations of PSO are formulated as follows:

$$\vec{x}_i'(t) = \vec{x}_i(t-1) + \vec{v}_i \quad (18)$$

$$\vec{v}_i(t) = W\vec{v}_i(t-1) + c_1r_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + c_2r_2(\vec{x}_{leader} - \vec{x}_i(t)) \quad (19)$$

Equation 18 calculates the position of each particles at time t of algorithm's running, where \vec{x}_i and \vec{v}_i represent the position and velocity of particle i , respectively. The parameter is determined by Equation 19 where W is the inertia weight for controlling particle's velocity through the search space during the run and guiding the swarm into the more appropriate areas. c_1 and c_2 reflex the preference of particles to move following the swarm's leader or their best experienced position in the past. That is noticeable based on the kind of search, global or local (using different

Being

```

Initialize swarm
Locate leader
g=0
While g < gmax
For each particle
Update Position (Flight)
Evaluation
Update pbest
End For
Update leader
g++
End While

End

```

Figure 1. Pseudo code of PSO algorithm.

neighborhood's topologies) \vec{x}_{leader} called \vec{x}_{gbest} or \vec{x}_{lbest} , respectively.

Whereas the PSO firstly had been developed for continuous problems, its discrete binary version also was presented in order to make it more efficient and applicable for real word problems [34].

Figure 1 shows a general procedure of PSO. At first step, a primarily swarm is initialized as same as generating a random population in general EA procedures. Then, the best particle is selected as leader of the swarm according to the position (fitness value) of all particles. In the next step, the position and velocity of each particle are updated via the mentioned equations in order to fly the swarm through the search space. Searching procedure is continued until the termination criterion (e.g., specific number of iterations) is met.

4.2. Multi-Objective PSO To extend single objective algorithms to the multi-objective form, the main procedure of solving has to be modified. Coello Coello [35] first presented the approach of solving MOPs with the PSO algorithm that uses an external archive of particles that is later used by

Being

```

Initializing random swarm
  Initializing of leaders in an external archive
  (Repository)
  Quality (leaders)
Locate leader
g = 0
While g < gmax
For each particle
  Select leader using a chosen criterion
  Update Position (Flight)
  Mutation
  Evaluation
  Update pbest
End For
  Update the content of external archive Quality
  (leaders)g++
End While
  Report results in the external archive as final
  output of algorithm
End

```

Figure 2. Pseudo code of MOPSO algorithm.

other particles to guide their own flight. Following, the main issues of extending PSO to its multi-objective form, called MOPSO, are reviewed [36].

How are the non-dominated particles selected from the swarm to be used as the leader of other particles?

1. How to retain the components of external archive (non-dominated solutions) found during the search process in order to report solutions that are non-dominated with respect to all the past populations (not only in compare of the current one)?
2. How to avoid premature convergence via maintaining diversity in the swarm?

Figure 2 demonstrates the general procedure of MOPSO that mainly is constructed based on PSO. The differences have marked by italic font. In this algorithm after initializing the first swarm, a set of non-dominated particles is chosen and stored in a repository called external archive. That is because in the MOPSO solving procedure, each particle may have a set of different leaders that means each component of the external archive is known as a candidate of being leader of other particles. The archive is updated during the run and in anticipation its members adapted in the last iteration, are usually reported as the output of the algorithm.

Various methods and criteria have been

TABLE 1. Comparison of the Proposed MOPSO with NSGAII in Terms of the Quality Measure.

| Pro. No. | Problem Definition | | | MOPSO | | Quality Measure % | | Preferred Algorithm |
|----------|--------------------|---------|-------|---------|---------|-------------------|---------|---------------------|
| | Part | Machine | Labor | c_1 | c_2 | MOPSO | NSGA-II | |
| 1 | 10 | 6 | 10 | 1.49618 | 1.49618 | 57 | 43 | MOPSO |
| | | | | 1.25 | 2.5 | 65 | 35 | MOPSO* |
| 2 | 10 | 8 | 10 | 1.49618 | 1.49618 | 0 | 100 | NSGAII |
| | | | | 1.25 | 2.5 | 63 | 37 | MOPSO* |
| 3 | 15 | 10 | 15 | 1.49618 | 1.49618 | 71 | 29 | MOPSO |
| | | | | 1.25 | 2.5 | 92 | 8 | MOPSO* |
| 4 | 15 | 15 | 15 | 1.49618 | 1.49618 | 60 | 40 | MOPSO |
| | | | | 1.25 | 2.5 | 71 | 29 | MOPSO* |
| 5 | 20 | 15 | 15 | 1.49618 | 1.49618 | 56 | 44 | MOPSO |
| | | | | 1.25 | 2.5 | 73 | 27 | MOPSO* |
| 6 | 20 | 15 | 20 | 1.49618 | 1.49618 | 77 | 23 | MOPSO* |
| | | | | 1.25 | 2.5 | 25 | 75 | NSGAII |
| 7 | 25 | 15 | 20 | 1.49618 | 1.49618 | 70 | 30 | MOPSO* |
| | | | | 1.25 | 2.5 | 63 | 37 | MOPSO |
| 8 | 25 | 20 | 20 | 1.49618 | 1.49618 | 33 | 67 | NSGAII |
| | | | | 1.25 | 2.5 | 55 | 45 | MOPSO* |
| 9 | 30 | 20 | 20 | 1.49618 | 1.49618 | 50 | 50 | - |
| | | | | 1.25 | 2.5 | 50 | 50 | - |
| 10 | 35 | 25 | 25 | 1.49618 | 1.49618 | 88 | 12 | MOPSO |
| | | | | 1.25 | 2.5 | 70 | 30 | MOPSO* |

TABLE 2. Comparison of the Proposed MOPSO with NSGAI in Terms of the Spacing Measure.

| Pro. No. | Problem Definition | | | MOPSO | | Spacing Measure | | Preferred Algorithm |
|----------|--------------------|---------|-------|---------|---------|-----------------|----------|---------------------|
| | Part | Machine | Labor | C1 | C2 | MOPSO | NSGA-II | |
| 1 | 10 | 6 | 10 | 1.49618 | 1.49618 | 2.3717 | 0 | NSGAI |
| | | | | 1.25 | 2.5 | 6.7727 | | NSGAI |
| 2 | 10 | 8 | 10 | 1.49618 | 1.49618 | 1.6514 | 3.1623 | MOPSO |
| | | | | 1.25 | 2.5 | 1.8257 | | MOPSO |
| 3 | 15 | 10 | 15 | 1.49618 | 1.49618 | 5.4772 | 0 | NSGAI |
| | | | | 1.25 | 2.5 | 1.3693 | | NSGAI |
| 4 | 15 | 15 | 15 | 1.49618 | 1.49618 | 3.6894 | 1361 | MOPSO |
| | | | | 1.25 | 2.5 | 3.5277 | | MOPSO |
| 5 | 20 | 15 | 15 | 1.49618 | 1.49618 | 4.6645 | 6.1645 | MOPSO |
| | | | | 1.25 | 2.5 | 376.0516 | | NSGAI |
| 6 | 20 | 15 | 20 | 1.49618 | 1.49618 | 0 | 7.1371 | MOPSO |
| | | | | 1.25 | 2.5 | 8.2664 | | MOPSO |
| 7 | 25 | 15 | 20 | 1.49618 | 1.49618 | 3.873 | 54.8443 | MOPSO |
| | | | | 1.25 | 2.5 | 11.225 | | MOPSO |
| 8 | 25 | 20 | 20 | 1.49618 | 1.49618 | 0 | 28.694 | MOPSO |
| | | | | 1.25 | 2.5 | 9.5277 | | MOPSO |
| 9 | 30 | 20 | 20 | 1.49618 | 1.49618 | 0 | 631.1991 | MOPSO |
| | | | | 1.25 | 2.5 | 857.6597 | | NSGAI |
| 10 | 35 | 25 | 25 | 1.49618 | 1.49618 | 1.9365 | 346.4703 | MOPSO |
| | | | | 1.25 | 2.5 | 6.1073 | | MOPSO |

introduced for selecting leaders and controlling the size of external archive in evolutionary multi-objective methods and specially MOPSO algorithm [36,37]. Since the proposed approach of Coello Coello [37] generally covers all of the above-mentioned issues, we use this approach to code the MOPSO algorithm in order to solve the proposed bi-objective model.

5. COMPUTATIONAL RESULTS

A desired number of test problems are designed in order to solve the presented bi-objective model. To emphasize the effectiveness of our proposed MOPSO with respect to NSGA II, three criteria are

used to compare these algorithms as follows: 1) quality measure, 2) spacing measure, and 3) diversity measure that are introduced by Coello Coello and Lamont [38]. As shown in Tables 1 and 2, the computational results obviously confirm that our proposed MOPSO are more confident and effective than NSGAI considering quality and spacing criteria.

Two approaches have been considered in the proposed MOPSO. In the first one (i.e., $C_1=C_2$) [39], transporting of particles based on their best last positions has the same preferences with regards to the best position in the whole swarm. While in the second approach (i.e., $C_1=1.25$ and $C_2=2.5$), particles tend to move following leaders than own best experiences.

Considering the quality measure shown in

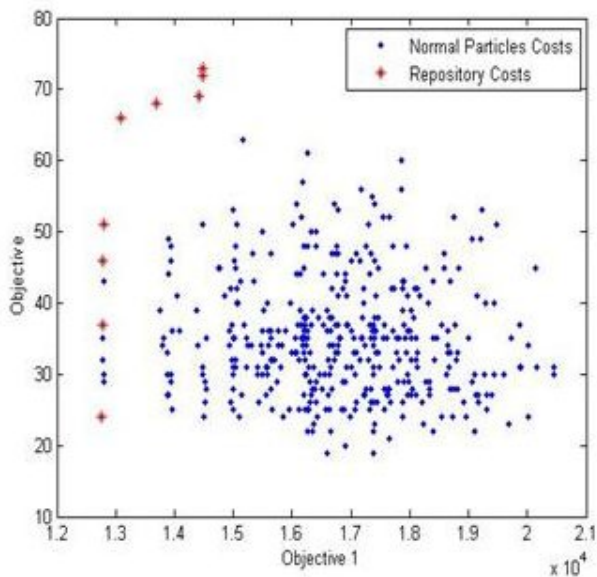


Figure 3. Conflict between objectives for problem 9.

Table 1, it is clear that in most of test problems, the second approach gives better results with respect to the first one. The sign "*" in this table shows the better approach in problems that both are won NSGAI. Figure 3 also demonstrates the conflict of two independent objective functions of the presented model that generally proves considering of the cell formation problem in a multi-objective form. Red marked solutions are the members of the Pareto optimal set as the desired output of the algorithm.

6. CONCLUSION

In this paper we have proposed a bi-objective model for forming manufacturing cells which is the first and most important step of designing cellular manufacturing system. The model contains two conflicting objectives that try to optimize labor assignment and maximize cell utilization respectively. Population based nature of swarm intelligence methods, make them more robust and efficient than classical methods like goal programming. As a result, a multi-objective particle swarm optimization (MOPSO) was applied to solve the proposed model. Computational results of recommended solving procedure are compared with the results obtained by a well-known evolutionary procedure called NSGA-II, in order to verify its effectiveness.

7. REFERENCES

1. Joines, J.A., Culbreth C.T., King R.E., "A comprehensive review of production oriented manufacturing cell formation technique", *Technical Report, NCSU-IE Technical Report*. Raleigh, NC: North Carolina State University, (1996).
2. Selim, H.M., Askin, R.G., Vakharia, A.J., "Cell formation in group technology: review, evaluation and directions for future research", *Computers and Industrial Engineering*, Vol. 34, (1996), 3-20.
3. Onwunbolu, C., Mutingi, M., "A Genetic Algorithm Approach to Cellular Manufacturing Systems", *Computers and Industrial Engineering*, Vol. 39, (2001), 135-144.
4. Gungor, Z., Arıkan, F., "A Parametric Model for Cell Formation and Exceptional Elements' Problems with Fuzzy Parameters", *Journal of Intelligent Manufacturing*, Vol. 16, (2005), 103-114.
5. Rogers, D.F., Kulkarni, S.S., "Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing", *European Journal of Operational Research*, Vol. 160, (2005), 423-444
6. Peker, A., Kara, Y., "Parameter setting of the Fuzzy ART neural network to part-machine cell formation problem", *International Journal of Production Research*, Vol. 42, No. 6, (2004), 1257-1278.
7. Foulds, L.R., Wilson, J.M., "The sustainable cell formation problem: manufacturing cell creation with machine modification costs", *Computers and Operations Research*, Vol. 33, (2006), 1010-1032.
8. Yang, M.S., Hung, W.L. and Cheng, F.C., "Mixed-variable fuzzy clustering approach to part family and machine cell formation for GT applications", *International Journal of Production Economic*, Vol. 103, (2006), 185-198.
9. James, T.L. and Brown, E.C., "A hybrid grouping genetic algorithm for the cell formation problem", *Computers & Operations Research*, Vol. 34, No. 7, (2007), 2059-2079.
10. Ozdemir, R.G., Gencyilmaz, G. and Aktin, T., "The modified fuzzy art and a two-stage clustering approach to cell design", *Information Sciences*, Vol. 177, (2007), 5219-5236.
11. Mahdavi, I., Paydar, M.M., Solimanpur, M. and Heidarzade, A., "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing", *Expert Systems with Applications*, Vol. 36, No. 3, (2009), 6598-6604.
12. Tariq, A., Hussain, I. and Ghaffoor, A., "A hybrid genetic algorithm for machine-part grouping", *Computers and Industrial Engineering*, Vol. 56, No. 1, (2009), 347-356.
13. Andres, C. and Lozano, S., "A particle swarm optimization algorithm for part-machine grouping", *Robotics and Computer-Integrated Manufacturing*, Vol. 22, (2006), 468-474.
14. Tavakkoli-Moghadam, R., Rabbani, M. and Godratnama, A., "An efficient algorithm to allocate parts to cells maximizing cell efficiency", *International Journal of Engineering*, Vol. 17, No. 4, (2004), 381 -

15. Kim, C.O., Baek, J.G. and Baek, J.K., "A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences", *International Journal of Production Research*, Vol. 42, No. 18, (2004), 3911–3927.
16. Lei, D. and Wu, Z., "Tabu Search-Based Approach to Multi-Objective Machine-Part Cell Formation", *International Journal of Production Research*, Vol. 43, No. 24, (2005), 5241–5252.
17. Tavakkoli-Moghaddam, R., Minaeian, S. and Rabbani, M., "A New Multi-Objective Model for Dynamic Cell Formation Problem with Fuzzy Parameters" *International Journal of Engineering*, Vol. 21, No. 2, (2008), 159-172.
18. Dimopoulos, C., "Multi-objective optimization of manufacturing cell design", *International Journal of Production Research*, Vol. 44, No. 22, (2006), 4855–4875.
19. Wu, X., Chu, C.H., Wang, Y. and Yan, W., "A genetic algorithm for cellular manufacturing design and layout", *European Journal of Operational Research*, Vol. 181, (2007), 156–167.
20. Bajestani, M.A. and Rabbani, M., "A multi-objective scatter search for a dynamic cell formation problem", *Computers and Operations Research*, Vol. 36, (2009), 777 – 794.
21. Tavakkoli-Moghaddam, R., Ranjbar-Bourani, M., Mirzahosseini, H. and Aryanezhad, M.B., "A New Multi-Objective Model for A Cell Formation Problem Considering Machine Utilization and Alternative Process Routes", *International Journal of Engineering*, Vol. 22, No. 1, (2009), 21-32.
22. Cesani, V.I. and Steudel, H.J., "A Study of Labor Assignment Flexibility in Cellular Manufacturing Systems", *Computers and Industrial Engineering*, Vol. 48, (2005), 571–591.
23. Suresh, N.C. and Slomp, J., "A Multi-Objective Procedure for Labor Assignments and Grouping in Capacitated Cell Formation Problems", *International Journal of Production Research*, Vol. 39, No.18, (2001), 4103–4131.
24. Suer, G.A. and Cosner, J., "Models for Cell Loading And Product Sequencing in Labor-Intensive Cells", *Computers and Industrial Engineering*, Vol. 56, No.1, (2009), 97-105.
25. Bidanda, B., Ariyawongrat, P., Needy, K.L., Norman, B. and Tharmmaphornphilas, W., "Human related issues in manufacturing cell design, implementation, and operation: a review and survey", *Computers and Industrial Engineering*, Vol. 48, (2005), 507–523.
26. Slomp, J., Chowdary, B.V. and Suresh, N.C., "Design of virtual manufacturing cells: a mathematical programming approach", *Robotics and Computer-Integrated Manufacturing*, Vol. 21, (2005), 273–288.
27. Slomp, J., Bokhorst, J.A.C. and Molleman, E., "Cross-training in a cellular manufacturing environment", *Computers and Industrial Engineering*, Vol. 48, (2005), 609–624.
28. Suer, G.A., Arıkan, F. and Babayigit, C., "Effects of Different Fuzzy Operators On Fuzzy Bi-Objective Cell Loading Problem in Labor-Intensive Manufacturing Cells", *Computers and Industrial Engineering*, Vol. 56, (2009), 476–488.
29. Satoglu, S.I. and Suresh, N.C., "A Goal Programming Approach for Design of Hybrid Cellular manufacturing Systems in Dual Resource Constrained Environments", Vol. 56, No. 2, (2009), 560-575.
30. Abraham, A., Jain, L. and Goldberg, R., "Evolutionary Multi-objective Optimization: Theoretical Advances and Applications", Springer.
31. Deb, K., "Multi-Objective Optimization using Evolutionary Algorithms", John Wiley and Sons, (2003).
32. Kennedy J. and Eberhart, R.C., "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks*, (1995), 1942–1948.
33. Kennedy, J., Eberhart, R.C. and Shi, Y., "Swarm intelligence", San Francisco, CA: Morgan Kaufmann, (2001).
34. Kennedy, J. and Eberhart, R.C., "A Discrete Binary Version of the Particle Swarm Algorithm", In Proceedings of the *IEEE International Conference on Neural Networks*, (1997), 4104–4109.
35. Coello Coello, C.A. and Lechuga, M.S., "MOPSO: A proposal for multiple objective particle swarm optimization", *IEEE*, (2002), 1051–1056.
36. Reyes-Sierra, M. and Coello Coello, C.A., "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art", *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, (2006), 287–308.
37. Coello Coello, C.A. and Lechuga, M.S., "Handling Multiple Objectives with Particle Swarm Optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, (2004), 256 – 279.
38. Coello Coello, C.A. and Lamont, G.B., "Evolutionary Algorithms for Solving Multi-Objective Problems", Springer, Second Edition, (2007).
39. Poli, R., Kennedy, J. and Blackwell, T., "Particle Swarm Optimization: An Overview", *Swarm Intelligence*, Vol. 1, (2007), 33–57.