Application of genetic algorithm for designing cellular manufacturing system incrementally

J. REZAEIAN[a], N. JAVADIAN[a], R. TAVAKKOLI-MOGHADDAM[b],
[a,b]Department of Industrial Engineering
[a]Mazandaran University of Science and Technology, [b]University of Tehran
[a,b]P.O.Box 734, Babol
[a,b]Iran
j_rezaeian@ustmb.ac.ir, n.javadian@ustmb.ac.ir, tavakoli@ut.ac.ir

Abstract: One important issue regarding the implementation of cellular manufacturing systems relates to deciding whether to convert an existing job shop into a cellular manufacturing system comprehensively in a single go, or in stages incrementally by forming cells one after the other taking the advantage of the experiences of implementation. In this paper two heuristic methods based on multi-stage programming and genetic algorithm are proposed for incremental cell formation. The results show that the multi-stage programming solves small problems faster than exact algorithms such as branch and bound. A heuristic procedure based on genetic algorithm is developed on the multi-stage programming to test larger problem sizes.

Key-Words: Incremental cell formation; Cellular manufacturing system; Multi-Stage programming; Genetic algorithm; Job shop; Comprehensive cell formation

1 Introduction

Global competition impels industries produce goods with low cost, high quality and just in time. Flexible manufacturing systems are tools to achieve these criteria.Cellular manufacturing is an application of flexible manufacturing system. It is the result of a direct application of the group technology philosophy. Parts with similar processing requirements such as machines, tools, route and/or geometrical shapes are classified into part families.

Many researchers used of mathematical models to solve their problems. Adam and Mihai [1] in a study mentioned to mathematical methods used in engineering. Cell formation is an area of engineering that considers designing the layout of industries.

The essential problem in designing of a cellular manufacturing system (CMS) is determination of machine-groups and part families popularly known as the machine cell formation (MCF), or also known as machine-component grouping (MCG) problem. Many researches considered the cell formation problem and proposed numerous techniques. Mahesh and Srinivasan [10] clustered a number of techniques and provided an overview of various algorithms that forms cells comprehensively (i.e., non-incrementally) in total. As pointed out by Mahesh and Srinivasan [10], Wemmerlov and Johnson [17] and several others, all the above methods aim at creating a comprehensive CMS in total in a single go. In practice, however, from the viewpoints of planning and implementation and also for capital investment reasons, it would be desirable to move progressively towards conversion of the existing system into cells one after the other.

Adil and Ghosh [2] developed a mathematical model which forms cells incrementally based on greedy random adaptive search procedures.

Balakrishnan and Cheng [3] proposed a model which considers cell formation over a multi-period planning horizon with demand and resource uncertainties. In this study, cell formation has been done non-incrementally where at each period the cell configuration can be changed; however, planning, implementation or capital investment issues have not been addressed. Rezaeian et al. [15] in a study presented a new non-linear model to form cells incrementally and the problem is solved via a genetic algorithm. Many researchers have tried to compare CM, hybrid CM and job shop together ([7], [11], [16], [18]).

In this paper a new nonlinear integer programming model is designed to convert an existing functional layout to a cellular manufacturing system. Cell formation is done incrementally. Two methods based on multi-

stage programming and genetic algorithms (GA) are applied for solving the model.
The rest of the paper is organized as follows. Section 2 introduces the problem; this is done by giving problem description, assumptions, notations and a new mathematical model. Our proposed algorithm based on multi-stage programming approach, genetic algorithm are designed in Section 3, 4 respectively. In Section 5, some experimentations and comparison are shown. Finally, Section 6 presents conclusions.

2 Problem Formulation

2.1 Problem description

We focus on cell formation decisions incrementally. Hence, here a functional layout is considered in the beginning of the planning horizon with the planning horizon being composed of multi periods. *N* parts are considered with each part visiting shops based on its requirements. Generally *M* machines are available in shops. The objective is to decide the number of cells formed in a period, and the assignment of machines to cells such that the total cost is minimized. The total cost consists of intra-cell and inter-cell material handling, intra-shop material handling, inter-shop material handling and material handling between cell and shop costs.

Assumptions

1. The demand for each part type in each period is known.
2. The number of cells formed in each period is limited.
3. Each cell consists of a minimum and maximum number of machines.
4. The unit cost of inter-cell movements, intra-cell movements and movements between cell and shop are known and constant over time.
5. The number of machines available is known and constant over time.

Notations

The following notations are used throughout the paper:

c index for cells
u , *t* indices for periods
m index for machines
p index for parts
s index for shops
α intra-cell material handling cost
β inter-cell material handling cost
γ cost of material handling between cell and shop
ω inter-shop material handling cost
D_{pt} demand for product *p* in period *t*
LB minimum number of machines to be assigned to a cell
UB maximum number of machines to be assigned to a cell
C_{max} maximum number of cells can be formed in a period
M Number of machines
S Number of shops in the beginning of planning horizon
P Number of parts
T Number of periods
$|k_j|$ Number of members of set k_j

$$X_{cu} = \begin{cases} 1 & \text{If cell } c \text{ is formed in period } u \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{mcu} = \begin{cases} 1 & \text{If machine } m \text{ is assigned to cell } c \text{ in period } u \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{pm} = \begin{cases} 1 & \text{If part } p \text{ needs machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$B_{pcu} = \begin{cases} 1 & \text{If part } p \text{ visits cell } c \text{ in period } u \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{pt} = \begin{cases} 1 & \text{If part } p \text{ visits a cell in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$K_{mst} = \begin{cases} 1 & \text{If machine } m \text{ belongs to shop } s \text{ in period } t \\ 0 & \text{Otherwise} \end{cases}$$

$$\lambda_{pt} = \begin{cases} 1 & \text{If part } p \text{ visits a shop in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$\varsigma_{pst} = \begin{cases} 1 & \text{If part } p \text{ visits shop } s \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

2.2 Mathematical model

The objective function and constraints can be formulated as follows:

$$\text{Min } \psi(Y_{mcu}, X_{cu}, B_{pcu}, \delta_{pt}, K_{mst}, \zeta_{pst}, \lambda_{pt})$$

$$= \sum_{t=1}^{T}\sum_{u=1}^{t}\sum_{c=1}^{c_{\max}} \alpha.X_{cu}.\sum_{p=1}^{P} D_{pt}.[\sum_{m=1}^{M}(Y_{mcu}.Z_{pm}) - B_{pcu}]$$
$$+ \sum_{t=1}^{T}\sum_{p=1}^{P} \beta.D_{pt}.\sum_{u=1}^{t}[\sum_{c=1}^{c_{\max}}(X_{cu}.B_{pcu}) - \delta_{pt}] +$$
$$\sum_{t=1}^{T}\sum_{p=1}^{P} \gamma.\lambda_{pt}.\delta_{pt}.D_{pt} +$$
$$\sum_{t=1}^{T}\sum_{p=1}^{P} \omega.D_{pt}[\sum_{s=1}^{S}\zeta_{pst} - \lambda_{pt}], \quad (1)$$

where

$$X_{cu} \geq X_{(c+1)u} \quad (2)$$

$$(1 - B_{pcu}).\sum_{m-1}^{M} Y_{mcu}.Z_{pm} = 0 \quad (3)$$

$$\sum_{m=1}^{M} Y_{mcu}.Z_{pm} \geq B_{pcu} \quad (4)$$

$$(1 - K_{msu}).K_{ms(u+1)} = 0 \quad (5)$$

$$K_{msu} - K_{msu}.K_{ms(u+1)} - \sum_{c=1}^{C_{\max}} Y_{mc(u+1)} = 0 \quad (6)$$

$$(1 - \lambda_{pt}).\sum_{s=1}^{S}\sum_{m=1}^{M} K_{mst}.Z_{pm} = 0 \quad (7)$$

$$\sum_{s=1}^{S}\sum_{m=1}^{M} K_{mst}.Z_{pm} \geq \lambda_{pt} \quad (8)$$

$$(1 - \delta_{pt}).\sum_{c=1}^{C_{\max}} B_{pcu} = 0 \quad (9)$$

$$\sum_{c=1}^{C_{\max}} B_{pcu} \geq \delta_{pt} \quad (10)$$

$$\sum_{c=1}^{C_{\max}} Y_{mcu} \leq 1 \quad (11)$$

$$(1 - X_{cu}).\sum_{m=1}^{M} Y_{mcu} = 0 \quad (12)$$

$$\sum_{m=1}^{M} Y_{mcu} \geq X_{cu} \quad (13)$$

$$(1 - \zeta_{pst}).\sum_{m-1}^{M} K_{mst}.Z_{pm} = 0 \quad (14)$$

$$\sum_{m-1}^{M} K_{mst}.Z_{pm} \geq \zeta_{pst}. \quad (15)$$

The objective function (1) represents the total cost. The total cost consists the costs of intra-cell material handling (first term in *objective function*), inter-cell material handling (second term), material handling between cell and shop (third term) and inter-shop material handling (fourth term). Eq. (2) ensures the order of cell formation in a period. Eqs. (3) and (4) show that part p visits cell c, when at least one of the required machines to process the part is allocated to the cell. Eq. (5) is to ensure that a machine could belong to a shop if it was in that shop in preceding period. Eq. (6) represents that a machine can be allocated only to a cell or a shop in each period. Eqs. (7) and (8) show that part p visits shop s when at least one of the required machines to process the part is allocated to this shop. Eqs. (9) and (10) ensure that a part moves inter-cell if the part visits more than one cell in a period. Eq. (11) ensures that each machine can be allocated to at most one cell in each period. Eqs. (12) and (13) ensure that a cell is formed in a period if at least one machine is allocated to the cell. Eqs. (14) and (15) show that part p visits shop s, when at least one of the required machines to process the part is allocated to the shop.

3 Multi-stage programming

Multi-stage programming is a powerful optimization technique that is particularly applicable to many complex problems requiring a sequence of interrelated decisions. Suarez and Roldan [4] presented a type of multi-stage programming as dynamic programming in Marko decision processes. Here our proposed algorithm is based on a multi-stage approach. Therefore, before applying the algorithm, the number of cells, formed in each period, and the initial location of each machine should be known. We apply a forward recursive approach to solve the problem.

The recursive relation defining the dynamic step is given by the following equation:

$$F_{k=\{k_1,k_2,..k_j\}}(i,j) = \min\left\{\begin{matrix} F_{\{k_1,k_2,...k_{j-h}\}}(i-1, j-h) + \\ \psi(\mathrm{Y_{mci}}, \mathrm{X_{ci}}, \mathrm{B_{pci}}, , \mathrm{K_{msi}}, \lambda_{\mathrm{pi}}) \end{matrix}\right\}, \quad (16)$$

$0 \leq h \leq C_{\max}$,

$i = 1,2,..,T$,

$j = 0,1,...,i*c$,

$LB \leq |k_j| \leq UB$,

$X = (Y_{\mathrm{mcu}}, X_{\mathrm{cu}}, B_{\mathrm{pcu}}, \delta_{pt}, K_{\mathrm{mst}}, \lambda_{pt})$,

where $F_{\{k_1,k_2,...,k_j\}}(i,j)$ means the minimum total cost from period 1 to period i, when j cells are

formed, and k_j shows the machines in cell j. $\psi(X)$ is the value of objective function based on the objective function (1) in current period and X is the vector of update values of decision variables in the period. k = ϕ means that no cells is formed. The optimum solution is achieved as follows:

$$\psi^* = \min \begin{pmatrix} F(T, j) \\ j = 0,1,2,.., \min(T \times C, \frac{M}{LB}) \end{pmatrix} \quad (17)$$

3.1 Proposed algorithm

Here, the steps of algorithm based on multi-stage programming are proposed.

Step 1. Setting the initial value of decision variables

According to the initial layout, in which all machines are assigned to shops before planning, no cells are formed. The decision variable values are set as follow:

$X_{cu} = 0$ *for c = 1,2..,* C_{max} *and u =1,2,..,T*

$Y_{mcu} = 0$ *for m = 1,2,..,M , c = 1,2..,* C_{max} *and u =1,2,..,T*

$B_{pcu} = 0$ *for p = 1,2..,P , c =1,2..,* C_{max} *and u =1,2,..,T*

$\delta_{pt} = 0$ *for p = 1,2..,P , t = 1,2..,T*

$$K_{ms0} = \begin{cases} 1 & \text{If machine } m \text{ belongs to shop } s \text{ in initial layout} \\ 0 & \text{otherwise} \end{cases}$$

$K_{mst} = 0$ *for m =1,2..,m , s =1,2..,S and t =1,2..,T*

$\lambda_{pt} = 0$ *for p = 1,2..,P , t =1,2..,T*

Set *t=1*

Step 2. Combination of machines

Here, all feasible sets of machines are constituted. A set of machines includes at least *LB* and at last *UB* machines and is feasible if the machines, belonging to the set has not been assigned to any cell in previous periods and belongs to the remaining shops.

A machine can be assigned to no cell and remain in the shop and remainder shop is such a cell.

Step 3. Cell formation

Here, a set of machines is assigned to a cell. When the cell is formed in the current period, the related variables will be updated according to the following rules. Each cell contains a feasible set of machines.

Rule 1: In period *t* a machine can be assigned to a cell if it has not been assigned to any cell in this and previous periods. In other worlds the machine should belong to reminder shop before period t.

Rule 2: If machine *m* is assigned to cell *c* in period *t* then the decision variables are set as follow:

$Y_{mct} = 1$ and $Y_{mcu} = 0$ *for u = t+1, t+2, ..., T and c= 1,2,..,* C_{max}.

$K_{mst} = 0$ *for s= 1,2..,S and t= t,t+1,..,T.*

Rule 3: If machine *m* belongs to cell *c* in period *t* ($Y_{mct} = 1$) and part *p* needs machine *m* then $B_{pcu} = 1$ and $\delta_{pt} = 1$.

Rule 4: If machine *m* belongs to shop *s* in period *t* ($K_{mst} = 1$) and part *p* needs machine *m* then $\lambda_{pt} = 1$.

Rule 5: In period *t* a machine cannot be assigned to a cell and a shop at the same time. In other words, in each period the following relation should be satisfied:

$K_{mst} \times Y_{mct} = 0$.

Step 4. Completion a solution, when a solution is completed, all machines in the period are assigned to a cell or remainder shop.

Step 5. Calculate the objective function with update variables.

Step 6. Using the recursive relation (16) and the value obtained in Step 5, use a multi-stage programming to obtain a value for the current solution.

Step 7. For all possible solutions in the current period repeat steps 3-6.

Step 8. Set *t= t+1*

Step 9. If $t \leq T$ then go to step 2 else go to step 10.

Step 10. Determine the best programming.

3.2 Numerical illustration

Here, a numerical example to demonstrate the proposed dynamic programming algorithm is presented. In this example, four machine types (M = 4) are required to process four parts (P = 4) a minimum and maximum of two machines per cell (LB=UB=2), a maximum of two cells each period and three periods in planning horizontal were considered. The machine-part incidence matrix is shown in Table 1. In Table 2 the demands for the parts in the three time periods are given. In the initial layout, machines 1 and 3 belong to shop 1 and machines 2 and 4 belong to shop 2 ($K_{110} = 1$, $K_{310} = 1$, $K_{220} = 1$,

$K_{420}=1$). The unit cost of intra-cell movements and inter-cell movements, inter shop movements, unit cost of intra shop movements and unit cost of movements between cells and shops are considered to be 4,10,12,6,8, respectively. Also maximum number of cells, which can be formed in each period, (C_{max}) is set to 2. The details of the solution process for this example follow next,

Table 1. Machine-part incidence matrix

Part	Machine			
	M_1	M_2	M_3	M_4
P_1	1	0	1	0
P_2	1	0	1	0
P_3	0	1	0	1
P_4	0	1	0	1

Table 2. Demand for parts

Part	Period		
	1	2	3
P_1	10	0	15
P_2	20	0	25
P_3	0	30	5
P_4	0	40	10

$$F_{\phi}(0,0)=0$$

$$F_{\phi}(1,0)=F_{\phi}(0,0)+\psi(k_{111}=1,k_{221}=1,k_{311}=1,k_{421}=1)$$
$$=\{(10+20)*12\}=360$$

$$F_{(1,2)}(1,1)=F_{\phi}(0,0)+\psi(y_{11}=1,y_{21}=1,k_{111}=0,k_{221}=0,$$
$$k_{311}=1,k_{421}=1)=\{(10+20)*4\}=120$$

$$F_{(1,3)}(1,1)=F_{\phi}(0,0)+\psi(y_{11}=1,y_{31}=1,k_{111}=0,k_{221}=1,$$
$$k_{311}=0,k_{421}=1)=\{(10+20)*8\}=240$$

$$F_{(1,4)}(1,1)=F_{\phi}(0,0)+\psi(y_{11}=1,y_{41}=1,k_{111}=0,k_{221}=1,$$
$$=0,k_{421}=0)=\{(10+20)*8\}=240$$

$$F_{(2,3)}(1,1)=F_{\phi}(0,0)+\psi(y_{21}=1,y_{31}=1,k_{111}=1,k_{221}=0,$$
$$=0,k_{421}=1)=\{(10+20)*8\}=240$$

$$F_{(2,4)}(1,1)=F_{\phi}(0,0)+\psi(y_{21}=1,y_{41}=1,k_{111}=1,k_{221}=0,$$
$$=1,k_{421}=0)=\{(10+20)*8\}=240$$

$$F_{(3,4)}(1,1)=F_{\phi}(0,0)+\psi(y_{31}=1,y_{41}=1,k_{111}=1,k_{221}=1,$$
$$=\{(10+20)*6\}=180$$

$$F_{\phi}(2,0)=\{F_{\phi}(1,0)+\psi(k_{111}=1,k_{221}=1,k_{311}=1,k$$
$$=1)\}=\{360+(30+40)*12\}=1200$$

$$F_{(1,2)}(2,1)=\min\{F_{\phi}(1,0)+\psi(y_{12}=1,y_{22}=1,k_{111}$$
$$=0,k_{311}=1,k_{421}=1),F_{(1,2)}(1,1)+\psi(y_{11}=1,y_{21}=$$
$$k_{221}=0,k_{311}=1,k_{421}=1)\}=\min\{360+(30+40)$$
$$120+(30+40)*6\}=540$$

$$F_{(1,3)}(2,1)=\min\{F_{\phi}(1,0)+\psi(y_{12}=1,y_{32}=1,k_{111}=0,k_{221}=1$$
$$,k_{311}=0,k_{421}=1),F_{(1,3)}(1,1)+\psi(y_{11}=1,y_{31}=1,k_{111}=0,k_{221}=1$$
$$,k_{311}=0,k_{421}=1)\}=\min\{360+(30+40)*8,$$
$$240+(30+40)*8\}=800$$

$$F_{(1,4)}(2,1)=\min\{F_{\phi}(1,0)+\psi(y_{12}=1,y_{42}=1,k_{111}=0,k_{221}=1,k_{311}$$
$$=1,k_{421}=0),F_{(1,4)}(1,1)+\psi(y_{11}=1,y_{41}=1,k_{111}=0,k_{221}=1,k_{311}$$
$$=1,k_{421}=0)\}=\min\{360+(30+40)*8,240+(30+40)*8\}=800$$

$$F_{(2,3)}(2,1)=\min\{F_{\phi}(1,0)+\psi(y_{22}=1,y_{32}=1,k_{111}=1,k_{221}=0$$
$$,k_{311}=0,k_{421}=1),F_{(2,3)}(1,1)+\psi(y_{21}=1,y_{31}=1,k_{111}=1,k_{221}=$$
$$0,k_{311}=0,k_{421}=1)\}=\min\{360+(30+40)*8,240+(30+40)*8\}=800$$

$$F_{(2,4)}(2,1)=\min\{F_{0}(1,0)+\psi(y_{22}=1,y_{42}=1,k_{111}=1,k_{221}=0,k_{311}=1$$
$$,k_{421}=0),F_{(2,4)}(1,1)+\psi(y_{21}=1,y_{41}=1,k_{111}=1,k_{221}=0,k_{311}=1,k_{421}$$
$$=0)\}=\min\{360+(30+40)*8,240+(30+40)*8\}=800$$

$$F_{(3,4)}(2,1)=\min\{F_{0}(1,0)+\psi(y_{32}=1,y_{42}=1,k_{111}=1,k_{221}=2,k_{311}=1,$$
$$k_{421}=1),F_{(3,4)}(1,1)+\psi(y_{31}=1,y_{41}=1,k_{111}=1,k_{221}=1,k_{311}=0,k_{421}$$
$$=0)\}=\min\{360+(30+40)*4,180+(30+40)*4\}=460$$

$$F_{\{(1,2)(3,4)\}}(2,2)=\min\{F_{(1,2)}(1,1)+\psi(y_{11}=1,y_{21}=1,$$
$$y_{32}=1,y_{42}=1,k_{111}=0,k_{221}=0,k_{311}=0,k_{421}=0),$$

$$F_{(3,4)}(1,1)+\psi(y_{31}=1,y_{41}=1,y_{12}=1,y_{22}=1,k_{111}=$$
$$0,k_{221}=0,k_{311}=0,k_{421}=0)\}=\min\{120+(30+40)*4$$
$$,180+(30+40)*4\}=400$$

$$F_{\{(1,3)(2,4)\}}(2,2)=\min\{F_{(1,3)}(1,1)+\psi(y_{11}=1,y_{31}=1,$$
$$y_{22}=1,y_{42}=1,k_{111}=0,k_{221}=0,k_{311}=0,k_{421}=0),$$
$$F_{(2,4)}(1,1)+\psi(y_{21}=1,y_{41}=1,y_{12}=1,y_{32}=1,k_{111}=$$
$$0,k_{221}=0,k_{311}=0,k_{421}=0)\}=\min\{240+(30+40)*10$$
$$,240+(30+40)*10\}=940$$

$$F_{\{(1,4)(2,3)\}}(2,2)=\min\{F_{(1,4)}(1,1)+\psi(y_{11}=1,y_{41}=1,y_{22}=1$$
$$,y_{32}=1,k_{111}=0,k_{221}=0,k_{311}=0,k_{421}=0),F_{(2,3)}(1,1)+$$
$$\psi(y_{21}=1,y_{31}=1,y_{12}=1,y_{42}=1,k_{111}=0,k_{221}=0,k_{311}=0,$$
$$k_{421}=0)\}=\min\{240+(30+40)*10,240+(30+40)*10\}=940$$

$F_{\phi}(3,0) = \min\{ F_{\phi}(2,0) + \psi(k_{111}=1, k_{221}=1, k_{311}=1, k_{421}=1)\} = \{1200 + (15+25+5+12)*12\} = 1860$

$F_{(1,2)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{13}=1, y_{23}=1, k_{111}=0, k_{221}=0, k_{311}=1, k_{421}=1), F_{(1,2)}(2,1) + \psi(y_{11}=1, y_{21}=1, k_{111}=0, k_{221}=0, k_{311}=1, k_{421}=1)\} = \min\{ 1200 + (15+25)*4 + (5+10)*6,540 + (15+25)*4 + (5+10)*6\} = 790$

$F_{(1,3)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{13}=1, y_{33}=1, k_{111}=0, k_{221}=1, k_{311}=0, k_{421}=1), F_{(1,3)}(2,1) + \psi(y_{11}=1, y_{31}=1, k_{111}=0, k_{221}=1, k_{311}=0, k_{421}=1)\} = \min\{ 1200 + (15+25+5+10)*8,800 + (15+25+5+10)*8\} = 1220$

$F_{(1,4)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{13}=1, y_{43}=1, k_{111}=0, k_{221}=1, k_{311}=1, k_{421}=0), F_{(1,4)}(2,1) + \psi(y_{11}=1, y_{41}=1, k_{111}=0, k_{221}=1, k_{311}=1, k_{421}=0)\} = \min\{ 1200 + (15+25+5+10)*8,800 + (15+25+5+10)*8\} = 1220$

$F_{(2,3)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{23}=1, y_{33}=1, k_{111}=1, k_{221}=0, k_{311}=0, k_{421}=1), F_{(2,3)}(2,1) + \psi(y_{21}=1, y_{31}=1, k_{111}=1, k_{221}=0, k_{311}=1, k_{421}=1)\} = \min\{ 1200 + (15+25+5+10)*8,800 + (15+25+5+10)*8\} = 1220$

$F_{(2,4)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{23}=1, y_{43}=1, k_{111}=1, k_{221}=0, k_{311}=1, k_{421}=0), F_{(2,4)}(2,1) + \psi(y_{21}=1, y_{41}=1, k_{111}=1, k_{221}=0, k_{311}=1, k_{421}=0)\} = \min\{ 1200 + (15+25+5+10)*8,800 + (15+25+5+10)*8\} = 1220$

$F_{(3,4)}(3,1) = \min\{ F_{\phi}(2,0) + \psi(y_{33}=1, y_{43}=1, k_{111}=1, k_{221}=1, k_{311}=0, k_{421}=0), F_{(3,4)}(2,1) + \psi(y_{31}=1, y_{41}=1, k_{111}=1, k_{221}=1, k_{311}=1, k_{421}=1)\} = \min\{ 1200 + (15+25)*6 + (5+10)*4,460 + (15+25)*6 + (5+10)*4\} = 760$

$F_{\{(1,2)(3,4)\}}(3,2) = \min\{ F_{\{(1,2),(3,4)\}}(2,2) + \psi(y_{11}=1, y_{21}=1, y_{32}=1, y_{42}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(3,4)}(2,1) + \psi(y_{31}=1, y_{41}=1, y_{13}=1, y_{23}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(1,2)}(2,1) + \psi(y_{11}=1, y_{21}=1, y_{33}=1, y_{43}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0)\} = \min\{ 400 + (15+25+5+10)*4,540 + (15+25+5+10)*4,460 + (15+25+5+10)*4\} = 620$

$F_{\{(1,3)(2,4)\}}(3,2) = \min\{ F_{\{(1,3),(2,4)\}}(2,2) + \psi(y_{11}=1, y_{31}=1, y_{22}=1, y_{42}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(1,3)}(2,1) + \psi(y_{11}=1, y_{31}=1, y_{23}=1, y_{43}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(2,4)}(2,1) + \psi(y_{21}=1, y_{41}=1, y_{13}=1, y_{33}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0)\} = \min\{ 940 + (15+25+5+10)*10,800 + (15+25+5+10)*10,800 + (15+25+5+10)*10\} = 1350$

$F_{\{(1,4)(2,3)\}}(3,2) = \min\{F_{\{(1,4),(2,3)\}}(2,2) + \psi(y_{11}=1, y_{41}=1, y_{22}=1, y_{32}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(1,4)}(2,1) + \psi(y_{11}=1, y_{41}=1, y_{23}=1, y_{33}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0), F_{(2,3)}(2,1) + \psi(y_{21}=1, y_{31}=1, y_{13}=1, y_{43}=1, k_{111}=0, k_{221}=0, k_{311}=0, k_{421}=0)\} = \min\{940 + (15+25+5+10)*10,800 + (15+25+5+10)*10,800 + (15+25+5+10)*10\} = 1350$

The optimum solution is obtained from the following equation:

$$\psi^{*} = \min \begin{pmatrix} F(T, j) \\ j = 0,1,2,.., \min(T \times C, \frac{M}{LB}) \end{pmatrix} =$$

$\min\{ F_{\phi}(3,0),\ F_{(1,2)}(3,1),\ F_{(1,3)}(3,1),\ F_{(1,4)}(3,1),$

$F_{(2,3)}(3,1),\ F_{(2,4)}(3,1),\ F_{(3,4)}(3,1),$

$F_{\{(1,2),(3,4)\}}(3,2),\ F_{\{(1,3),(2,4)\}}(3,2),$

$F_{\{(1,4),(2,3)\}}(3,2)\} = \min\{1860, 790, 1220, 1220,$

$1220, 1220, 760, 620, 1350, 1350\} = 620$

Thus, the minimum total cost is 620, which corresponds to the following program. Two cells are formed, cell 1 consists of machines 1 and 2 and is formed in period 1, cell 2 consists of machines 3 and 4 and is formed in period 2.

4 Genetic algorithm

Genetic algorithms are heuristic search methods that emulate survival of the fittest with operations similar to those occurring naturally. Many researchers tried to solve optimization problems with GA or a hybrid algorithm of GA and other algorithms. Non-linear optimization is such problem that GA is applicable to solve it.

Nopiah et al. [14] developed a GA-based clustering method in cluster analysis of hetero scaled dataset. Javadian et al. [9] in a study used of GA to solve a cell formation problem. Mastorakis proposed a genetic algorithm [13] and combination of GA and Nelder-Mead [12] to solve non-linear problems. Here, we used of a proposed form of GA to solve cell formation problem.

4.1 GA approach

In this section a genetic algorithm for solving the problm is introduces. The components of genetic algorithm are selected based on Jans and Degraeve [8] study which review metaheuristic algorithm in a dynamic environment. The proposed GA consists of following steps:

4.1.1 Representation

In the increment cell formation problem, each solution is presented by a $T \times M$ matrix which rows show periods and columns show machines. The values of cells are set between zero and C_{max}, each value demonstrates the position of a machine in a period.

Table 3. Problem representation-chromosome

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	0	0	0	0	0	0
P_2	0	1	0	0	1	0
P_3	2	1	2	3	1	3
P_4	2	1	2	3	1	3

4 Periods

6 Machines

4.1.2 Initialization and evaluation

The initialization process is executed with a randomly generated solution space. An initial population size (*popsize*) is set 50. The objective function is transformed fitness function infinite cost is attached to this for infeasible solution. (Dellaert et al. [6]):

$$f^i(t)= \begin{cases} f^i_{\max} - g^i(t) & when \quad g^i(t) < f^i_{\max} \\ 0 & if\ otherwise \end{cases}$$

Where $f^i(t)$ is the fitness value of *solution i*, $g^i(t)$ is the objective function with penalty cost and $f^i_{\max}$ is the largest objeve function value in the current solution.

4.1.3 Selection strategy

The selection of individuals to produce successive generations plays an extremely important role in a genetic algorithm. There are many methods for selecting the population and each has its own advantages and disadvantages. In this study, we use the roulette wheel and elitist as selection strategies, the two most popular methods in cell formation (CF).

4.1.4 Genetic operators: crossover and mutation

Reproduction is carried out on the selected parents by using genetic operators. Crossover and mutation are the two major types of operators.

Here, the one column cross-over (Dellaert and Jeunet [5]) the matrixes of the two parents are cut in two at some random point and are recombined into one new solution. The crossover operator is given in Fig.1

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	0	0	0	0	0	0
P_2	0	1	0	0	1	0
P_3	2	1	2	3	1	3
P_4	2	1	2	3	1	3

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	1	0	0	1	0	0
P_2	1	2	2	1	0	0
P_3	1	2	2	1	0	0
P_4	1	2	2	1	3	3

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	0	0	0	0	0	0
P_2	0	1	0	0	0	0
P_3	2	1	2	0	0	0
P_4	2	1	2	3	3	3

Figure 1. Crossover operator

The mutation operator changes the value of a cell randomly, for example Fig.2 shows the mutation operator.

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	0	0	0	0	0	0
P_2	0	1	0	0	0	0
P_3	2	1	2	3	0	0
P_4	2	1	2	3	3	3

	M_1	M_2	M_3	M_4	M_5	M_6
	0	0	0	0	0	0
	0	1	0	0	0	0
	2	1	2	0	0	0
	2	1	2	3	3	3

Figure 2. Mutation operator

5 Some experiments and comparisons

In this section a number of numerical examples are solved using the multi-stage programming and genetic algorithm. Results along with the computational times and quality solutions are compared with branch and bound algorithm. The results are shown in Fig. 3. Our programs were written in Delphi 7 and tested on a PC (core2duo 2 GHz) running Windows XP Home Edition. Branch and bound (B and B) and Global solvers are applied for using Lingo 9. The results are shown in Fig. 3.The relatively few research papers that deal with the incremental cell formation problem, the experimental data are produced randomly in small, medium and large scales. The costs considered are the same as the ones specified in the previous example in section 6.

Example	Number of Parts	Number of Machines	Number of Periods	C_{max}	*B&B or Global solver (LINGO)*			Multistage Programming		Genetic Algorithm	
					Best solution	Optimal solution	Computational time	Best solution	Computational time	Best solution	Computational time
1	4	4	2	2	3050	3050	0:0:1	3050	0:0:10	3050	0:0:0
2	6	6	3	2	19560	19560	0:0:8	19560	0:0:50	19560	0:0:1
3	6	8	3	2	30284	___	0:0:20	30284	0:1:00	30284	0:0:5
4	6	10	3	2	43452	___	0:2:00	43452	0:5:00	43452	0:1:0
5	6	10	3	3	35641	___	0:3:00	35641	0:8:00	35641	0:0:48
6	6	10	3	4	28377	___	0:2:00	28377	0:7:00	28377	0:0:59
7	8	10	4	3	78743	___	0:4:00	78743	0:9:00	78743	0:1:30
8	8	10	4	4	66858	___	0:4:10	66858	0:10:20	66858	0:1:57
9	10	10	4	4	85070	___	0:10:00	85070	0:6:30	87624	0:2:08
10	10	12	4	3	129144	___	0:3:00	129144	0:7:25	129177	0:2:18
11	10	12	4	4	110646	___	0:3:20	110646	0:30:56	110646	0:2:25
12	10	12	5	3	156921	___	0:5:50	156921	0:45:35	156921	0:2:35
13	10	15	5	3	232758	___	0:6:20	232758	1:0:34	229798	0:2:22
14	10	15	5	4	367189	___	0:8:20	367189	1:35:25	367189	0:2:59
15	10	20	5	4	252894	___	0:9:0	252894	2:5:45	274937	0:3:00
15	10	20	8	4	528386	___	0:27:0	528386	2:35:33	542615	0:2:31
17	15	20	8	4	907905	___	1:45:0	___	___	949276	0:4:45
18	20	20	8	4	1383520	___	2:0:0	___	___	1129510	0:5:56
19	20	20	10	4	___	___	2:0:0	___	___	1921456	0:7:12
20	20	30	10	4	___	___	2:0:0	___	___	2134862	0:8:0

Figure 3. Comparative analysis (Computational time (hour : minute : second))

For clarity, the data of Fig. 3 is clustered into two performance measures as solution quality and computational time and demonstrated graphically in Fig 4 and 5.

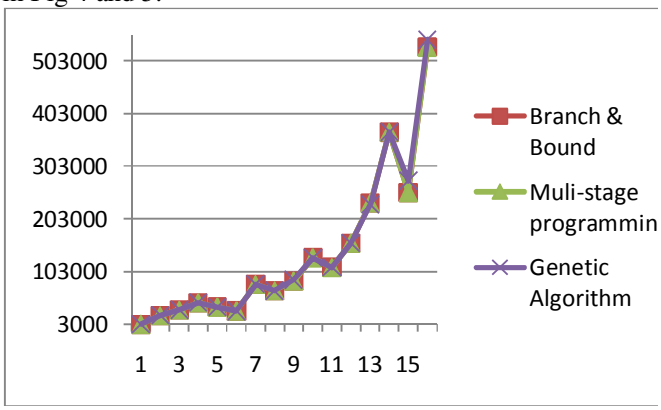

Figure 4. Comparison of solutions quality

From Fig. 4, it can be easily found that proposed genetic algorithm performs on solution quality as well as exact branch and bound algorithm, but branch and bound algorithm cannot find any feasible solution in a reasonable time. Hence the consumed computational time for solving test problems is an important performance measure. This performance measure is analyzed according Fig. 5.

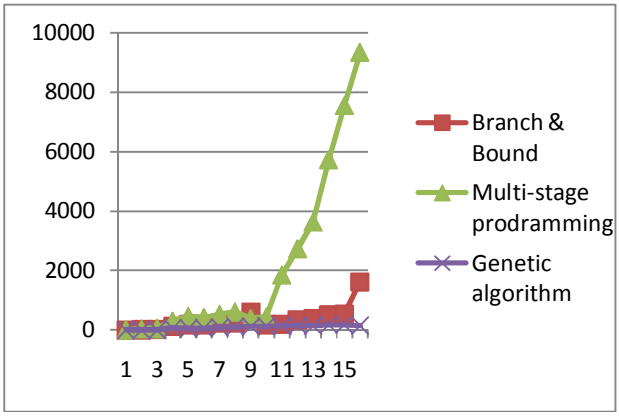

Figure 5. Comparison of computational time

It is clear that the proposed genetic algorithm is faster than exact algorithms such as branch and bound, multi-stage programming.

6 Conclusion

This paper addresses a nonlinear programming model for designing a cellular manufacturing system incrementally. The proposed algorithms based on multi-stage programming approach and genetic algorithm are applied to 20 experimental data and the results are compared with branch and bound and global solver techniques.

The branch and bound technique doesn't yield any feasible solution in a reasonable time, but the global solver finds local optimum solutions. Compared with these methods,the multi-stage method provides the optimal solutions in lesser number of iterations and number of levels for small size problems and hence the computational time is the least. For large size problems genetic algorithm is applied which produce good solutions in a reasonable time. Thus the proposed methods have the advantage of fast and accurate computations and have the ability to handle large-scale industrial problems. The present work also leads to several interesting areas of further research. Comparison of incremental and non-incremental cell formation problems and application of other metaheuristics such as simulated annealing, tabu search and etc. are the area for more work.

References:

[1] Adam, G.C. and Mihai, G.I., Mathematical methods used in engineering. Proceedings of 12th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems., Kantaoui, Sousse, Tunisia, May, 3-6, 2010.

[2] Adil, G.K. and Ghosh, J.B., Forming GT cells incrementally using GRASP. Int. J. Adv. Manuf. Technol., 26, 2005, 1402-1408.

[3] Balakrishnan, J. and Cheng, C.H., Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions, Eur. J. Oper. Res., 177, 2007, 281–309.

[4] Cruz-Suarez, H. and Ilhuicatzi-Roldan, R., Stochastic Optimal Control for Small Noise Intensities: The Discrete-Time Case. WSEAS Transactions on Mathematics., Issue 2, Volume ٩, February 2010, pp. 120-129.

[5] Dellaert, N. and Jeunet, J., Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. Int. J. Prod. Res., 38, 2000, 1083-1099.

[6] Dellaert, N., Jeunet, J. and Jonard, N., A genetic algorithm to solve the general multi-level lot-sizing problem with time varying costs. Int. J. Production Economics, 68, 2000, 241-257.

[7] Djassemi, M., A simulation analysis of factors influencing the flexibility of cellular manufacturing. Int. J. Prod. Res., 43, 2005, 2101-2111.

[8] Jans, R. and Degraeve, Z., Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. Eur. J. Oper. Res., 177, 2007, 1855-1857.

[9] Javadian, N. Rezaeian, J. and Maali.Y., Multi-objective cellular manufacturing system under machines with different life-cycle using genetic algorithm. Int. J. Appl. Sci. Eng. Technol., 4, 2007, 223-227.

[10] Mahesh, O. and Srinivasan, G., Incremental cell formation considering alternative machines. Int. J. Prod. Res., 40, 2002, 3291–3310.

[11] Manzini, R., Gambei, M., Regattieri, A. and Persona, A., Framework for designing a flexible cellular assembly system. Int. J. Prod. Res., 42, 2004, 3505-3528.

[12] Mastorakis, N.E., Genetic algorithm with Nelder-Mead optimization in the variational methods of boundary value problems. WSEAS Transactions on Mathematics., Issue 3, Volume 8, March 2009, pp. 107-116.

[13] Mastorakis, N.E., Solving Non-linear Equations via Genetic Algorithm. WSEAS Transactions on Information Science and Applications., Issue 5, Volume 2, 2005, pp. 455-459.

[14] Nopiah, Z.M., Khairir, M.I. and Abdullah. S., Time complexity estimation optimization of the genetic algorithm clustering method. WSEAS Transactions on Mathematics., Issue ٥, Volume ٩, May 2010, pp. 334-344

[15] Rezaeian, J. Norouzi, A. and Eizadi, H. Designing an incremental cellular manufacturing system based on heuristic methods. Proceedings of 14th WSEAS International Conference on Computers, Corfu Island, Greece, July, 23-25, 2010.

[16] Venkumar, P. and Noural Hag, A., Fractional cell formation in group technology using modified ART1 neural networks. Int. J. Adv. Manuf. Technol., 28, 2006, 761-765.

[17] Wemmerlov, U. and Johnson, D.J., Empirical findings on manufacturing cell design. Int. J. Prod. Res., 38, 2000, 481–507.

[18] Won, Y. and Lee, K.C., Modified *p*-median approach for efficient GT cell formation. Computer and IE., 46, 2004,495-510.