Developing a Multi-objective Mathematical Model for Dynamic Cellular Manufacturing Systems

Mohammad Saidi-Mehrabad[a], Seyedeh Maryam Mirnezami-ziabari[b,*]

[a]*Professor, Department of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran*

[b]*MSc, Faculty of Industrial Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Received 3 October, 2010; Revised 25 November, 2010; Accepted 27 December, 2010

Abstract

This paper is in search of designing the cellular manufacturing systems (CMSs) under dynamic and flexible environment. CM is proper for small-to-medium lot production environment that helps the companies to produce variable kind of productions with at least scraps. The most important benefits of CM are decline in material handling, reduction in work-in-process, reduction in set-up time, increment in flexibility, improved quality, and shorter lead time. In this research A multi-objective mixed integer model is presented that considers some real-world critical conditions same as costs of multi-period cell formation and production planning , human resource assignment to cells and balancing workload of cells. This model groups the parts and machines concurrently with labor assignment This study aims to 1) minimize various costs including reassignment cost of human resource, the batch inter-cell material handling cost, constant and variable cost of machines, relocation and purchase cost of machines, 2) minimize cell load variation and 3) maximize utilization rate of human resource. The model is complicate, so it is verified with Lingo 8. 0. Soft ware. Since particle swarm optimization approach less than many other metaheuristic approaches have been applied to solve multi-objective CMS problems so far, we utilize this method to solve our model. The results are presented at the last part.
Keywords: Cellular manufacturing systems; Dynamic and flexible environment; Multi objective; Mixed-integer.

1. Introduction

Group technology (GT) is a manufacturing philosophy which groups the parts into part families according to their similarity in processing and/or design functions. Cellular Manufacturing (CM) is one specific application of GT that allocates the machines into machine-cells according to the parts that should be manufactured by them. CM is proper for small-to-medium lot production environment that helps the companies to produce variable kind of productions with at least scraps. The most important benefits of CM are decline in material handling, reduction in work-in-process, reduction in set-up time, increment in flexibility, improved quality, and shorter lead time. The design of a cellular manufacturing system (CMS) consists of four stages: (1) cell formation (CF) – grouping parts with similar geometric design features or processing requirements into part families to take advantage of their similarities for manufacturing purpose, and collecting required machines into machine cells, (2) group layout – laying out machines within each cell (intra-cell layout) and cells with respect to one another (inter-cell layout), (3) group scheduling – scheduling various parts and part families per period for production, and (4) resource allocation – transmission various resources like tools, manpower and materials capitals in to manufacturing system (Wemmerlov and Hyer [16]). Although all of these four stages are important for designing a CMS, the first stage plays a critical role to flourish system. Lately, with considering the concept of dynamic cellular manufacturing system (DCMS), it is easier to overcome disadvantages of traditional CMS and reach to real word conditions. In dynamic cellular manufacturing systems suppose that product mix and part demands are variable such as seasonal products demands, for the total planning horizon. A schema of DCMS contains machine relocation for two following periods is shown in Fig. 1.

*Corresponding author E-mail: mirnezami181z@yahoo.com

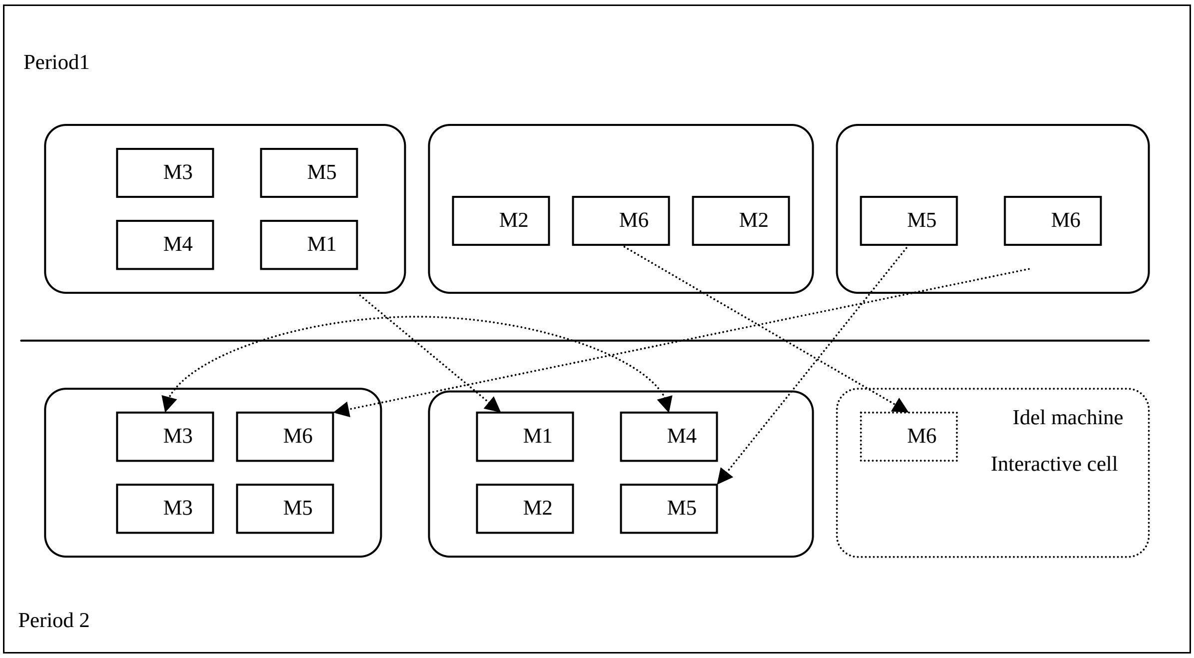

Fig. 1. A schema for machine relocation and changing the number of cells in a DCMS

2. Literature Review

In this section, Firstly, a brief description of the works which have been done to address dynamic issue in CMS will be provided, and then the multi-objective approaches in CMS will be temporarily argued.

2.1. Dynamic Issues in Cellular Manufacturing Systems

Alternative CF methods such as similarity coefficient algorithm, ROC (rank order clustering) algorithms and direct clustering algorithm have been proposed in literature. Example of these methods was presented by Singh [12]. Extensive classifications of studies relate to part-machine grouping problems have been presented by Kusiak [6], Mansouri et al. [8] and Selim et al. [11]. Defersha and Chen [3] presented a comprehensive mathematical model and a two-phase genetic-algorithm-based heuristic to generate manufacturing cells over multiple time periods. The model attempted to minimize machine investment cost, inter-cell material handling cost, operating cost, subcontracting cost, tool consumption cost, set-up cost and system reconfiguration cost in an integrated manner. The proposed genetic algorithm included a number of problem-specific genetic operators, heuristic and two searching phase to form independent cells in the first phase. Safaei et al.[10] applied a hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. They developed a mixed-integer programming model to design cellular manufacturing systems (CMSs) under dynamic environment. Mahdavi et al. [7] presented an integer mathematical programming model for the design of cellular manufacturing systems in a dynamic environment. The aim of the proposed model was to minimize holding and backorder costs, inter-cell material handling cost, machine and reconfiguration costs and hiring, firing and salary costs.

2.2. Multi-objective Approaches in Cellular Manufacturing Systems

Gupta et al. [5] proposed a genetic algorithm based solution approach to address the machine cell-part grouping problem. Three different objective functions considered are (1) minimize total moves (inter-cell as well as intra-cell moves), (2) minimize cell load variation, and (3) minimize both the above objective functions simultaneously. Venugopal and Narendran [15] proposed a bi-criteria mathematical model with a solution procedure based on a genetic algorithm. Trials on a sample problem suggested that the proposed algorithm can be a powerful tool that can be gainfully employed in a cellular manufacturing environment. The algorithm is inherently parallel and is capable of super linear speed-up in multi-processor systems. Solimanpur et al. [13] have presented a multi-objective integer programming for the design of cellular manufacturing systems with independent cells. A genetic algorithm with multiple fitness functions is proposed to solve the formulated problem. The proposed algorithm finds multiple solutions along the Pareto optimal frontier. Sujono and Lashkari [14] proposed a method for simultaneously determining the operation allocation and material handling system selection in an FMS environment with multiple performance objectives. The 0–1 integer programming model is developed to select machines, assign operations of part types to the selected machines, and assign material handling equipment to transport the parts from machine to machine, as well as to handle the part at a given machine. Mansouri [9] formulated a multi-objective optimization problem (MOP) to simultaneously take into account optimization of four conflicting objectives regarding: intercellular movements, cost, utilization, and workload balance. Due to the complexity of the developed MOP, neither exact optimization techniques nor total enumeration are applicable for large problems. For this, a multi-objective genetic algorithm (MOGA) solution approach is proposed. Aramoon Bajestani et al.[1] have presented a multi-objective dynamic cell formation problem, where total cell load variation and sum of miscellaneous costs (machine cost, inter-cell material handling cost, and machine relocation cost) are minimized simultaneously. They designed a new multi-objective scatter search (MOSS) for finding locally Pareto-optimal frontier. Ghotboddini et al. [4] have presented a multi-objective mixed integer model for DCMS. Their model solves the part and machine grouping simultaneously with labor assignment to minimize the cost of various terms like reassignment cost of human resource, over time cost of equipments and labors, and maximize utilization rate of human resource. They used the Benders' decomposition method this to solve Their model.

3. Problem Description

This section describes a nonlinear mixed integer cell formation model in a dynamic environment in presence of machine flexibility, alternative process plans for part types, and the possibility machine-relocation. First object is to extend the objective function that Safaei et al. [10] have presented in their DCMS model, which minimized the function of divers cost variables (machine fixed and variable cost, inter-cell and intra-cell material handling cost, machine relocation cost) .in addition to this terms, we consider purchase cost, labor inter-cell movement cost without considering intra-cell material handling cost. Also we consider labor assignment ratio to each cell in each period. This objective function tries to maximize the total labor assignment ratio for all cells in each period that considered by Ghotboddini et al[4]. Furthermore in this paper cell load variation concept is considered.

3.1. Assumptions

1. There are several operations for each part type which must be processed according to their numbers.
2. Process time for all operations of a part type on various machine types are known and constant.
3. Part types' demands in each period are known and constant.
4. Purchasing price is known and constant over the planning horizon.
5. Fixed cost of each machine type is known and deterministic. It covered overall service and maintenance cost.
6. Variable cost of each machine type is known and constant. It covered operating cost depending on the workload assigned to the machine.
7. There is not inventory and back order. All demands must be complete during certain period.
8. All machine types are multipurpose. Then there is no modification cost for performing one or more operations.
9. Total number of labors is fixed for all periods. Firing and hiring are not allowed.
10. Inter-cell relocation cost of each machine between periods is known and constant.
11. Labor inter-cell transferring cost is known and deterministic for each period.
12. Intra-cell movement cost of machines and labors is not considered.
13. Machine capacity of each machine type is known and deterministic.

3.2. Indices

p Index for part types (p= 1,. . . , P)

j Index for operations belong to part p (j=1,…, J_p)

m Index for machine types (m= 1,. . . , M)

c Index for manufacturing cells (c= 1,. . . , C)

t Index for time periods (t= 1,. . . , T)

3.3. Input Parameters

B_p Batch size for inter-cell movements of part p

η_p Inter-cell material handling cost per batch of part p

π_m Constant cost of machine type m in each period

λ_m Variable cost of machine type m for each unit time

h_{jpm} Processing time required to perform operation j of part type p on machine type m

δ_{jpm} =1 if operation j of part p can be done on machine type m ; 0 otherwise

r_m Relocation cost of machine type m

γ_m Time-capacity of machine type m in each period

UB Upper bound for cell size cell size

φ_m Purchase cost of machine type m

$D_p(t)$ Demand for part p in period t

L Total number of labors

$\beta(t)$ Constant cost of inter-cell labor moving in period t

q_{jpm} Manual workload time required to perform operation j of part type p on machine type m

a_{jpm} =1 if operation j of part p can be done on machine type m ; 0 otherwise

3.4. Decision Variables

$N_{mc}(t)$ Number of machines type m allocated to cell c in period t

$K^{+}_{mc}(t)$ Number of machines type m added in cell c in period t

$K^{-}_{mc}(t)$ Number of machines type m removed from cell c in period t

$I^{+}_{mc}(t)$ Number of machines type m purchased in period t

$I^{-}_{mc}(t)$ Number of machines type m sold in period t

$y_{jpmc}(t)$=1 if operation j of part type p is done on machine type m in cell c in period t; 0 otherwise

$w_{jpmc}(t)$ Workload on machine m by part p in cell c during period t

$U_{jpc}(t)$ Average intra-cell processing time for part p in cell c during period t

$L_{c}(t)$ Number of labors assigned for cell c in period t

$L^{+}_{c}(t)$ Number of labors added in cell c in period t

$L^{-}_{c}(t)$ Number of labors removed in cell c in period t

0(t) Minimum labor ratio for period t

3.5. Mathematical Model

$$minZ_1 = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{c=1}^{C} N_{mc}(t)\pi_m + \sum_{t=1}^{T}\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{j=1}^{J_p} D_p(t).y_{jpmc}(t)h_{jpm}.\lambda_m + \frac{1}{2}\sum_{t=1}^{T}\sum_{c=1}^{C}\sum_{m=1}^{M}\left(K^{+}_{mc}(t)+K^{-}_{mc}(t)\right).r_m + \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{c=1}^{C}\varphi_m I^{+}_{mc}(t) + \frac{1}{2}\sum_{t=1}^{T}\sum_{p=1}^{P}\left\lceil\frac{D_p(t)}{B_p}\right\rceil.\eta_p \times \sum_{j=1}^{Jp-1}\sum_{c=1}^{C}\left|\sum_{m=1}^{M} y_{(j+1)pmc}(t) - \sum_{m=1}^{M} y_{jpmc}(t)\right| + \frac{1}{2}\sum_{t=1}^{T}\sum_{c=1}^{C}\beta(t)\left(L^{+}_{c}(t)+L^{-}_{c}(t)\right) \quad (1)$$

$$\min Z_2 = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{p=1}^{P}(w_{pmc}(t) - U_{pmc}(t))^2 \quad (2)$$

$$\max Z_3 = \sum_{t=1}^{T} O(t) \quad (3)$$

$$\sum_{c=1}^{C}\sum_{m=1}^{M}\delta_{jpm}.y_{jpmc}(t) = a_p(t) \qquad \forall j,p,t \quad (4)$$

$$\sum_{p=1}^{P}\sum_{j=1}^{J_p} D_p(t).y_{jpmc}(t).h_{jpm} \le N_{mc}(t).\gamma_m \qquad \forall m,c,t \quad (5)$$

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

$$N_{mc}(t-1) + I^{+}_{mc}(t) - I^{-}_{mc}(t) = N_{mc}(t) \qquad \forall m,c \quad \forall t>1 \quad (7)$$

$$L_c(t) = L_c(t-1) - L^{+}_{c}(t) - L^{-}_{c}(t) \qquad \forall c,t \quad \forall t>1 \quad (8)$$

$$\sum_{c=1}^{C} L_c(t) = L \qquad \forall t \quad (9)$$

$$L_c(t)\left|\sum_{m=1}^{M}\sum_{j=1}^{J_p}\sum_{p=1}^{P} D_p(t).q_{jpm}.y_{jpmc}(t) \ge O(t) \qquad \forall c,t \quad (10)\right.$$

$$w_{pmc}(t) = \sum_{j=1}^{j_p}\left(h_{jpm}.D_p.y_{jpmc}(t)\right)/\gamma_m \qquad \forall p,m,c,t \quad (11)$$

$$U_{pmc}(t) = \left(\sum_{m=1}^{M} W_{pmc}(t).N_{mc}(t)\right)\bigg/\left(\sum_{m=1}^{M} N_{mc}(t)\right) \qquad \forall p,c,t \quad (12)$$

$$\sum_{c=1}^{C} N_{mc}(t) - \sum_{c=1}^{C} N_{mc}(t-1) = I^{+}_{m}(t) - I^{-}_{m}(t) \qquad \forall m, \forall t>1 \quad (13)$$

$$y_{jpmc}(t)\epsilon\{0,1\} \qquad \forall j,p,m,c,t \quad (14)$$

$$N_{mc}(t), I^{+}_{mc}(t), I^{-}_{mc}(t), L_c(t), L^{+}_{c}(t), L^{-}_{c}(t), K^{+}_{mc}(t), K^{-}_{mc}(t) \ge 0 \ \ \& \ Integer \quad (15)$$

$$O(t) \ge 0 \qquad \forall t \quad (16)$$

The first objective function given in Eq. (1) is to minimize the sum of various costs. The first term indicates sum of constant cost of all machines like maintenance costs which we use over the planning horizon for entire cells. The second term represents the variable cost of machines in whole cells and periods. This cost is incurred by the time-workload allocated to each machine type during each period. The third term represents machines relocation costs. The forth term points to the total purchasing cost for entire machines during all periods. The fifth term represents inter-cell material handling costs. The last term of this equation considers the total cost of labor's inter-cell transference over the planning horizon. Coefficient ½ in the third, fifth and sixth term is inserted, because each relocation in the model has taken into account two times.

The second objective function given in Eq. (2) is to minimize the sum of machines workload variation squares for entire machines during all periods. It causes balancing of cells workload.

The third objective function given in Eq. (3) is to maximize the sum of minimum labor ratio for entire periods. This ratio is defined in Eq. (10) for each period. When the third objective is maximized, the whole labors' ideal time is minimized and we could utilize manpower proportion in each cell for each period. In this situation assignment of labors to each cell is more efficient over the planning horizon.

The first constraint is presented in Eq. (4) ensures that each operation is assigned only to one machine and one cell. Eq. (5) guarantees that machines capacity are not exceeded. Eq. (6) specifies the upper bounds for cell size.

Eq. (7) indicates that the number of machines type in the current period in a specific cell equals to the number of the same machine type in the previous period, plus the number of machines moved into and minus the number of machines eliminated from that cell. This restriction is called balance constraint. Eq. (8) indicates that the number of labors in the current period in a specific cell equals to the number of the labors in the previous period, plus the number of labored moved into and minus the number of labors eliminated from that cell. Eq. (9) indicates that number of labors is assigned to each cell at each period equals to the total number labors. Eq. (10) defines the minimum labor ratio for period t. The right-hand side of constraint is a fraction which its numerator is number of labors in cell c for period t and its denominator is total required manual workload time for each cell. This fraction defines for entire cells in a period, and then all of them compare with o as an index. Eq. (11) represents the workload on machine m in cell c during period t. Eq. (12) defines the average intra-cell processing time for part p in cell *c* during period t. Eq. (13) shows the number of each machine type we buy or sell during each period. Eq. (14) represents that variable y equals to 0 or 1. Eq. (15) is the integer constraint. Eq. (16) defines that labor ratio O(t) has a positive amount.

4. A Numerical Example

The aim of this section is to verify the applicability of the proposed model by a hypothetical example with randomly generated data. This example is solved by branch and bound (B&B) method under Lingo 8.0 software on an Intel(R), core (TM) i5, 3.23 GHz lap top with 512 Mb RAM. For solving the model with Lingo software, we use of Standardized method for converting multi-objective optimization problem into a single objective optimization problem. First of all in this approach, each objective function has to be solved independently, considering all related constrains. Then all the objective functions' optimal values are considered to create a new standard objective function. The standardized objective function is defined as (17):

$$Min f(x)=\frac{1}{f_1^*(x)}*f_1(x)+\frac{1}{f_2^*(x)}*f_2(x)-\frac{1}{f_3^*(x)}*f_3(x) \qquad (17)$$

Where $f_i(x)$ is the ith objective function and $f_i^*(x)$ is optimal value of ith objective function. Since the model is NP-Hard problem and linearization with exact methods is not possible, therefore we solved that as nonlinear model with branch & bound algorithm under lingo software. The random example is generated according to the information provided in Table 1.

Table 1
Test problem generation

parameter	value	parameter	value	parameter	value
L	$U(20,40)$	δ_{jpm}	$\sum_{j=1}^{m}\delta_{jpm} \sim u(2,4)$	λ_m	$U(1,10)$
M	(p/2)+2	h_{jpm}	$u(0.1,1)$	γ_m	$U(400,600)$
UB	$\lceil 1.5\sqrt{p}+1, 2\sqrt{p}+1 \rceil$	$D_p(t)$	U(1000,100)	r_m	$\pi_m \times U(.4,.6)$
B_p	$U(10,50)$	φ_m	U(10000,20000)	$\beta(t)$	100
η_p	$U(45,55)$	π_m	$U\ \varphi_m \times (0.06,0.08)$	q_{jpm}	$\frac{h_{jpm}}{u(8,12)}$

In table 1, term ''U'' implicates to the uniform distribution. The data set related to the considered example is shown in Table 2. The considered example consists of three part types, four machine types, twenty labors and three periods in which each part type is assumed to have three operation must be processed respectively. Table 2 includes the machines and parts information like variable, constant and relocation cost of machines, quantity of demand and batch sizes for each part type. Runtime is limited to one hour. Thus, the best solution for each objective function found after one hour (Zi best) and z normal are reported in Table 3. Also costs that introduced in the model are shown in this table. The cell configurations for three periods corresponding to the best obtained solution are shown in Fig. 2. This figure shows some of the characteristics and advantages of the proposed model. The sequence of operations can be recognized by the numbers shown in the column associated with each part type. For example, the first and second operations of part 1 in period 1 must be performed on machines 3 and 4 respectively in cell 1, but its third operation must be performed on machine 1 in cell 1. Thus, the process plan allocated to part 1 needs one intra-cell movements and one inter-cell movement. The number of inter cell movements in 3 periods is same as to each other, but the difference is in their locations, also in the number of intra-cell movements. For example in period 1 part 3 has two intra-cell movements, but it has one intra-cell movement in period 2. In addition inter-cell movements in periods1, 2 must be performed for part1and part 2, while in period 3 inter-cell movements are related to part1 and part 3.

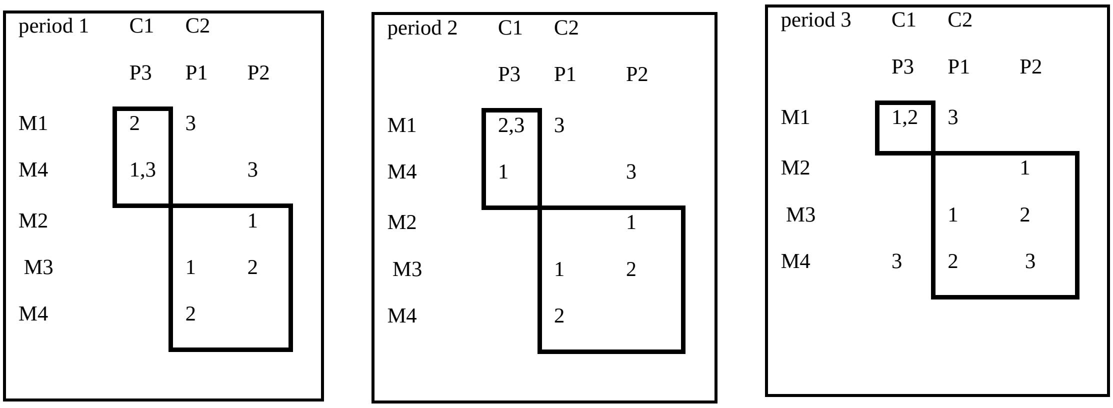

Fig.2. Best obtained cell configurations for typical test problem presented in Table 2.

Table 2
Typical test problem

Machine info.						P1			p2			p3		
γ_m (hour)	π_m($)	λ_m ($)	r_m ($)	φ_m ($)		p11	p12	p13	p21	p22	p23	p31	p32	p33
473	720	6	393	12000	M1	0.31	0.72	0.25	0.14	0.8	0	0.45	0.15	0.57
540	862	3	500	11000	M2	0	0	0	0.18	0.28	0.92	0	0	0
454	1020	4	462	17000	M3	0.29	0	0.78	0	0.19	0.78	0	0	0
500	950	6	385	15000	M4	0	0	0	0	0	0.34	0.45	0.45	0.47
$D_p(t)$					period1	900			745			780		
					period2	125			863			575		
					period3	200			350			379		
B_p						34			37			10		
η						48			46			49		
L:20														
UB :4														
$\beta(t)$:100														
C:2														

Table 3
Best obtained solution in details

Z1best	Z2best	Z3best	Amount of Z normal	CPU time	
43136.11	6.6499	0.1807	1.12944	3600	
Machine constant cost	Machine variable cost	Machine Reconfiguration cost	Machine Purchasing cost	Inter-cell movements cost	labor transfer cost
15126.04	27492.57	517.5000	0	0	0

Table 4
Obtained solution by MOPSO

Z1	Z2	Z3	CPU time(s)
47159	6.97	0.1976	76.87

Table 5
Solving some problems with small, medium and large sizes with MOPSO algorithm

NO.	Size of problem M-P-O-C-T-L	z1	z2	z3	CPU time(s)	Percent of feasible solutions
1	4-4-[3,5]-2-2-20	82153	5.7443	0.3083	83.16	100%
2	5-6-[3,5]-3-2-20	175842	12.46468	0.19806	135.12	80%
3	6-8- [3,5]-3-2-20	156558	9.1522	0.15072	148.18	90%
4	7-10- [3,5]-3-2-20	181780	19.58294	0.07678	181.92	88%
5	8-10- [3,5]-3-2-20	245342	33.4754	0.04816	207.64	60%
6	9-14- [3,5]-3-2-22	411860	44.9739	0.0404	235.64	40%
7	10-16- [3,5]-3-2-20	397040	52.2477	0.0274	278.20	60%
8	10-16- [3,5]-3-2-22	308940	53.4087	0.0487	250.24	80%
9	11-18- [3,5]-3-2-22	390310	52.848	0.037	281.07	50%
10	12-20- [3,5]-3-3-24	528960	93.6942	0.0333	435.34	60%
11	14-24- [3,5]-3-3-26	886730	147.449	0.0266	597.06	40%
12	16-28- [3,5]-4-3-28	1162500	196.2805	0.0269	833.51	40%
13	17-30- [3,5]-4-3-30	988200	140.0854	0.033	895.34	80%
14	17-30- [3,5]-5-3-30	1001400	145.2228	0.0374	1066.59	40%
15	17-30- [3,5]-5-3-40	1332000	159.3593	0.0508	1143.13	30%

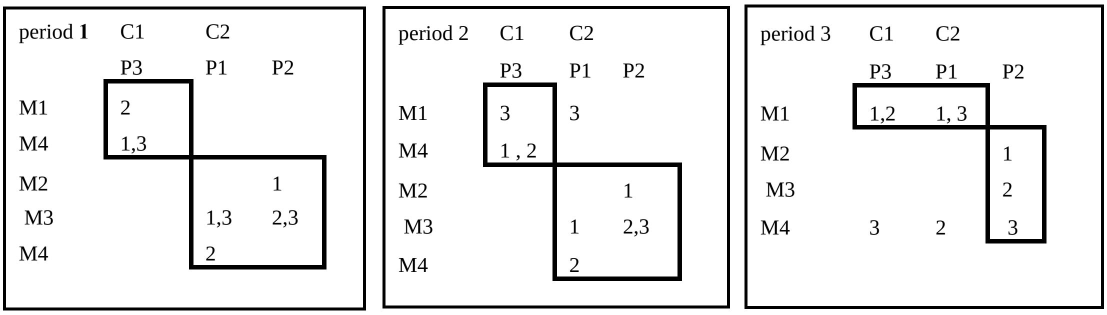

Fig.3. Final result of solving model by MOPSO algorithm.

5. Solution Algorithm

5.1. Particle Swarm Optimization Approach

Cell formation problems with their combinatorial nature are usually known to be NP-hard and so finding an optimal even a feasible solution in a reasonable time is difficult. Therefore in lots of studies heuristic or Meta heuristic algorithms are used to solve cell formation models. Between Meta heuristic algorithms PSO is a useful method because of its reasonable run time and effectiveness.

5.2. The main steps of MOPSO [2]

1. Initialize the population POP:
   a) FOR $i = 0 \quad to \quad number\ of\ \ particles$
   b) Initialize POP[i]
2. Initialize the speed of each particle:
   a) FOR $i = 0 \quad to \quad number\ of\ \ particles$
   b) $VEL[i] = 0$
3. Evaluate each of particles in POP.
4. Store the positions of the particles that represent nondominated vectors in the repository REP .
5. Generate hypercubes of the search space explored so far, and locate the particles using this hypercubes as a coordinate system where each particles coordinates are defined according to the values of its objective functions.
6. Initialize the memory of each particle (this memory serves as a guide to travel through the search space. This memory is also stored in the repository):
   a) FOR $i = 0 \quad to \quad number\ of\ \ particles$
   b) PBEST[i]=POP[i]

7. While maximum number of cycles has not been reached do:
   a) Compute the spead of each particle using the following expression:

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

R_1, R_2 are random numbers; 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 hypercubes containing more than one particle are assigned a fitness equal to the result of dividing any number $x \succ 1$ by the number of particles that they contain. This aims to decrease the fitness of those hypercubes that contain more paerticles 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 in case they go beyond their boundaries. When a desition variable goes beyonds its boundaries, then we do two things: (1) the desition variable takes the value of its corresponding boundry (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 hypercubes. This update consists of inserting all the currently nondominated locations into the repository. Any dominated locations from the repository are eliminated in the process. Since the size of repository is limited, whenever it gets full, we apply a 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:

   PBEST[i]=POP[i]

The criterion to decide what position from memory should be retained is simply to apply pareto dominance.

g) Increment the loop counter

8. End while.

6. Experimental Result

In order to evaluate the effectiveness of Particle swarm optimization algorithm for solving a DCMS problem, numerical example in section 4 was tested again. General assumption are as follow: number of population=100, max iteration = 100, W=1, $R_1 = R_2$ =2. Final result after 100 iteration is shown in fig.3 and table 4. Also in table 5 some problems with small, medium and large sizes are produced randomly and solved with MOPSO algorithm under MATLAB software. In this table results including the amount of each of objective functions, CPU time and percent of feasible solutions have been reported. These results are the average of 10 times run. In all of this problems, Number of operations has been considered in the range of [3,5].

7. Conclusions

This paper proposes a multi-objective mathematical model to solve the problems of DCMS. This model is a non-linear mathematical model that minimizes cell-formation costs like constant and variable costs of machine, inter - cell material handling costs by assuming the operation sequence, machine relocation costs. Also the model minimizes cell load variation and maximizes utilization rate of human resource. The model is verified with one small-size test problem that solved by Lingo software and MOPSO algorithm and it is implemented for small, medium and large size test problems under MOPSO algorithm. We suggest utilizing of heuristic or other meta-heuristic approaches to solve the proposed model for real-sized problems and compare their results with the results of MOPSO. Furthermore, considering some variables such as demand variables in an uncertain environment and solving the mode with suitable methods such as fuzzy goal programming is useful.

8. References

[1] M. Aramoon Bajestani, M. Rabbani, A. R. Rahimi-Vahed, G. Baharian Khoshkhou, A multi-objective scatter search for a dynamic cell formation problem, Computers & Operations Research, 36, 777–7940, 2009.

[2] C. Coello, Handling multiple objectives with particle swarms optimization, IEEE Trans. On evolutionary computation, vol.8, NO.3, Jun. 2004.

[3] F. M. Defersha, M. Chen, A comprehensive mathematical model for the design of cellular manufacturing systems, International Journal of Production Economics, 103, 767–783, 2006.

[4] M. M. Ghotboddini, M. Rabbani, H. Rahimian, A comprehensive dynamic cell formation design: Benders' decomposition approach, Expert Systems with Applications, 38, 2478–2488, 2011.

[5] Y. Gupta, M. Gupta, A. Kumar, C. Sundaram, A genetic algorithm-based approach to cell composition and layout

design problems, International Journal of Production Research, 34, 447–82, 1996.
[6] A. Kusiak, The generalized group technology concept, International Journal of Production Research, 25, 561–569, 1987.
[7] I. Mahdavi , A. Aalaei, M. M. Paydar, M. Solimanpur, Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment, Computers and Mathematics with Applications, 60, 1014-1025, 2010.
[8] S. A. Mansouri, S. M. Husseini, S. T. Newman, A review of the modern approaches to multi-criteria cell design. International Journal of Production Research, 38, 1201–1218, 2000.
[9] A. Mansouri, Elimination of exceptional elements in cellular manufacturing systems using multi-objective genetic algorithms, Applications of Multi-Objective Evolutionary Algorithms, World Scientific Publishing Co, 505-527, 2010.
[10] N. Safaei, M. Saidi-Mehrabad, M. S. Jabal-Ameli, A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, European Journal of Operational Research, 185, 563–592, 2008.
[11] H. M. Selim, R. G. Askin, A. J. Vakharia, Cell formation in group technology: Review, evaluation and direction for future research. Computers & Industrial Engineering, 34, 2–30, 1998.
[12] N. Singh, Design of cellular manufacturing systems-an invited review. European Journal of Operational Research, 69(3), 284–291, 1993.
[13] M. Solimanpur, P. Vrat, R. Shankar, A multi-objective genetic algorithm approach to the design of cellular manufacturing systems, International Journal of Production Research, 42, 7, 1419 – 1441, 2004.
[14] S. Sujono, R. S. Lashkari, A multi-objective model of operation allocation and material handling system selection in FMS design, International Journal of Production Economics, 105, 1, 116–133, 2007.
[15] V. Venugopal, T. T. Narendran, A genetic algorithm approach to the machine-component grouping problem with multiple objectives, Computer & Industrial Engineering,22(4),469-480, 2003.
[16] U. Wemmerlov, N. L.Hyer, Procedures for the part family/machine group identification problem in cellular manufacture, Journal of Operations Management, 6, 125–147, 1986.