

# Cell Forming and Cell Balancing of Virtual Cellular Manufacturing Systems with Alternative Processing Routes Using Genetic Algorithm

Adib Hosseini<sup>a</sup>, Mohammad Mahdi Paydar<sup>b</sup>, Iraj Mahdavi<sup>c,\*</sup>, Javid Jouzdani<sup>d</sup>

<sup>a</sup>MSc, Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

<sup>b</sup>Assistant Professor, Department of Industrial Engineering, Babol Noshirvani University of Technology, Babol, Iran

<sup>c</sup>Professor, Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

<sup>d</sup>PhD, Department of Industrial Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

Received 15 April, 2014; Revised 21 November, 2015; Accepted 27 December, 2015

---

## Abstract

Cellular manufacturing (CM) is one of the most important subfields in the design of manufacturing systems. As a recently emerged field of study and practice, virtual cellular manufacturing (VCM) is of enormous importance as one of the types of CM. One kind of VCM problems is VCM with alternative processing routes from which the route for processing each part should be selected. In this study, a bi-objective mathematical programming model is designed in order to obtain optimal routing of parts, the layout of machines and the assignment of cells to locations while minimizing the production costs and balancing the cell loads. The proposed mathematical model is solved by multi-choice goal programming (MCGP). Since CM models are NP-Hard, a genetic algorithm (GA) is utilized to solve the model for large-sized problem instances and the results obtained from both methods are compared. Finally, a conclusion is reached and some suggestions for future works are offered.

*Keywords:* Virtual Cellular Manufacturing, Mathematical Programming, Multi-Choice Goal Programming, Genetic Algorithm.

---

## 1. Introduction

The introduction is composed of two parts: the first part is dedicated to a brief description of group technology (GT) and cellular manufacturing systems (CMS) as well as a presentation of some previous studies in this field. The second part focuses on VCM, related concepts and the previously published studies in this field.

### 1.1. Group technology and cellular manufacturing

Industries always try to maximize their productivity; therefore, in situations where there are many products with low production volumes; GT is a solution for the problem. GT is proposed for maintaining the flexibility of a Job Shop manufacturing and the efficiency of Flow Shop manufacturing. The philosophy of GT is based on grouping similar parts into groups which need similar production processes. In the literature on manufacturing, the groups of parts are called part-families and the group of similar machines are called machine-cells. CM is based on the principles of GT, which seek to take full advantage of the similarity between parts through standardization and common processing. CM leads to setup reduction and

provides the workers with the tools to operate multiple processes and to be multifunctional. It makes improvements in quality, reduces the waste, and simplifies machine maintenance. CM has helped firms to make significant improvements in throughput time performance (Isaand Tsuru, 2002; Waterson et al., 1999; Wemmerlov and Johnson, 1997; Wemmerlov and Hyer, 1989). This allows workers to easily self-balance within the cell while reducing lead times, resulting in the ability for companies to manufacture high quality products at a low cost, on time, and in a flexible way (Black, 1991). A detailed survey of studies and approaches was carried out by Papaioannou and Wilson (2010) and Paydar and Saidi-Mehrabad (2013).

### 1.2. Virtual cellular manufacturing

VCM is one of the subfields of CM in which there is no physical separator for dividing cells. VCM Systems (VCMS) are used when it is not possible to use CMS because of technical or financial perspective. VCMS aims to reduce setup times by grouping similar jobs in

---

\* Corresponding author E-mail: irajarash@rediffmail.com

production planning and control. Hence, flow time and performance of shop may be improved (Knnan and Ghosh, 1996). In this way, VCM achieves many benefits associated with CM, while retaining and building on the routing flexibility of a functional layout (Nomden and Zee, 2008). Nomden et al. (2006) defined two types of similarities and classified the studies conducted on VCM based on types of similarities they focused on. Subash et al. (2000) applied several clustering algorithms for forming virtual cells. Saad et al. (2002) presented a three-step method for production planning and cell formation. Ko and Egbelu (2003) designed two algorithms to create virtual cells. Slomp et al. (2004 and 2005) proposed a goal programming formulation that first groups jobs and machines followed by assigning workers to the groups to form virtual cells.

The concept of alternative processing routes was firstly presented by Kusiak (1987) in generalized CMS in which each part can have more than one process plans. The processing routes refer to different sequence of operations on different machines in which a part should flow to become a finished part (Nsakanda et al., 2006). Uddin and Shanker (2002) formulated a grouping problem in which each part has more than one processing route, using a GA to deal with the problem. Kia et al. (2013) presented a mathematical model for the intra-cellular layout design of dynamic cell formation in the presence of alternative process routings. The objective was to minimize the total costs of inter-cellular movement, forward and backward intra-cellular movement, setting up route, machine relocation, purchasing new machines, machine overhead and machine processing. They have developed simulate annealing algorithm with a straightforward but effective solution structure and neighborhood generation mechanism. Paydar et al. (2013) investigated the problem of designing CMS incorporating several features including sequence of operations, alternate process routings, intra-cellular layout, and reconfiguration. The main constraints were demand satisfaction, machine availability, cell size, machine capacity, material flow conservation. Computational results are demonstrated by solving some examples to verify the model. Yadollahi et al. (2014) presented a bi-objective model for a CMS considering the sequence data, alternative process plans, candidate locations for machines, maximum capacity for each machine and variable failure rate of each. The variable failure rate is considered as a dependent variable of “number of setups” and “total processing time” in a regression equation. To show the performance of the model, an example is solved using augmented e-constraint method.

When virtual cells are considered in different time horizons they are called Dynamic Virtual Cellular Manufacturing Systems (DVCMS). Mahdavi et al. (2011) developed a mathematical model of cell formation and production planning in a dynamic virtual cellular manufacturing system. The aim of their research was to

minimize backorder and holding costs and exceptional elements. Rezazadeh et al. (2011) proposed a DVCMS problem using linear programming along a swarm particle optimization algorithm. Paydar and Saidi-Mehrabad (2015) developed a multi-objective possibilistic model for simultaneous supply chain design and virtual cell formation considering multi-period production planning under uncertain demands and capacities. A revised multi-choice goal programming approach was applied to solve the mathematical model and to find a preferred compromise solution in a real world industrial case.

In this research, a mathematical model for a VCM problem is proposed considering decisions about various aspects of production. The model is then solved using two different approaches. In section 2, the mathematical model is discussed and the modification of the model for solving it by using MCGP is presented in section 3. Section 4 is dedicated to validating the model. In part 5 the model is solved using two various methods and the results are compared. The final part outlines conclusion, and suggestions for further future works in this field.

## 2. Problem Formulation

In this part a mathematical model for VCMS with alternative processing routes is formulated. The proposed model assigns machines to locations and cells, and chooses the best route among available routes for each product. Assumptions and other definitions are presented as below:

### 2.1. Assumptions

1. Each part has some pre-defined routes whose set up cost is known.
2. The demand for each part is known.
3. The intra-cell cost per distance unit is known for each product.
4. All the locations are known. Therefore, all possible pairwise distances are given.
5. The number of machines and their maintenance and overhead costs are known.
6. The operating cost per time unit is known for all the machines.
7. The processing time of each product on machines is given.
8. All of the machines have a known, finite available time.
9. Each location cannot accept more than one machine but it can remain empty.
10. Each cell has a lower limit for assigning machines.
11. The number of cells is known.

### 2.2. Sets

$i$ : Index set of part types ( $i = 1, 2, 3, \dots, P$ )

$r$ : Index set of routes of part  $i$  ( $r = 1, 2, 3, \dots, R_i$ )

$s$ : Index set of operations of part type  $i$  in route  $r$  ( $s = 1, 2, 3, \dots, S_r$ )

$j$ : Index set of machine types ( $j = 1, 2, 3, \dots, M$ )

$u$ : Index set of locations ( $u = 1, 2, 3, \dots, U$ )

$k$ : Index set of cells ( $k = 1, 2, 3, \dots, C$ )

### 2.3. Parameters

$\gamma_i^{inter}$ : Material handling cost between cells per part type  $i$

$\alpha_j$ : Maintenance and overhead cost of machine type  $j$

$\beta_j$ : Operating cost for machine type  $j$  per unit of time

$\phi_{ir}$ : Setup cost for route  $r$  of part  $i$

$\lambda_{irsj}$ : Processing time of sequence  $s$  of part type  $i$  along route  $r$  with machine  $j$

$T_j$ : Available time for machine type  $j$

$D_i$ : Demand for part type  $i$

$L_k$ : The lower bound of number of machines in cell  $k$

$d_{uu'}$ : Distance between location  $u$  and  $u'$

$a_{irsj}$ : 1 If operation  $s$  of part type  $i$  along  $r$  must be processed on machine type  $j$ ; 0 Otherwise

$A$ : A large positive number

### 2.4. Decision Variables

$X_{irsjuk}$ : 1 If operation  $s$  of part type  $i$  along route  $r$  is processed with machine type  $j$  located in location  $u$  is assigned in cell  $k$ ; 0 Otherwise

$Z_{juk}$ : 1 If machine type  $j$  is located in location  $u$  which is placed in cell  $k$ ; 0 Otherwise.

$R_{ir}$ : 1 If route  $r$  is set up to produce part type  $i$ ; 0 Otherwise.

Minimize

$$Z_1 =$$

$$\sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{j'=1}^M \sum_{u=1}^U \sum_{u'=1}^U \sum_{k=1}^C \sum_{k'=1}^C \sum_{\substack{u' \neq u \\ k' \neq k}} \gamma_i^{inter} D_i d_{uu'} X_{irsjuk} X_{irsj'uk'} \quad (1)$$

$$+ \sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^C \alpha_j Z_{juk} \quad (2)$$

$$+ \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^C \beta_j \lambda_{irsj} X_{irsjuk} D_i \quad (3)$$

$$+ \sum_{i=1}^P \sum_{r=1}^{R_i} \phi_{ir} R_{ir} \quad (4)$$

Minimize

$$Z_2 = \sum_{k=1}^C \left| \begin{array}{l} \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i - \\ \frac{1}{C} \sum_{k=1}^C \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i \end{array} \right| \quad (5)$$

Subject to:

$$\sum_{r=1}^{R_i} R_{ir} = 1 \quad \forall i \quad (6)$$

$$\sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^C X_{irsjuk} = R_{ir} \quad \forall i, r, s \quad (7)$$

$$\sum_{u=1}^U \sum_{k=1}^C X_{irsjuk} \leq A \times a_{irsj} \quad \forall i, r, s, j \quad (8)$$

$$\sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} X_{irsjuk} \leq A \times Z_{juk} \quad \forall j, u, k \quad (9)$$

$$A \times \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} X_{irsjuk} \geq Z_{juk} \quad \forall j, u, k \quad (10)$$

$$\sum_{j=1}^M \sum_{k=1}^C Z_{juk} \leq 1 \quad \forall u \quad (11)$$

$$\sum_{j=1}^M \sum_{u=1}^U Z_{juk} \geq L_k \quad \forall k \quad (12)$$

$$\sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \lambda_{irsj} X_{irsjuk} D_i \leq T_j \quad \forall j, u, k \quad (13)$$

$$X_{irsjuk}, Z_{juk}, R_{ir} \in \{0, 1\} \quad \forall i, r, s, j, u, k \quad (14)$$

The first objective function is composed of four terms and it minimizes the total cost. Term (1) denotes inter-cell cost. Term (2) denotes computing, maintenance and overhead cost of machines. Term (3) represents the operating costs of assigned machines. Term (4) denotes set up costs of selected routes.

The Second objective balances the workloads assigned to cells. The absolute value in (5) consists of two terms: the first term calculates the workload assigned to each cell and the second term denotes the average workload. Generally, this objective minimizes the difference between the total workload of each cell and the average workload considering all the cells.

Constraint (6) ensures that only one route is selected for producing each part. Constraint (7) ensures that machines must be assigned to all the sequences of a selected route. Constraint (8) guarantees that only when machine  $j$  is needed for sequence  $s$  of part  $p$  along route  $r$ , variable  $X_{irsjuk}$  can assign it to a location. In any other

situation, variable  $X_{irsjuk}$  is forced to have 0 value. Constraints (9) and (10) ensure that if machine type  $j$  is assigned to location  $u$  which is assigned to cell  $k$ , variable  $Z_{juk}$  accepts value 1. Constraint (11) doesn't let a location accept more than one machine. Inequality (12) is related to the lower limit of number of machines in each cell. Constraint (13) doesn't let workloads to be assigned to machines more than their available time. Constraint (14) defines the variables types.

### 3. Linearization and Modification of the Model

The above model is a bi-objective problem; therefore a Multi Objective Decision Making (MODM) method like, Weighting Method, Goal Programming, MCGP, Lexicographic Method,  $\epsilon$ -Constraint and etc. should be used to solve the model. Here a MCGP is used to solve it. For solving the model using MCGP it should be modified and linearized.

#### 3.1. Linearization

The objective functions of the proposed model are non-linear. The first objective function contains a multiplication of two binary variables. For linearizing, a binary variable  $W_{irss+1jj'uu'kk'}$  is defined and two inequalities are added to the model. The inequalities are presented as below. These inequalities guarantee that only moment variables  $X_{irsjuk}$  and  $X_{irs+1j'u'k'}$  have value 1, the variable  $W_{irss+1jj'uu'kk'}$  has value 1. Otherwise it would be 0.

$$X_{irsjuk} + X_{irs+1j'u'k'} - 1 \leq W_{irss+1jj'uu'kk'} \quad \forall i, r, s, j, j', u, u', k, k' \quad (15)$$

$$W_{irss+1jj'uu'kk'} \leq (X_{irsjuk} + X_{irs+1j'u'k'}) / 2 \quad \forall i, r, s, j, j', u, u', k, k' \quad (16)$$

Due to positivity of coefficients of variable  $W_{irss+1jj'uu'kk'}$ , the constraint (16) can be omitted and the variable  $W_{irss+1jj'uu'kk'}$  can be relaxed to a non-negative real variable (Solimanpur and Kamran, 2010).

There is an absolute value function in the second objective which should be linearized. For omitting the absolute value sign, two non-negative real variables,  $f_1$  and  $f_2$ , are used. In equation (17), when absolute value sign contains a negative value, the variable  $f_2$  takes the absolute value; on the other hand, if the absolute value sign contains a positive value, then the  $f_1$  variable takes the value.

$$\sum_{k=1}^C \left( \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i - \frac{1}{C} \sum_{k=1}^C \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i \right) = f_1 - f_2 \quad (17)$$

#### 3.2. Modification

A general form for MCGP is as what follows.  $g_{ij}$  is  $j$ th aspiration level for  $i$ th goal.

$$\text{Minimize} \quad \sum_{i=1}^n |f_i(X) - g_{i1} \text{ or } g_{i2} \text{ or } \dots \text{ or } g_{im}| \quad (18)$$

Subject to:

$$X \in F \text{ (} F \text{ is a feasible set)} \quad (19)$$

As mentioned before, for solving the model using MCGP, objective functions should be modified. For each objective two non-negative deviation variables are added and subtracted, one for positive deviations ( $d^+$ ) and the other for negative deviations ( $d^-$ ).

$$OBJ_l - d_l^+ + d_l^- = \sum_{q=1}^{N_g} g_q^l \times b_q^l \quad \forall l = 1, 2 \quad (20)$$

Also some aspiration levels ( $g_q$ ) for each objective are considered which are multiplied by binary variables ( $b_q$ ). In the end, two equalities are added to force the model to accept only one aspiration level for each goal.

$$\sum_{q=1}^{N_g} b_q^l = 1 \quad \forall l = 1, 2 \quad (21)$$

The achievement function consists of deviations of goals and a weighting variable ( $0 \leq \omega \leq 1$ ) is used to show the importance of the goals. In this article, we assume that we have  $\omega=0.5$ ; however, it can be changed according to decision maker's opinion. The linearized model based on MCGP is presented as follows.

Minimize

$$TOTALDEV = \omega(d_1^+ + d_1^-) + (1 - \omega)(d_2^+ + d_2^-) \quad (22)$$

Subject to:

$$OBJ_l - d_l^+ + d_l^- = \sum_{q=1}^{N_g} g_q^l \times b_q^l \quad (23)$$

$$\sum_{q=1}^{N_g} b_q^1 = 1 \quad (24)$$

$$OBJ_1 = \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}-1} \sum_{j=1}^M \sum_{j'=1}^M \sum_{u=1}^U \sum_{u'=1}^U \sum_{k=1}^C \sum_{k'=1}^C \gamma_i^{inter} D_i d_{uu'} W_{irss+1jj'uu'kk'} \quad (25)$$

$$+ \sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^C \alpha_j Z_{juk} \quad (26)$$

$$+ \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^K \beta_j \lambda_{irsj} X_{irsjuk} D_i \quad (27)$$

$$+ \sum_{i=1}^P \sum_{r=1}^{R_i} \phi_{ir} R_{ir} \quad (28)$$

$$OBJ_2 - d_2^+ + d_2^- = \sum_{q=1}^{N_g} g_q^2 \times b_q^2 \quad (29)$$

$$\sum_{q=1}^{N_g} b_q^2 = 1 \quad (30)$$

$$\sum_{k=1}^C \left( \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i - \frac{1}{C} \sum_{k=1}^C \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \sum_{j=1}^M \sum_{u=1}^U \lambda_{irsj} X_{irsjuk} D_i \right) = \quad (31)$$

$$f_1 - f_2 \quad (32)$$

$$OBJ_2 = f_1 + f_2 \quad (32)$$

$$\sum_{r=1}^{R_i} R_{ir} = 1 \quad \forall i \quad (33)$$

$$\sum_{j=1}^M \sum_{u=1}^U \sum_{k=1}^C X_{irsjuk} = R_{ir} \quad \forall i, r, s \quad (34)$$

$$\sum_{u=1}^U \sum_{k=1}^C X_{irsjuk} \leq A \times a_{irsj} \quad \forall i, r, s, j \quad (35)$$

$$\sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} X_{irsjuk} \leq A \times Z_{juk} \quad \forall j, u, k \quad (36)$$

$$A \times \sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} X_{irsjuk} \geq Z_{juk} \quad \forall j, u, k \quad (37)$$

$$\sum_{j=1}^M \sum_{k=1}^C Z_{juk} \leq 1 \quad \forall u \quad (38)$$

$$\sum_{j=1}^M \sum_{u=1}^U Z_{juk} \geq L_k \quad \forall k \quad (39)$$

$$\sum_{i=1}^P \sum_{r=1}^{R_i} \sum_{s=1}^{S_{r_i}} \lambda_{irsj} X_{irsjuk} D_i \leq T_j \quad \forall j, u, k \quad (40)$$

$$X_{irsjuk}, Z_{juk}, R_{ir}, b_q^1, b_q^2 \in \{0, 1\} \quad (41)$$

$$\forall i, r, s, j, u, k, q$$

$$d_1^+, d_1^-, d_2^+, d_2^-, f_1, f_2, W_{irss+1jj'uu'kk'} \geq 0 \text{ And real} \quad (42)$$

$$\forall i, r, s, j, j', u, u', k, k'$$

#### 4. Solving the Mathematical Model

##### 4.1. Exact method

In this part the modified model is solved by LINGO9.0<sup>®</sup> using branch and bound method. Two randomly generated examples of sizes are generated and are solved by the aforementioned software. The general information of examples is presented in Table 1 and the decision variables of example 2 are shown in Table 2 and Table 3.

Table 1  
Randomly generated examples and exact method results

	Example Sizes				Model Sizes		Lingo 9.0 Results	
	No. of Parts	No. of Machines	No. of Locations	No. of Cells	No. of Variables	No. of Constraints	CPU Time(s)	Achievement Fun.
Example 1	3	5	5	2	6978	25086	18	1266850
Example 2	4	5	7	3	79521	306549	671	970308

Table 2  
Decision variables of example 1

Variable	Value
X( PART1, ROUTE3, SEQUENCE1, MACHINE2, LOCATION4, CELL2)	1
X( PART1, ROUTE3, SEQUENCE2, MACHINE5, LOCATION1, CELL2)	1
X( PART2, ROUTE1, SEQUENCE1, MACHINE3, LOCATION5, CELL1)	1
X( PART2, ROUTE1, SEQUENCE2, MACHINE1, LOCATION3, CELL1)	1
X( PART3, ROUTE1, SEQUENCE1, MACHINE2, LOCATION4, CELL2)	1
X( PART3, ROUTE1, SEQUENCE2, MACHINE3, LOCATION2, CELL2)	1
Z( MACHINE1, LOCATION3, CELL1)	1
Z( MACHINE2, LOCATION4, CELL2)	1
Z( MACHINE3, LOCATION2, CELL2)	1
Z( MACHINE3, LOCATION5, CELL1)	1
Z( MACHINE5, LOCATION1, CELL2)	1
R( PART1, ROUTE3)	1
R( PART2, ROUTE1)	1
R( PART3, ROUTE1)	1

Table 3  
Decision variables of example 2

Variable	Value
X( PART1, ROUTE1, SEQUENCE1, MACHINE1, LOCATION6, CELL2)	1
X( PART1, ROUTE1, SEQUENCE2, MACHINE2, LOCATION7, CELL1)	1
X( PART2, ROUTE1, SEQUENCE1, MACHINE3, LOCATION2, CELL2)	1
X( PART2, ROUTE1, SEQUENCE2, MACHINE5, LOCATION4, CELL2)	1
X( PART3, ROUTE2, SEQUENCE1, MACHINE2, LOCATION5, CELL3)	1
X( PART3, ROUTE2, SEQUENCE2, MACHINE4, LOCATION3, CELL3)	1
X( PART4, ROUTE2, SEQUENCE1, MACHINE4, LOCATION1, CELL1)	1
X( PART4, ROUTE2, SEQUENCE2, MACHINE2, LOCATION7, CELL1)	1
Z( MACHINE1, LOCATION6, CELL2)	1
Z( MACHINE2, LOCATION5, CELL3)	1
Z( MACHINE2, LOCATION7, CELL1)	1
Z( MACHINE3, LOCATION2, CELL2)	1
Z( MACHINE4, LOCATION1, CELL1)	1
Z( MACHINE4, LOCATION3, CELL3)	1
Z( MACHINE5, LOCATION4, CELL2)	1
R( PART1, ROUTE1)	1
R( PART2, ROUTE1)	1
R( PART3, ROUTE2)	1
R( PART4, ROUTE2)	1

CM problems are considered as NP-Hard problems; therefore with an increase in example size, the computing time for exact methods increases exponentially. The LINGO software could not get the input data for larger-sized problems. This shows the inefficiency of exact methods for such NP-Hard problems. Therefore, a meta-heuristic method should be utilized to solve real size problems in a reasonable time. In this research, a GA-based method is developed and used.

4.2. Heuristic method (genetic algorithm)

A GA based on MCGP is designed for the proposed model. GA was introduced by John Holland in 1975 and was inspired from Darwinian Evolution Theory. Based on this theory, individuals who fit more to their environment are more likely to transfer their genes to the next generation. GA starts with some initial random solutions called *initial population* and each member of population is called a *chromosome*. The chromosomes are modified and merged for producing next generation. The

modification and merging process are known as mutation and crossover functions. This process is repeated until the stopping criterion is met. Here, it is a general form of the GA (Mahdavi et al., 2009) which

1. Generates initial population randomly.
2. Evaluates and computes fitness for each chromosome.
3. Selects the parents using the roulette wheel technique and then applies mutation and crossover to the selected parents to produce new offspring and evaluates the newly generated offspring.
4. Selects new generation from offspring using elitism policy.
5. If stopping criterion is met stop the algorithm, otherwise go to step 3.

4.2.1. Chromosome representation

For implementing GA, there is a need to show chromosomes in strings. The chromosomes of the designed algorithm are composed of three matrices. Fig. 1

shows a general form of a chromosome. First matrix is a  $P \times \max_i(R_i)$  matrix which represents the variable  $R_{ir}$ .

The rows of this matrix show the parts, and the columns show the possible routes; therefore, as an example, if route 3 is selected for part 1, then all of the elements of row 1 is zero except  $R_{13}$  which is equal to 1. This matrix configuration satisfies constraint (6).

Second part of the chromosome is a  $U \times C$  matrix whose rows refer to locations and the columns refer to cells. This matrix represents the variable  $Z_{juk}$ . As an example, if machine 2 is assigned to location 4 and location 4 is grouped in cell 2, the element  $Z_{42} = 2$  and the other elements of row 4 are equal to 0. Constraint (11) is satisfied by this matrix and the number of machines for

each cell is measured by counting non-zero elements of each column of the matrix. It is necessary to consider constraint (12) while filling second matrix.

Third part of the chromosome is a  $P \times \max_i(S_i)$  matrix in which the rows refer to parts and the columns represent the sequences. The elements of this matrix refer to locations which contain needed machine for that sequence. For example, if element of row 3 and column 2 is equal to 4, it means the second operation of part 3 is performed by the machine assigned to location 4. The collection of all the three matrices represent variable  $X_{irsjuk}$ . It is worth noting that the constraints (7), (8), (9), (10) and (13) should be considered while filling the third matrix.

An example for chromosome representations is shown in Fig. 2.

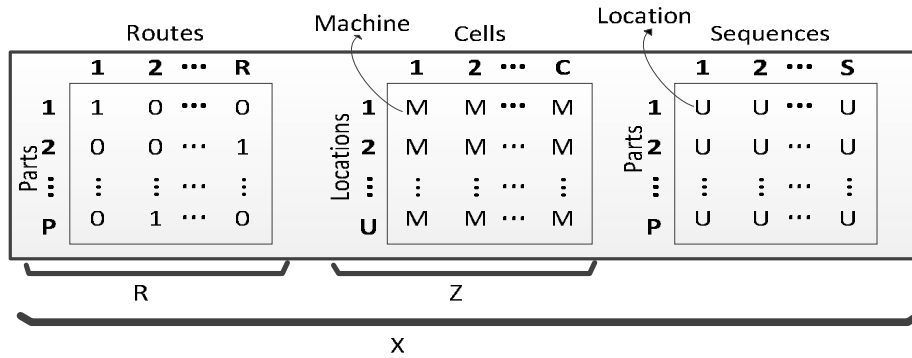


Fig. 1. Chromosome representation

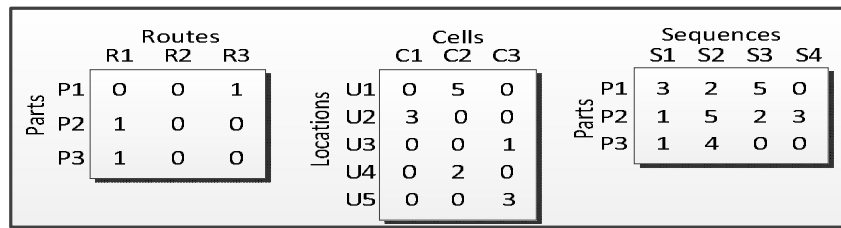


Fig. 2. Example for chromosome representation

Objective function is calculated based on MCGP. Two  $1 \times N_g$  matrices are defined in which  $N_g$  is the number of aspiration levels used by decision maker. Each of these matrices shows the selected aspiration level for goals. For example if the 4<sup>th</sup> aspiration level is selected for goal 2, then the element  $b_{14}$  is equal to 1 and all other elements are equal to 0 (Fig. 3). Finally, the total deviation is calculated for each chromosome.

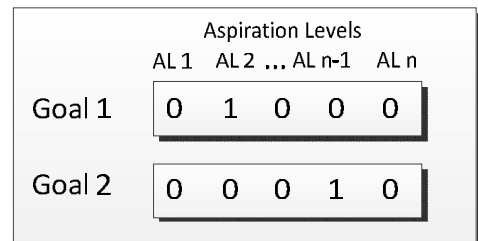


Fig. 3. Example for selecting aspiration levels

4.2.2. Selection

GA needs to select parents for crossover and mutation functions. Selection process is used to stochastically select the basis of the next generation based on chromosomes fitness values. The chromosome fitness value is calculated by inverting the total deviation of the chromosome. The probability of selecting a chromosome is equal to the fitness value divided by the sum of fitness values of all chromosomes.

$$Fitness_i = (TotalDeviation_i)^{-1}$$

$$\forall i = 1, 2, \dots, PopulationSize \quad (43)$$

$$P(Selection_i) = \frac{Fitness_i}{\sum_{i=1}^{PopulationSize} Fitness_i}$$

$$\forall i = 1, 2, \dots, PopulationSize \quad (44)$$

If the total deviation of a chromosome is lower than the others, then its probability of selection will be higher and this makes the generations evolve through successive iterations.

4.2.3. Crossover

When two parents are selected by selection function, they should be merged using crossover function. The

crossover is done only on matrix *R* or matrix *Z* because the probability of exiting from solution space is increased by using the crossover on both matrices. For a definite proportion of population, the crossover function is applied on matrix *R* and the remainder is done on matrix *Z*. As an example, if 100 parents are selected for crossover function, the crossover operator is applied on 30% of the population (30 chromosomes) on matrix *R* and consequently, the crossover operation is performed on the other 70 on matrix *Z*.

First a row is selected randomly with uniform chance of selection. Then, the first *n* rows (with *n* being determined beforehand) are inherited from parent 1 and the other rows are inherited from parent 2. For the other child, it is vice versa. Fig. 4 shows the crossover function process.

4.2.4. Mutation

Mutation increases the diversification of algorithm and makes it possible to search the solution space thoroughly. Two types of mutation are used, one on matrix *R* and the other on matrix *Z*. First, mutation selects a row from matrix *R* and rebuilds it randomly. Second type of mutation selects randomly two rows of matrix *Z* and swaps them. Fig. 5 and Fig. 6 show two types of mutation for matrix *R* and matrix *Z*.

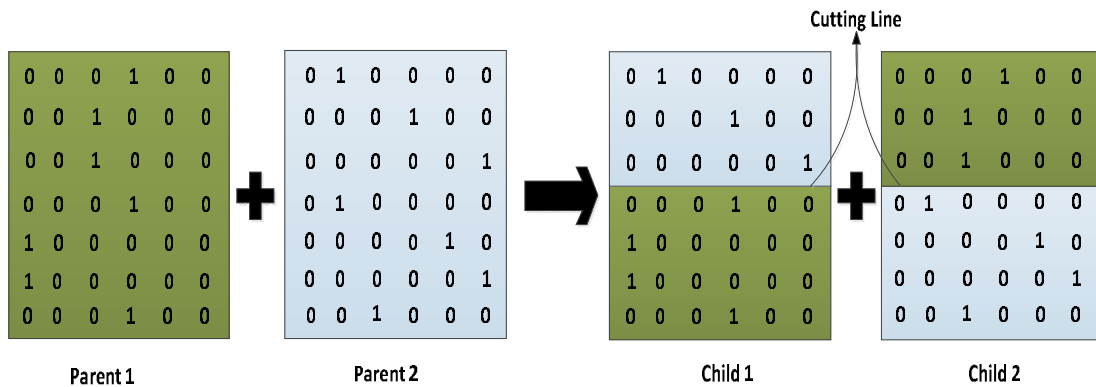


Fig. 4. Crossover function

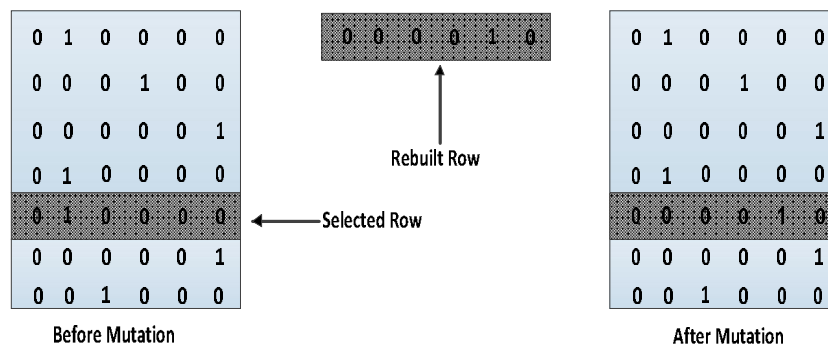


Fig. 5. Mutation for Matrix R



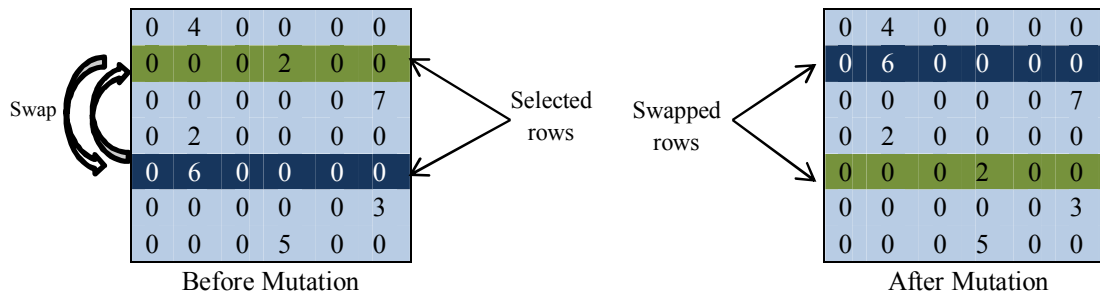


Fig. 6. Mutation for Matrix Z

Because of strong relationships between all parts of each chromosome, it is possible that invalid solutions be produced by crossover and mutation functions; therefore, a checking function is used to check the satisfaction of constraints and feasibility of the offspring. If the checking function rejects the feasibility of the children, new parents are selected and the related function is performed again. Otherwise the children are transferred to the mating pool. This procedure is repeated until the needed population for next generation is obtained.

The checking function ensures that the matrix R and matrix Z are kept in feasible space. To keep matrix X in the solution space after crossover and mutation procedures, it is randomly rebuilt according to constraints. This makes sure that the whole solution is a feasible solution.

4.2.5. Handling infeasible chromosomes

Two mechanisms are designed to deal with the infeasible chromosomes: one mechanism is applied immediately after the mutation and crossover operations to check that if the constraints (6), (11), (12) and (13) are held. If the chromosome does not satisfy the aforementioned constraints, the mutation and crossover operations are repeated until a feasible solution is obtained.

The other mechanism reconstructs the whole matrix, X, to make it feasible. Specifically, after utilizing mutation and crossover operator for Z and R matrices, matrix X is totally reconstructed in order to prevent creating infeasible chromosomes. The reason behind this mechanism is the fact that there is a strong relationship between different parts of a chromosome; especially, for matrix X. It should be noted that other constraints are maintained by the way the solution is represented.

4.2.6. Elitism

When offspring are generated, they are placed in a mating pool with the parents, and then the best individuals are selected for the next generation. This procedure makes sure that if there are any individual with suitable characteristics among parents they are transferred to the next generation.

4.2.7. Stopping criterion

In this study a convergence condition is used for stopping criterion. If the mean of objective function of population is equal to the best of objective function for several iterations it shows that the population is converged to one solution and the algorithm will stop (Bootaki et al., 2014).

4.2.8. Numerical results

The randomly generated examples presented in the previous section are solved and the results are described and compared in Table 4. The comparison between GA and LINGO9.0 software shows the efficiency of GA. Because of negligible deviation from optimal solution in small- and medium-sized problems, GA can be used for large-sized problems (Gap column). For large-sized problems the LINGO 9.0 software could not even reach to feasible space after several hours of computing (Example 7-10).

The convergence charts for example 1 and example 2 are shown in Fig. 7 and Fig. 8 (Red dots are best values of objective function and blue dots are mean of objective function on each iteration), respectively. The results show that the branch and bound method is not suitable for medium- and large-sized problems; however, the designed GA yields satisfying results in shorter times.

Table 4  
Comparison of GA and MCGP

	Example Sizes				Lingo 9.0 Results		GA Results		Gap
	No. of Parts	No. of Machines	No. of Locations	No. of Cells	CPU Time (s)	Objective function	CPU Time (s)	Objective function	
Example 1	3	5	5	2	18	1266850	3	1266850	0
Example 2	4	5	7	3	671	970308	57	970441	0.02
Example 3	6	8	10	2	8083	3234821	158	3286578	1.6
Example 4	8	8	12	2	22984	5716492	252	5853687	2.4
Example 5	10	14	16	2	36000	7131675	347	7359887	3.2
Example 6	10	14	16	3	36000	6712903	335	7088825	5.6
Example 7	12	15	20	2	36000	-	446	93802171	-
Example 8	14	15	20	2	36000	-	476	107619615	-
Example 9	14	17	22	3	36000	-	554	130729164	-
Example 10	18	20	25	3	36000	-	743	141620361	-

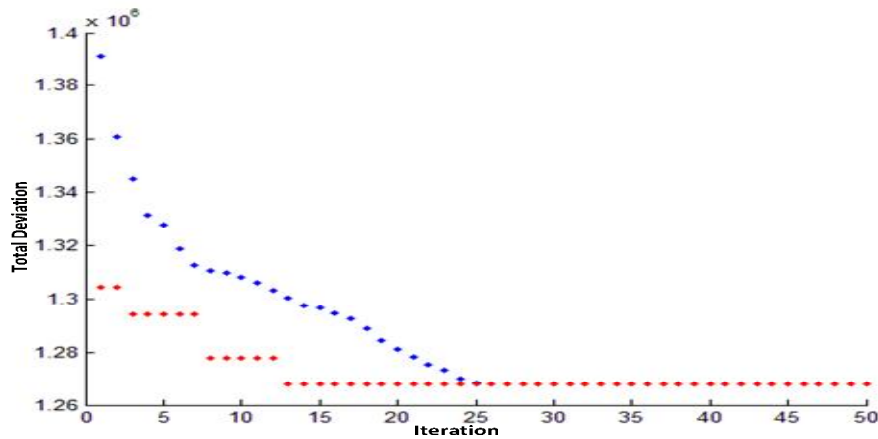


Fig. 7. Convergence curves for Example 1

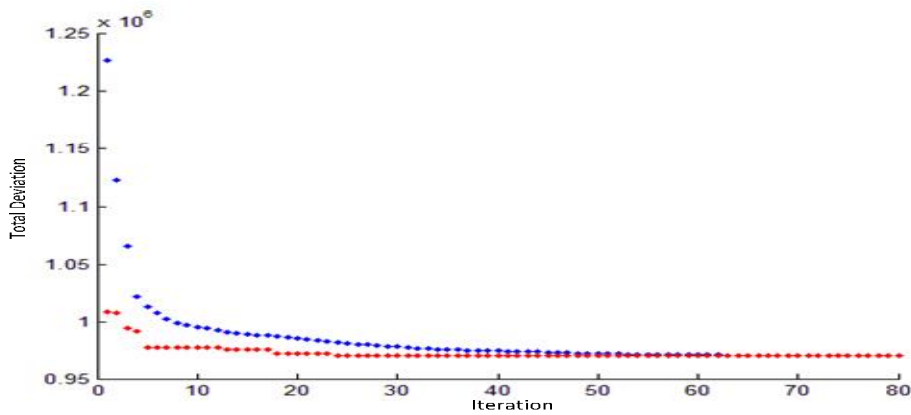


Fig. 8. Convergence curves for Example 2

### 5. Conclusion

In this paper, a mathematical model is designed for solving a VCM problem to decide about various design factors such as selecting machines and locations, grouping machines in virtual cells and selecting suitable route among predefined routes while minimizing the intra-cell movements and balancing the cells loads. The proposed model is solved by MCGP for small problems. However, for medium- and large-size problems, the MCGP is not

efficient. Therefore a GA is designed based on MCGP and the results show the effectiveness of GA compared to exact method.

The problem can be extended in various ways. For example, the demand can be considered as a stochastic or fuzzy parameter and the problem can take place in continuous environment. Other design features such as worker assignment, outsourcing, considering time

periods, machine utilization, etc. can be considered. Solution approaches are other ways of extending this study by using exact methods or heuristic methods such as NSGA-II, Ant Colony, Taboo Search and etc.

## References

- Black, J. (1991). *The Design of the Factory with a Future*. McGraw-Hill, Inc.
- Bootaki, B., Mahdavi, I. and Paydar, M. (2014). A hybrid GA-AUGMECON method to solve a cubic cell formation problem considering different worker skills. *Computers & Industrial Engineering*, 75, 31-40.
- Isa, K. and Tsuru, T. (2002). Cell Production and Workplace Innovation in Japan: Toward a New Model for Japanese Manufacturing? *Industrial Relations*, 41(4), 548–578.
- Kia, R., Javadian, N., Paydar, M. and Saidi-Mehrabad, M. (2013). A simulated annealing for intra-cell layout design of dynamic cellular manufacturing systems with route selection, purchasing machines and cell reconfiguration. *Journal of Operational Research*, 30(4), 1-41.
- Knnan, V. and Ghosh, S. (1996). A virtual cellular manufacturing approach to batch production. *Decision Sciences*, 27, 519-539.
- Ko, K. C. and Egbelu, P. J. (2003). Virtual cell formation, *International Journal of Production Research*. *International Journal of Production Research*, 41(11), 2365–2389.
- Kusiak, A. (1987). The generalized group technology concept. *International Journal of Production Research*, 24(4), 561-569.
- Mahdavi, I., Aalaei, A., Paydar, M. and Solimanpur, M. (2011). Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems. *International Journal of Production Research*, 49(21), 6517-6537.
- Mahdavi, I., Paydar, M., Solimanpur, M. and Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications*, 36(3), 6598-6604.
- Nomden, G. and Zee, D. J. (2008). Virtual cellular manufacturing: Configuring routing flexibility. *International Journal Production Economics*(112), 439-451.
- Nomden, G., Slomp, J. and Suresh, N. C. (2006). Virtual Manufacturing Cells: a taxonomy of past research and identification of future research issues. *International Journal of Flexible Manufacturing Systems*, 17(2), 71–92.
- Nsakanda, A., Diaby, M. and Price, W. (2006). Hybrid genetic approach for solving largescale capacitated cell formation problems with multiple routings. *European Journal of Operational Research*, 171, 1051-1070.
- Papaioannou, G. and Wilson, J. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, 206(3), 509-521.
- Paydar, M. and Saidi-Mehrabad, M. (2013). A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy. *Computers & Operations Research*, 40(4), 980-990.
- Paydar, M. and Saidi-Mehrabad, M. (2015). Revised multi-choice goal programming for integrated supply chain design and dynamic virtual cell formation with fuzzy parameters. *International Journal of Computer Integrated Manufacturing*, 28(3), 251-265.
- Paydar, M., Saidi-Mehrabad, M. and Kia, R. (2013). Designing a new integrated model for dynamic cellular manufacturing systems with production planning and intra-cell layout. *International Journal of Applied Decision Sciences*, 6(2), 117-143.
- Rezazadeh, H., Mahini, R. and Zarei, M. (2011). Solving a Dynamic Virtual Cell Formation Problem by Linear Programming Embedded Particle Swarm Optimization Algorithm. *Applied Soft Computing*, 11(3), 3160-3169.
- Saad, S. M., Baykasoglu, A. and Gindy, N. Z. (2002). An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells. *Production Planning & Control*, 13(4), 381–393.
- Slomp, J., Chowdary, B. V. and Suresh, N. C. (2004). Design and operation of virtual manufacturing cells. *Proceedings of FAIM Conference*. Tampa, FL.
- Slomp, J., Chowdary, B. V. and Suresh, N. C. (2005). Design and Operation of Virtual Manufacturing Cells. *Robotics and Computer Integrated Manufacturing*, 21(3), 273–288.
- Solimanpur, M., & Kamran, M. A. (2010). Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm. *Computers & Industrial Engineering*, 59(4), 830-839.
- Subash Babu, A., Nandurkar, K. N. and Thomas, A. (2000). Development of virtual cellular manufacturing systems for SMEs. *Logistics Information Management*, 13(4), 228–242.
- Uddin, M. and Shanker K. (2002). Grouping of parts and machines in presence of alternative process routes by genetic algorithm. *International Journal of Production Economics*, 76(3), 219-228.
- Waterson, P. E., Clegg, C. W., Bolden, R., Pepper, K., Warr, P. B. and Wall, T. D. (1999). The use and effectiveness of modern manufacturing practices: a survey of UK industry. *International Journal of Production Research*, 37(10), 2271–2292.
- Wemmerlov, U. and Hyer, N. L. (1989). Cellular manufacturing in the U.S. industry: a survey of users. *International Journal of Production Research*, 27(9), 1511–1530.
- Wemmerlov, U. and Johnson, D. J. (1997). Cellular manufacturing at 46 user plants: implementation experiences and performance improvements. *International Journal of Production Research*, 35(1), 29–49.
- Yadollahi, M., Mahdavi, I., Paydar, M. and Jouzdani, J. (2014). Design a bi-objective mathematical model for cellular manufacturing systems considering variable failure rate of machines. *International Journal of Production Research*, 52(24), 7401-7415.

