# Multi Objective Particle Swarm Optimization for A Dynamic Cell Formation Problem

Fatemeh Sarayloo[1], Reza Tavakkoli-Moghaddam [2]

***Abstract*— Cellular manufacturing system, an application of group technology, has been considered as an effective way to obtain productivity in a factory. A new multi objective dynamic cell formation with production planning consideration is presented in this paper, where total workload variations, inter-intra cellular movements and the sum of costs consisting machine costs, production planning costs, reconfiguration costs, are to be minimized. Because this type of problem is Np-hard, a multi objective particle swarm optimization is applied to achieve locally Pareto-optimal frontier. Multi objective particle swarm optimization (MOPSO) is compared with a multi objective genetic algorithm, i.e. NSGAII, based on some comparison metrics to show its efficiency. The computational results depict the superiority of MOPSO compared to NSGAII.**

***Index Terms*—Cellular Manufacturing System, Multi Objective, Particle Swarm Optimization, NSGAII.**

## I. Introduction

Group technology (GT) is a manufacturing philosophy which identifies and assigns the parts into part families and the machines into cells by taking advantage of part similarity in processing and design functions. One specific application of GT is cellular manufacturing (CM) which strives to make the small-to-medium-sized batches of a large variety of part types produced in the flow shop manner [1][2][3]. The major benefits of CM have been reported in the literature as simplification and reduction in material handling, decreasing the work-in-process inventories, reduction in set-up time, increment in flexibility, better production control and shorter lead time [4][5]. However cell formation is the first step in designing, other aspects such as production planning is important to be considered.

Conflicting objectives and dynamic environment make it difficult to solve an integrated problem. There are lots of multi objective models to solve a cell formation problem. However multi objective approaches are more useful and valuable compared to several previous models that have been presented by single objective.

## II. Multi Objective Optimization

A general multi objective minimization problem is described as follows:

$$\min\{ f_1(x), f_2(x),..., f_n(x)\}$$
$$s.t. \quad g(x) \le 0$$

[1]Department of Industrial Engineering, Graduate School Islamic Azad University-South Tehran Branch, Tehran, Iran, f.saraylou@yahoo.com
[2] Department of Industrial Engineering and Department of Engineering Science, College of Engineering, University of Tehran, tavakoli@ut.ac.ir

Where, $x = (x_1, x_2,..., x_n)$ is a vector of decision variable.

$f_1(x), f_2(x),..., f_m(x)$ is the vector of solution in the objective space and $g(x)$ is constraint vector. In order to describe the basic concepts of multi objective optimization, some definitions are to be reviewed [6].

**Definition 1.** Given two vectors $x, y \in \Re^k$, we say that $x \le y$ If $x_i \le y_i$ for i=1,…, k, and that $x$ *dominates* $y$ (Denoted by $x < y$) if $x \le y$.

Fig.1 shows a particular case of the *dominance relation* in the presence of two objective functions.

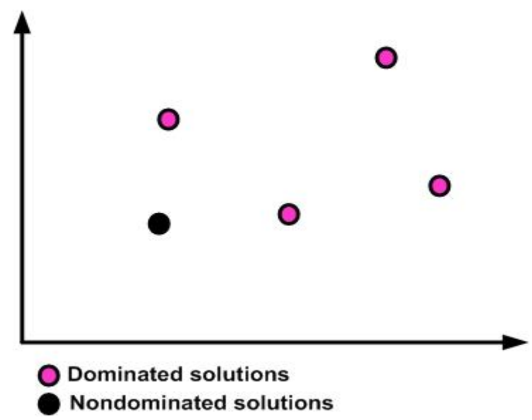


Fig.1: Dominance relation in Bi-Objective space.

**Definition 2.** We say that a vector of decision variables $x \in X \subset \Re^n$ is *nondominated* with respect to $X$, if there is no existence of another $x' \in X$ such that $f(x') < f(x)$.

**Definition 3**. We say that a vector of decision variables $x^* \in F \subset \Re^n$ ($F$ is the feasible region) is Pareto-optimal if it is nondominated with respect to $F$.

**Definition 4.** The *Pareto Optimal Set* $P^*$ is defined by:
$P^* = \{x \in F \mid x \text{ is Pareto-optimal}\}$

**Definition 5.** The Pareto Front $PF^*$ is defined by:
$PF^* = \{ f(x) \in \Re^k \mid x \in P^* \}$

Fig. 2 shows a particular case of the Pareto front for the two objective functions.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

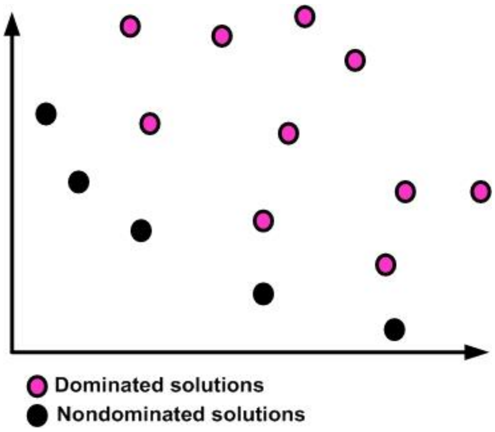


Fig.2: The pareto front of a set of solutions in Bi-Objective space.

## III. PROBLEM FORMULATION

In this section, the integrated multi objective problem is formulated as a nonlinear mixed-integer programming model. The problem is formulated under the following assumptions borrowed from [7]:

*Assumptions:*

(1) Each part type has a number of operations that must be processed as numbered, respectively.
(2) The processing time for all operations of a part type on different machine types are known and deterministic.
(3) The demand for each part type in each period is known and deterministic.
(4) The capabilities and time-capacity of each machine type is known and constant over the planning horizon.
(5) The constant cost of each machine type is known.
(6) The variable cost of each machine type is known. This cost implies operating cost that is dependent on the workload allocated to the machine.
(7) The relocation cost of each machine type from one cell to another between periods is known. All machine types can be moved to any cell. This cost is the sum of uninstalling, shifting and installing costs. The time required for relocation is assumed to be zero.
(8) The maximum number of cells can be formed in each period is specified in advance.
(9) The maximal cell size is known in advance.
(10) All machine types are assumed to be multi-purposed ones. Thus, each machine type can perform one or more operations without incurring a modification cost. Likewise, each operation-part can be performed on different machine types with different processing times.
(11) Holding and backorders inventories are allowed between periods with known costs. Thus, the demand for a part in a given period can be satisfied in the preceding or succeeding periods.

*Indices:*

c index for manufacturing cells, c=(1,…,$C$)
m index for machine types, m=(1,…,$M$)
p index for part types, p=(1,…,$P$)
h index for time periods, h=(1,…,$H$)
j index for operations which belongs to part j=(1,…, $O_p$)

*Input parameters:*

| | |
|---|---|
| $P$ | number of part types |
| $O_p$ | number of operations for part p |
| $M$ | number of machine types |
| $C$ | maximum number of cells that can be formed |
| $D_{ph}$ | demand for part p in period h |
| $\gamma^{\text{inter}}$ | Inter-cell movement cost |
| $\gamma^{\text{intra}}$ | Intra-cell movement cost |
| $\alpha_m$ | constant cost of machine type m in each period |
| $\beta_m$ | variable cost of machine type m for each unit time |
| $\delta_m$ | relocation cost of machine type m |
| $T_m$ | time-capacity of machine type m in each period |
| $UB$ | maximal cell size |
| $t_{jpm}$ | processing timer required to perform operation j of part type p on machine type m |
| $a_{jpm}$ | equals to1,if operation j of part p can be done on machine type m; 0 otherwise |
| $\eta_p$ | inventory carrying cost per unit part p during each period |
| $\rho_p$ | backorder cost per unit part p during each period |
| $l$ | lead time where $l \le H-1$ |

*Decision variables:*

| | |
|---|---|
| $N_{mch}$ | number of machines type m allocated to cell c in period h |
| $x_{jpmch}$ | the portion of operation j of part type p is done on machine type m in cell c in period h |
| $Q_{ph}$ | number of production of part p produced in period h |
| $I_{ph}$ | inventory/backorder level of part p at the end of period h. A negative value of $I_{ph}$ means the backordered level or shortage. |

*Mathematical model*:

By using mentioned notations, the proposed model is written as follows:

$$\min Z_1 = \sum_{h=1}^{H}\sum_{m=1}^{M}\sum_{c=1}^{C} N_{mch}\alpha_m \tag{1}$$

$$+\sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{j=1}^{O_p}\sum_{m=1}^{M} B_m Q_{ph} t_{jpm} x_{jpmch}$$

$$+\frac{1}{2}\sum_{h=1}^{H}\sum_{c=1}^{C}\sum_{m=1}^{M}\delta_m \left|N_{mch}-N_{mch-1}\right|$$

$$+\sum_{h=1}^{H}\sum_{p=1}^{P}(\eta_p I_{hp}^{+}+\rho_p I_{hp}^{-})$$

$$\min Z_2 = \frac{1}{2}\sum_{h=1}^{H}\sum_{p=1}^{P}\sum_{j=1}^{O_p-1}\sum_{c=1}^{C} Q_{ph}\gamma^{\text{inter}} \tag{2}$$

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\times\left|\sum_{m=1}^{M} x_{(j+1)pmch} - \sum_{m=1}^{M} x_{jpmch}\right|$$

$$+\frac{1}{2}\sum_{h=1}^{H}\sum_{p=1}^{P}\sum_{j=1}^{O_p-1}\sum_{c=1}^{C} Q_{ph}\,\gamma^{\text{intra}}$$

$$\times\left(\sum_{m=1}^{M}\left|x_{(j+1)pmch} - x_{jpmch}\right| - \left|\sum_{m=1}^{M} x_{(j+1)pch} - \sum_{m=1}^{M} x_{jpch}\right|\right)$$

$$\min Z_3 = STD(RWT_{mch}) + STD(CWT_{ch}) \qquad (3)$$

$$STD(RWT_{mch}) = \left[\frac{1}{m-1}\sum_{m=1}^{M}(RWT_{mch} - A)^2\right]^{\frac{1}{2}}$$

$$WT_{mch} = \sum_{p=1}^{P}\sum_{j=1}^{J} Q_{ph}.t_{jpm}.x_{jpmch}$$

$$RWT_{mch} = \frac{WT_{mch}}{T_m.N_{mch}}$$

$$A = \frac{\sum_{m=1}^{M} RWT_{mch}.N_{mch}}{\sum_{m=1}^{M} N_{mch}}$$

$$STD(CWT_{ch}) = \left[\frac{1}{c-1}\sum_{c=1}^{C}(CWT_{ch} - A')^2\right]^{\frac{1}{2}}$$

$$CWT_{ch} = \sum_{m=1}^{M} WT_{mch}$$

$$A' = \frac{\sum_{c=1}^{C} CWT_{ch}}{C}$$

s.t.

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{Op} a_{jpm}x_{jpmch} = 1 \qquad \forall h \qquad (4)$$

$$\sum_{p=1}^{P}\sum_{j=1}^{Op} Q_{ph}t_{jpm}x_{jpmch} \le T_m N_{mch} \qquad \forall m,c,h \qquad (5)$$

$$\sum_{m=1}^{M} N_{mch} \le UB \qquad \forall c,h \qquad (6)$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M} N_{mch} \ge 1 \qquad \forall h \qquad (7)$$

$$N_{mc(h-1)} + K^{+}_{mch} + K^{-}_{mch} = N_{mch} \qquad \forall m,c,h \qquad (8)$$

$$I_{ph} = I_{p(h-1)} + Q_{ph} - D_{ph} \qquad \forall p,h \qquad (9)$$

$$I^{+}_{ph} \le I_{ph}, I^{-}_{ph} \ge -I_{ph} \qquad \forall p,h \qquad (10)$$

$$x_{jpmch} \in [0,1], N_{mch}, Q_{ph}, I^{+}_{ph}, I^{-}_{ph}, \ge 0 \ \text{ And integer}$$

$$-\infty \langle I_{ph} \langle \infty \ \text{ and integer}$$

The first objective of model (Eq.1) consists of four terms. The first term is the total sum of constant costs. The second one is the variable costs of machines. The third term presents cell reconfiguration costs which are sum of adding, removing and relocating costs of machines between cells in consecutive periods. Coefficient 1/2 in the third term is embedded because each reconfiguration cost is taken into account twice in calculation. The forth term is the cost of production planning.

The second objective (Eq.2) signifies intra-cellular and inter-cellular movements. Coefficient 1/2 in this objective is embedded because each movement is taken into account twice in calculation.

The last objective consists of sum of standard deviation of intra and inters cellular workload. Some parameters are defined as follows:

| | |
|---|---|
| $WT_{mch}$ | is working time of machine type m in cell c in period h. |
| $RWT_{mch}$ | is working time of each machine of type m in cell c in period h. |
| $A$ | Signifies average workload of each machine. |
| $CWT_{ch}$ | is working time of each cell in each period *h* |
| $A'$ | is average workload of cells |

Equation (4) guarantees that all part-operation is assigned to machines. Equation (5) ensures that machine capacities are not exceeded and must satisfy the demand. Equation (6) guarantees the maximum cell size is not violated. Equation (7) guarantees that the number of all machines in one period is not 0. Equation (8) is called a balance constraint ensuring that the number of machines in the current period is equal to the number of machines in the previous period, plus the number of machines being moved in, and minus the number of machines being moved out. Equation (9, 10) indicates the balance inventory constraint between periods for each part type at each period. It means that the inventory level of each part at the end of each period is equal to the inventory level of the part at the end of the previous period plus the quantity of production minus the part demand rate in the current period.

It is worth mentioning that the first objective is in direct contradictory with the second and third objective. The number of machines should be decreased to optimize the first objective which result in increasing the second and third objective

We propose separated objectives for movements and workload unbalancing because the scaling of these objectives is considerably different.

## IV. MULTI OBJECTIVE SWARM OPTIMIZATION ALGORITHM

Cell formation problem with considering production planning is defined as an NP-hard problem. Meta heuristic and evolutionary algorithms are useful methods to solve NP-hard problems. Particle swarm optimization (PSO) is classified into population based method, like Genetic Algorithm. PSO can be a useful method in multi objective optimization because of its reasonable run time and effectiveness.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

### A. Solution Coding

Our proposed solution coding is presented in two steps:

1. Raw solutions:

The raw solutions which consider no constraints are produced. Each solution composes of three components:

*Sol.N*: A hyper matrix $[N]_{mch}$, signifying the number of machine type $m$ in cell $c$ in period $h$. The elements of $[N]$ are random integer values within $[0, UB]$.

*Sol.q*: $[q]_{jpmc\ h}$ is defined as the portion of $j$th operation of part $p$ which is processed on machine $m$ in cell $c$ in period $h$. the elements of [q] are real random values within [0,1].

*Sol.r*: The hyper matrix $[r]_{ph}$ signifies the production of part $p$ to be produced in period $h$. The elements of $[r]$ are random values within $[0, r_{max}]$. $r_{mch}$ is a maximum possible production of part $p$ in period $h$ ,say 3 in this problem.

2. Processed solutions

Raw solutions are converted to processed solutions which satisfy all of our constraints. *Sol.N* is converted to *Sol2.N* to satisfy constraints (6,7 and 8). We convert *Sol.q* with regarding to constraint (4) and $a_{jpm}$.

### B. The main steps of MOPSO [7 ]:

1) Initialize the population:
(a) FOR i=1 TO number of particles
(b) Initialize
2) The speed of each particle should be initialized.
For i=0 to max number of particles
VEL[i]=0
3) All of particles in population are evaluated by objectives.
4) Position of particles representing non dominated vector should be stored in repository [REP].
5) Hyper cube of search space which is explored so far is generated and each of the particles in REP is located in these hyper cubes.
6) The memory of each particle should be initialized.
(a) For i=0 to max
PBEST[i]=POP[i]
7) WHILE maximum number of cycles has not been reached DO

*a*. Compute the speed of each particle (shown in Fig.3) using Eq.11 :

$$VEL[i]=W\times VEL[i]+ R_1 \times (PBESTS[i]-POP[i])+ R_2\times (REP[h]-POP[i]) \quad (11)$$

Where W (inertia weight) takes a value of 0.7; $R_1$ and $R_2$ are random numbers in the range [0 1]; PBEST [i] is the best position that the particle i has had; REP [h] is a value that is taken from the repository; the index h is selected in the following way: those hyper cubes containing more than one particle are assigned a fitness equal to the result of dividing any number x > 1 (we used x = 10 in our experiments) by the number of particles that they contain. This aims to decrease the fitness of those hyper cubes that contain more particles and it can be seen as a form of fitness sharing. Then, we apply roulette-wheel selection using these fitness values to select the hypercube from which we will take the corresponding particle. Once the hypercube has been selected, we select randomly a particle within such hypercube. POP [i] is the current value of the particle i .

*b*. Compute the new positions of the particles adding the speed produced from the previous step

POP[i]=POP[i]+VEL[i].

*c*. Maintain the particles within the search space incase they go beyond their boundaries (avoid generating solutions that do not lie on valid search space). When a decision variable goes beyond its boundaries, then we do two things: 1) the decision variable takes the value of its corresponding boundary (either the lower or the upper boundary) and 2) its velocity is multiplied by (-1) so that it searches in the opposite direction.
*d*. Evaluate each of the particles in POP.
*e*. Update the contents of REP together with the geographical representation of the particles within the hyper cubes. This update consists of inserting all the currently no dominated locations into the repository. Any dominated locations from the repository are eliminated in the process. Since the size of the repository is limited, whenever it gets full, we apply secondary criterion for retention: those particles located in less populated areas of objective space are given priority over those lying in highly populated regions.
*f*. When the current position of the particle is better than the position contained in its memory, the particle's position is updated using PBESTS[i]=POP[i].
The criterion to decide what position from memory should be retained is simply to apply Pareto dominance (i.e., if the current position is dominated by the position in memory, then the position in memory is kept; otherwise, the current position replaces the one in memory; if neither of them is dominated by the other, then we select one of them randomly).
*g*. Increment the loop counter.
8) END WHILE.

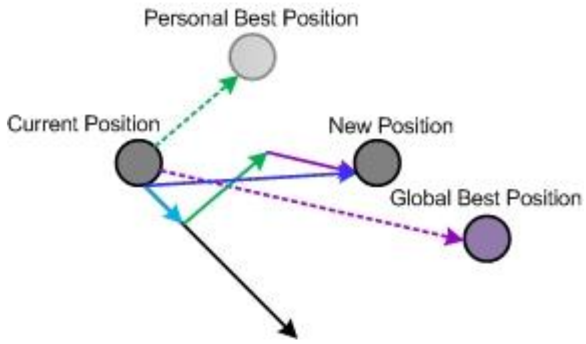

Fig.3: Velocity in particle swarm optimization.

## IV. EXPERIMENTAL RESULTS

The performance of MOPSO is compared with NSGAII, another well-known evolutionary algorithm, in some test

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

problems generated randomly.

### A. Algorithm assumptions:

General assumption: number of population=20, max iteration =30
MOPSO: W=0.7, R1=1.5, R2=1.5
NSGAII: the crowded comparison-operator is used for selection. Operator simulated binary cross over (SBX) and polynomial mutation are used as genetic operators.

### B. Comparison metrics:

1. The number of non dominated solutions: this metrics compare the number of non dominated solution produced in each Pareto front of two algorithms.
   It is obvious, the more nondominated solutions, the better pareto front.
2. Quality metric:
   Reference [8] presented only one relative metric. The definition of this metric is given in Eq.12. Let two sets of $F'$ and $F''$ be given. The following function C transforms the two sets $F'$ and $F''$ into a real value included in the interval [0, 1]:

$$C(F', F'') = \frac{card(\{x'' \in F'', \exists x' \in F' | x' < x''\})}{card(F'')} \quad (12)$$

   Where card $F''$ corresponds to the number of elements inside the set $F''$, and $x' \leq x''$ means that the vector $x'$ dominates the vector $x''$.
   So this metric allows us to compute what portion of the surface $F''$ is dominated by the tradeoff surface $F'$.
3. Diversity metric:
   The Diversity metric [9] is calculated in Eq.13.

$$Diversity\ Metric = \sqrt{\sum_{i=1}^{m} max\{\|\acute{a}_i - \acute{b}_i\|;\ \acute{a}, \acute{b} \epsilon \acute{X}\}} \quad (13)$$

   For better understanding we use the relative metric calculated in Eq.14, Eq.15:

$$Relative\ Diversity(MOPSO) = \frac{diversity\ (MOPSO)}{diversity\ (MOPSO) + diversity\ (NSGAII)} \quad (14)$$

$$Relative\ Diversity(NSGAII) = \frac{diversity(NSGAII)}{diversity(MOPSO) + diversity(NSGAII)} \quad (15)$$

4. Time:
   Run time is another important metric which affects the performance of algorithm significantly. By having the same computer for both simulations we present time metric evaluation as follows (Eq.16):

$$Time\ Metric = \frac{runtime\ (MOPSO)}{runtime\ (NSGAII)} \quad (16)$$

### C. Computational Results

Five test problems generated randomly (Table.1 ) and the comparison between two algorithms is shown in Table.2 to Table.6. MOPSO could achieve solutions with high quality in comparison with NSGAII based on Table.2 to Table.6. MOPSO is superior to NSGAII regarding to quality metric, number of non dominated solutions and time metric, however, MOPSO provides lower diversity metric. Based on obtained results, Generally MOPSO has a better performance in comparison with NSGAII, shown in Table.2 to Table.6.

Table.1: Test problems.

| No. test problem | $p(\sum Op) \times M \times H$ |
|---|---|
| 1 | $5(11) \times 5 \times 3$ |
| 2 | $7(12) \times 6 \times 2$ |
| 3 | $8(18) \times 7 \times 2$ |
| 4 | $9(27) \times 8 \times 2$ |
| 5 | $10(23) \times 9 \times 2$ |

Table.2: Comparison metrics of Test Problem 1.

| Test Problem 1 | | Count | Diversity | Quality | Time |
|---|---|---|---|---|---|
| MOPSO | Best | 50 | 0.4 | 1 | 0.7 |
| | Worst | 49 | 0.31 | 0.88 | 0.6 |
| | Ave. | 49.78 | 0.32 | 0.93 | 0.66 |
| | Std. | 0.02 | 0.07 | 0.02 | 0.04 |
| NSGAII | Best | 28 | 0.69 | 0.4 | 0.6 |
| | Worst | 20 | 0.6 | 0.3 | 0.7 |
| | Ave. | 23 | 0.66 | 0.34 | 0.65 |
| | Std. | 0.05 | 0.04 | 0.03 | 0.03 |

Table.3: Comparison metrics of Test Problem 2.

| Test Problem 2 | | Count | Diversity | Quality | Time |
|---|---|---|---|---|---|
| MOPSO | Best | 50 | 0.42 | 1 | 0.72 |
| | Worst | 48 | 0.32 | 0.9 | 0.62 |
| | Ave. | 49.6 | 0.31 | 0.95 | 0.65 |
| | Std. | 0.025 | 0.08 | 0.03 | 0.03 |
| NSGAII | Best | 29 | 0.69 | 0.42 | 0.61 |
| | Worst | 20 | 0.61 | 0.32 | 0.71 |
| | Ave. | 24 | 0.65 | 0.33 | 0.66 |
| | Std. | 0.05 | 0.04 | 0.05 | 0.04 |

Table.4: Comparison metrics of Test Problem 3.

| Test Problem 3 | | Count | Diversity | Quality | Time |
|---|---|---|---|---|---|
| MOPSO | Best | 49 | 0.39 | 1 | 0.71 |
| | Worst | 48 | 0.32 | 0.96 | 0.63 |
| | Ave. | 48.91 | 0.34 | 0.98 | 0.65 |
| | Std. | 0.03 | 0.06 | 0.01 | 0.04 |
| NSGAII | Best | 30 | 0.7 | 0.4 | 0.63 |
| | Worst | 20 | 0.6 | 0.32 | 0.73 |
| | Ave. | 25 | 0.63 | 0.33 | 0.65 |
| | Std. | 0.07 | 0.03 | 0.03 | 03 |

Table.5: Comparison metrics of Test Problem 4.

| Test Problem 4 | | Count | Diversity | Quality | Time |
|---|---|---|---|---|---|
| MOPSO | Best | 50 | 0.41 | 1 | 0.73 |
| | Worst | 49 | 0.31 | 0.95 | 0.66 |
| | Ave. | 49.1 | 0.34 | 0.97 | 0.68 |
| | Std. | 0.02 | 0.05 | 0.01 | 0.03 |
| NSGAII | Best | 30 | 0.7 | 0.41 | 0.75 |
| | Worst | 22 | 0.6 | 0.33 | 0.65 |
| | Ave. | 25 | 0.66 | 0.38 | 0.67 |
| | Std. | 0.03 | 0.05 | 0.02 | 0.04 |

Table.6: Comparison metrics of Test Problem 5.

| Test Problem 5 | | Count | Diversity | Quality | Time |
|---|---|---|---|---|---|
| MOPSO | Best | 50 | 0.48 | 1 | 0.72 |
| | Worst | 48 | 0.34 | 0.94 | 0.69 |
| | Ave. | 48.7 | 0.4 | 0.95 | 0.7 |
| | Std. | 0.04 | 0.06 | 0.02 | 0.01 |
| NSGAII | Best | 30 | 0.71 | 0.43 | 0.73 |
| | Worst | 20 | 0.65 | 0.35 | 0.65 |
| | Ave. | 27 | 0.66 | 0.37 | 0.68 |
| | Std. | 0.06 | 0.05 | 0.05 | 0.03 |

## REFERENCES

[1] J. McAuley, “Machine grouping for efficient production”, production engineer,1972,pp.53-57

[2] JR. King, “Machine component group formation in group technology”, international journal of management science, 1979.

[3] I. Ham, K. Hitomi, T. Yoshida, “Group technology application to production management”, Nijhoff, Dordrecht: Kluwer, 1985.

[4] RG. Asking, S. Estrada, “Investigation of cellular manufacturing practices”, hand book of cellular manufacturing systems, New York: John Wiley, 1999, pp. 25-34.[Chapter 1]

[5] M.Aramoon bajestani, M.Rabbani, A.R Rahimi –vahed, G. Baharian khoshkhou, “ A multi objective scatter search for a dynamic cell formation problem ”, Computers and Operation Research, vol.36, pp.777-794,2009.

[6] N. Safaei, R. Tavakkoli-Moghaddam, “Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems” Int. J. Production Economics, 2009, pp.301-314.

[7] C. Coello coello, “Handling multiple objective with particle swarm optimization,” IEEE Trans. On evolutionary computation, vol.8, NO.3, Jun.2004.

[8] E. Zitzler, “evolutionary algorithms for multi objective optimization: methods and applications, doctoral dissertation ETH 13398, Swiss Federal institute of technology, 1999.

[9] E.zitzler, K.deb, L.thiele, “comparison of multiobjective evolutionary algorithms: empirical results”, Evolutionary computation, 1999, pp.173-195.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)