

A two-layer genetic algorithm for the design of reliable cellular manufacturing systems

Hassan Rezazadeh^{a*} and Amin Khiali-Miab^a

^aDepartment of Industrial Engineering, Faculty of Mechanical Engineering, University of Tabriz, Iran

CHRONICLE

Article history:

Received October 2 2016
Received in Revised Format
October 28 2016
Accepted January 2 2017
Available online
January 5 2017

Keywords:

Cell Formation
Reliability
Mathematical Model
Two-Layer Genetic Algorithm

ABSTRACT

This study presents a new mathematical model for the design of reliable cellular manufacturing systems, which leads to reduced manufacturing costs, improved product quality and improved total reliability of the manufacturing system. This model is expected to provide a more noticeable improvement in time and solution quality in comparison with other existing models. Each part to be manufactured may select each of the predefined manufacturing routes, such that the total reliability of the system is increased. On the other hand, the model adopts to categorize the machines to determine the manufacturing cells (cell formation) and reduce the transportation costs. Thereby, both criteria of system reliability and manufacturing costs will be simultaneously improved. Due to the complexity of cell formation problems, a two-layer genetic algorithm is applied on the problem in order to achieve near optimal solutions. Furthermore, the performance of the proposed algorithm is shown for solving some computational experiments. Finally, the results of a practical study for designing a cellular manufacturing system as a case study in Iranian Diesel Engine Manufacturing Co., Tabriz, Iran are present.

© 2017 Growing Science Ltd. All rights reserved

1. Introduction

System definition is an approach to develop sciences for involvement of interactions in the problem and realization of the decisions and results more than before. Reliability index can be defined for whatever known as the system. It is for many year, all effective components in manufacturing a product are known as the manufacturing system. In other words, having a same goal (producing a qualified product) the components have an interactive cooperation and influence. Taking into account the above mentioned definitions, a reliable manufacturing system can also be discussed. In the literature, the previous studies about reliability theory are mainly limited to its definition in the field of physical product, although a reliable product will be manufactured in a reliable system. There are some papers on this context, but we could say that some characteristics of the suggested mathematical models have made them rather impractical or of limited industrial application. The evolution of these papers will be addressed in the next section.

* Corresponding author

E-mail: hrezazadeh@tabrizu.ac.ir (H. Rezazadeh)

The remainder of this paper is organized as follows: In Section 2, we review relevant literature on the cellular manufacturing system (CMS). Section 3 presents the reliability considerations in designing CMS. Section 4 and 5 presents the Problem description and mathematical formulation for the CMS. In Section 6, we introduce a brief review of Two-Layer Genetic Algorithm. Computational results are reported in Section 7 and the conclusion is given in Section 8.

2. Literature Review

Machinery is the main element of a manufacturing system. They play a key role in quality and price of the products. Maintenance and repair conditions are also of major concerns on these two factors of the products (Das et al., 2007a). Frequent breakdowns and repeated repairing and resetting the machinery lead to non-uniformity and reduced quality of the products in addition to incur greater maintenance and labor costs. Therefore, organization of the machinery in the form of an effective manufacturing system may basically improve quality and costs of the products. Conventional manufacturing paradigms (Job shop and Flow shop) have found it difficult to satisfy the competitive advantages as well as control and improve the manufacturing efficiency (Wemmerlov & Hyer, 1989; Wemmerlov & Johnson, 1997; Askin, 1999; Das et al., 2007b). Although advantages of the cellular manufacturing systems are properly addressed by either users or researchers, some others have criticized about drawbacks of these systems (Flynn & Jacobs, 1986; Morris & Tersine, 1990; Boughton & Arokiam, 2000; Agarwal & Sarkis, 1998; Das et al., 2007b). Their results can be summarized as below:

- 1- *Cellular Manufacturing System (CMS) decreases flexibility of the system (in comparison to Job shop manufacturing),*
- 2- *CMS reduces utilization rate of machine due to allocation of them to cells,*
- 3- *Breakdowns of machines significantly influence due date of operations,*
- 4- *Increased in-process inventories due to allocation of machines to cells (Das et al., 2007b),*
- 5- *Among parameters which influence the CMS, allocation of machines to cells, combination of manufactured parts and reliability of machinery seem to play the most significant role in performance of a CMS (Seifoddini and Djassemi, 2001; Das et al., 2007b).*

A considerable amount of researchers on the field of CMS previously have presumed the machinery 100% reliable, while more recent works have embedded the reliability in a rather complicated and undesirable way in the CMS problems (Das et al., 2007a; Das et al., 2007b; Jabal Ameli et al., 2007; Jabal Ameli & Arkat, 2008; Saxena & Jain, 2011). Das et al. have treated reliability of the manufacturing system as being serial so they have deemed breakdown of a machine equal to that of the whole manufacturing system (Das et al., 2007a; Das et al., 2007b). Approximately 75% of the manufacturing systems produce their products as a large set of parts via batch processing (Akturk and Turkcan, 2000 ; Saxena & Jain, 2011). In this case, the whole system does not necessarily stop with breakdown of a machine, because the system will never stop if repairing the failed machine lasts for some time shorter than the current batch operation. Thereby, the system will be incurred heavy costs in order to retain a reliability level which is not needed in practice. Complexity of the manufacturing systems in comparison with other common engineering systems will make it impossible to be dealt with them like electronic systems. This will practically limit application of the paper of K. Das et al. However, other authors have just considered the costs of system breakdown in a way that can be improved (Jabal Ameli et al., 2007; Arkat, 2008; Saxena & Jain, 2011).

A complete set of reliability-related costs have been addressed in this paper by adding the relevant term of maintenance labor force to the objective function. These costs are: (1) calculation of hardware costs for each machine, (2) calculation of maintenance labor force in each breakdown, (3) breakdown costs of production line. This three mentioned reliability-related costs in addition to transportation costs will form the objective function.

Other characteristic of the objective function presented in this paper is proper linearization which reduces

the number of integer variables of this model in comparison with other models suggested in (Jabal Ameli et al., 2007; Jabal Ameli and Arkat, 2008). This is expected to reduce the computational effort for solving by exact algorithms. In average, 20-50% of costs in the manufacturing systems are associated with intercellular and intracellular transportations as well as proper innovative planning in allocation of machinery and transportation. This can decrease 10-30% of the production costs (Tompkins et al., 2003; Saxena and Jain, 2011). Another issue which must be discussed in relation with the objective function of this model is that the costs of intercellular transportation are considered in it, which plays a significant role in efficiency of the model. A great number of studies have developed their own model by assuming only one processing route (Lokesh & Jain, 2008, 2009, 2010a, 2010b). Some authors have discussed on the flexibility obtained from alternative process routing problems for the parts (Gupta et al., 1996; Sofianopoulou, 1999; Zhao and Wu, 2000; Kim et al., 2004). This assumption has several features in design phase, as summarized below: less needed machinery, greater utilization rate of the machinery, less intercellular transportation (less relation between the cells), and improved output of the system (Kusiak, 1987). Moreover, multiple processing routes were mentioned in some studies with the effect and importance of this assumption being shown on the results obtained from this model (Defersha and Chen, 2006a; 2006b, 2007, 2008; Ahkioon et al., 2009, Bulgak & Bektas, 2009; Nsakanda, 2006).

3. Problem description and formulation

3.1. Notations

3.1.1. Indexes

i	Index of part type where $i=1, 2, \dots, n$
j	Index of process routing where $j=1, 2, \dots, q_i$
k	Index of machine type where $k=1, 2, \dots, m$
l	Index of cell number where $l=1, 2, \dots, C$
r	Index of process in a rout $r=1, 2, \dots, k_{ij}$

3.1.2. Input parameters

k_{ij}	Number of processes in j th rout of part i
A	Intercellular movement unit cost
$T_{i(u_{ij}^{(r)})}$	The process time of part i on r^{th} machine from u_{ij} processing route
$TBF(m)$	Stochastic variable for time between failures for machine m
$MTBF_{(u_{ij}^{(r)})}$	The mean time between failures for r^{th} machine from u_{ij} processing route
$TTR(m)$	Stochastic variable for time to repair machine m
$MTTR(m)$	The mean time to repair machine m
$t_p(m)$	Total production time on machine m
$T_{i(u_{ij}^{(r)})}$	Process time for part i on machine $u_{ij}^{(r)}$
$B_{(u_{ij}^{(r)})}$	The hardware cost of r^{th} machine from u_{ij} processing route;
H	The man-hour cost of the maintenance labor force;
P_i	Production volume for part i
$B(m)$	Hardware breakdown Costs for machine m
V	Maximum number of machines in each cell
$r = 1, 2, \dots, k_{ij}$	the counter of machines in u_{ij} processing route;

NOTE1: $\{u_{ij}^{(1)}, u_{ij}^{(2)}, \dots, u_{ij}^{(r)}, \dots, u_{ij}^{(k_{ij})}\}$ is the index matrix of the machines required in u_{ij} processing route.

NOTE2: The machine associated with operation r^{th} through u_{ij} processing route might be k^{th} machine in the system, while the index k (machine number) acts as name of the machine.

3.1.3. Decision variables

$$x_{ijrl} = \begin{cases} 1 & \text{if } \text{processofroutejofpartiisallocatedtocelll} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{kl} = \begin{cases} 1 & \text{if } \text{if machine k is allocated to cell l} \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ij} = \begin{cases} 1 & \text{if } \text{processofroutejofpartiisallocatedtocelll} \\ 0 & \text{otherwise} \end{cases}$$

f_{ijrl}, g_{ijrl} : Auxiliary variables for linearization of intercellular transportation cost of parts

3.2. Reliability Considerations in Designing CMS

One of the most important features of CMS, which can improve this model, is the effect of breakdowns of the machine on the system. Conventional CMS models the design of systems which are vulnerable against breakdown of the machinery. The conventional models used to work with 100% reliability of the machinery in designing the cells and allocation of their tasks and components. However, the machinery may experience breakdown in practice. Breakdowns of the machines have the most significant effect on delivery times, even though the system represents a great flexibility. Taking into consideration the reliability of machinery would contribute to select more realistic processing routes in designing models of the manufacturing cells with multiple process routes. Very few studies have considered the effect of machine breakdown in cell formation problems (Saxena & Jain, 2011). Reliability of the manufacturing system is generally defined as the probability of desired performance of the system in a given time and under certain conditions (Saxena & Jain, 2011). The reliability costs may include hardware costs (spare parts) and labor costs.

3.3. Hardware Costs

The hardware costs of every breakdown including the shut downs as well as spare parts, resetting and restarting costs belonged to the machine m can be calculated as below.

$$BC(m) = \{(\text{Production demand} \times \text{Operation time} \times \text{unit breakdown cost of machine per breakdown}) / MTBF(m)\}$$

Where, TBF follows an exponential distribution with λ parameter:

$$TBF(m) \sim \exp(\lambda(m)) \quad (1)$$

A common technique to calculate $MTBF(m)$, which can be used during design phase of CMS can be extracted as follows:

$$MTBF(m) = 1/\lambda(m) \quad (2)$$

So, the number of breakdowns for the machine m during its operation time (t_p) can be calculated as below:

$$N(m, t_p) = t_p/MTBF(m) \quad (3)$$

While the breakdown costs of the machine m could be derived from this formula:

$$BC(m, t_p) = (t_p * B(m)/MTBF(m)) \tag{4}$$

3.4. Labor Costs

The following equation can be used to find the labor costs of every breakdown for the machine m :

$$HC(m) = \{(\text{Production demand} \times \text{Operation time} \times \text{unit human cost of machine per hour}) \times MTTR(m) / MTBF(m)\}$$

Assuming that $TTR(m)$ follows an exponential distribution, that is:

$$TTR(m) \sim \exp(\mu(m)) \tag{5}$$

A straightforward method to calculate $MTTR(m)$ is given below, where $\mu(m)$ is repairing rate of the machine m :

$$MTTR(m) = 1/\mu(m) \tag{6}$$

Having multiplied the number of failures $N(m, t_p)$ by the mean time to repair $MTTR(m)$, the total time allocated to repair during the manufacturing process would be obtained as below:

$$t_{rep}(m, t_p) = t_p * MTTR(m)/MTBF(m) \tag{7}$$

The total software (labor) cost associated with the system reliability will be obtained if the product above is multiplied in the per hour labor cost.

Reliability costs of the system (RC) are entered the model as below:

$$RC = \left[\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} Z_{ij} \left[B_{u_{ij}^{(r)}} \frac{P_i T_{i(u_{ij}^{(r)})}}{MTBF_{(u_{ij}^{(r)})}} + H \frac{P_i T_{i(u_{ij}^{(r)})} MTTR_{(u_{ij}^{(r)})}}{MTBF_{(u_{ij}^{(r)})}} \right] \right] \tag{8}$$

3.5. Calculation of Intercellular Transportation Costs in Objective Function

Consider the following equations:

$$IC = A \left[\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} \sum_{l=1}^c P_i Z_{ij} Y_{u_{ij}^{(r)}l} \left(1 - Y_{u_{ij}^{(r+1)}l} \right) \right] \tag{9}$$

IC represents a nonlinear objective function for minimizing the cost of intercellular flows, such that considering each pair of the existing consecutive machines in a given processing route, a number is counted once allocated in different cells and is then added to the total value which is obtained by multiplying total number of “one”s in the cost parameter A after counting the “one”s.

It was difficult to solve large scale problems with this nonlinear objective function, in spite of the few variables introduced to the model. Limitation in using nonlinear objective function for large problems is still deemed as an important concern. Thus, some studies have been directed in the past for linearization of this function.

3.5.1. Linearization of the proposed Intercellular Transportation Costs in Objective Function

An objective function very similar to the one in section 4-1 is reported in the papers of Jabal ameli et al. (2007; 2008) It is obvious that y and z are binary variables. They have suggested other binary variables in combination with a set of limitations for linearization. It should be noted that as mentioned in the results section, this kind of linearization needs a rather long time to be solved by optimization algorithms after programming due to the significantly great number of variables and limitations. It will additionally make programming of exact solution (Simplex) and meta-heuristic methods rather difficult. As a result, the problems with large dimensions would become almost unsolvable.

This paper proposes the following objective function for this problem by defining the decision variable based on allocation of the operations to the cells (and not the machines to the cells).

$$IC = \frac{A}{2} \left(\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} \sum_{l=1}^c |x_{ij,r+1,l} - x_{ijrl}| \right) \quad (10)$$

Eq. (10) counts the number of operations done in separate cells, which will be entered the main model as shown below using the variables g_{ijrl} and f_{ijrl} :

$$IC = \frac{A}{2} \left(\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} \sum_{l=1}^c (f_{ijrl} + g_{ijrl}) \right). \quad (11)$$

The limitations for linearization are represented as below:

$$x_{ijrl} \leq Y_{u_{ij}^{(r)}} \quad (12)$$

This limitation is for showing the relation between two decision variables $y_{u_{ij}^{(r)},l}$ and x_{ijrl} , once a process is allocated to that cell. The required machine must be allocated to that cell for processing the operation. Furthermore,

$$x_{ij,r+1,l} - x_{ijrl} = f_{ijrl} - g_{ijrl} \quad (13)$$

is used for linearization of the term containing absolute value in calculating the total costs of intercellular transportation.

3.6. proposed mixed integer linear model

In this section a mixed linear integer programming model is introduced for the problem by definition of the decision variable based on allocation of processes to cells:

$$\min \quad C = IC + RC \quad (14)$$

subject to

$$IC = \frac{A}{2} \left(\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} \sum_{l=1}^c (f_{ijrl} + g_{ijrl}) \right) \quad (15)$$

$$MC = \left[\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} Z_{ij} \left[B_{(u_{ij}^{(r)})} \frac{P_i T_i(u_{ij}^{(r)})}{MTBF_{(u_{ij}^{(r)})}} + H \frac{P_i T_i(u_{ij}^{(r)}) MTTR_{(u_{ij}^{(r)})}}{MTBF_{(u_{ij}^{(r)})}} \right] \right] \quad (16)$$

$$\sum_{l=1}^c x_{ijrl} = z_{ij} \quad , \quad i = 1,2, \dots, n \quad , \quad j = 1,2, \dots, q_i \quad , \quad r = 1,2, \dots, k_{ij} \quad (17)$$

$$x_{ijrl} \leq Y_{u_{ij}^{(r)}l} \quad , \quad i = 1, 2, \dots, n \quad , \quad j = 1, 2, \dots, q_i \quad , \quad r = 1, 2, \dots, k_{ij} \quad , \quad l = 1, 2, \dots, c \quad (18)$$

$$x_{ij,r+1,l} - x_{ijrl} = f_{ijrl} - g_{ijrl} \quad , \quad i = 1, 2, \dots, n \quad , \quad j = 1, 2, \dots, q_i \quad , \quad r = 1, 2, \dots, k_{ij} - 1 \quad , \quad l = 1, 2, \dots, c \quad (19)$$

$$\sum_{k=1}^p Y_{kl} \leq V \quad , \quad l = 1, 2, \dots, c \quad (20)$$

$$\sum_{l=1}^c Y_{kl} = 1 \quad , \quad k = 1, 2, \dots, p \quad (21)$$

$$\sum_{j=1}^{q_i} Z_{ij} = 1 \quad , \quad i = 1, 2, \dots, n \quad (22)$$

$$z_{ij}, y_{kl} \in \{0, 1\} \quad , \quad i = 1, 2, \dots, n \quad , \quad j = 1, 2, \dots, q_i \quad ; \quad k = 1, 2, \dots, p \quad ; \quad l = 1, 2, \dots, c \quad (23)$$

$$x_{ijrl} \geq 0 \quad , \quad i = 1, 2, \dots, n \quad , \quad j = 1, 2, \dots, q_i \quad , \quad r = 1, 2, \dots, k_{ij} \quad ; \quad l = 1, 2, \dots, c \quad (24)$$

As previously discussed, in this model the objective function is the sum of intercellular transportation cost of the materials, and repair and maintenance cost of the machinery. Eq. (15) calculates the total intercellular transportation cost of the material, which is in fact a linearized version of the nonlinear Eq. (25) below.

$$IC = \frac{A}{2} \left(\sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{r=1}^{k_{ij}-1} \sum_{l=1}^c p_i |x_{ij,r+1,l} - x_{ijrl}| \right) \quad (25)$$

Eq. (16) calculates the total repair and maintenance costs of the machinery, in which the first term gives the hardware cost of repair and maintenance whereas the second term provides the labor force cost required for it. Constraint (17) represents the relation between decision variables Z_{ij} and X_{ijrl} . It further ensures that: (a) the r^{th} process of manufacturing route j from part i is only allowed when the manufacturing route j for the part i is selected, and (b) the r^{th} process of manufacturing route j from part i must be allocated only to one cell if the manufacturing route j is selected for the part i . Constraint (18) demonstrates the relation between decision variables $y_{u_{ij}^{(r)}l}$ and X_{ijrl} . It additionally states that the

machine needed for processing that operation must be also allocated to that cell if an operation is allocated to a cell. Constraint (19) is employed to linearize the term containing absolute value in calculation of the total intercellular transportation costs of the materials (Eq. (25)). Constraint (20) controls the number of machinery for each cell and sets upper limit of V for this number. Constraint (21) indicates that each machine must be allocated to one cell, while constraint (22) guarantees that only one manufacturing route is selected for each part. Constraints (23) and (24) are determinant for the type of decision variables. As previously discussed in section 4, there are other linear models in the literature for this problem or other similar problems. A pure 0-1 linear integer model is proposed in these models using the nonlinear model introduced in section 4-1 by Jabal Ameli et al. (2007). This model influences the great number of variables and its performance constraints. Table 1 illustrates a comparison between the nonlinear model developed in Section 4-1, the absolute 0-1 linear integer model proposed in Section 4-2, and the mixed linear integer model suggested by this paper in Section 4-3, where Q denotes the total number of existing manufacturing routes, R denotes the total number of existing processes in these routes where $Q = \sum_{i=1}^n q_i$ are $R = \sum_{i=1}^n \sum_{j=1}^{q_i} k_{ij}$, respectively. It is observed from table 1 that the numbers of continuous variables and constraints of the proposed model are considerably smaller than that of the Jabal Ameli & Arkat's model (2008) (except for the case with 3 or less machines, which is not practically applicable). Therefore, solving their release model (linear, Lagrangian or etc.) becomes more extensive in large scale problems. As a result, the proposed model will be both faster and more efficient:

Table 1
Comparison between proposed model and other existing models

Model	Number of variables			Number of constraints
	0-1	continuous	total	
NILP	Q+MC	0	Q+MC	N+M+C
pure ILP (literature)	Q+MC	RMC	Q+MC+RMC	N+M+C+4RMC
proposed MILP	Q+MC	3RC	Q+MC+3RC	N+M+C+R+RC+(R-Q)C

$$R = \sum_{i=1}^n \sum_{j=1}^{q_i} k_{ij}, Q = \sum_{i=1}^n q_i, M: \text{number of machines, } C: \text{number of cells, } N: \text{number of parts}$$

4. Two-Layer Genetic Algorithm

Genetic Algorithm (GA), introduced by Holland (1975) and developed by Goldberg (1989) is a widespread, parallel, stochastic search and optimization method that has been employed for solving numerical optimization problems in a wide variety of application fields including engineering, biology, economics, agriculture, business, telecommunications, and manufacturing (Goldberg, 1989; Gen and Cheng, 1997; Man et al., 1999; Onwubolu & Mutingi, 2001). It is a model of machine learning which derives its behaviour from a metaphor of the processes of evolution in nature. GA starts with encoding the solutions in a chromosome structure containing several genes and then, an initial population of chromosomes will be formed as the initial solutions for the problem. Afterwards, the superior chromosomes are selected by the selection operator for the new mating pool. The evolution is simulated by using reproduction operators such as crossover operator, imitating propagation, and mutation operator, imitating random changes occurring on chromosomes in nature. Reproduction operators are employed on the mating pool to generate new solutions, called off springs. Then the replacement operator chooses some chromosomes among former generation and off springs to form new generation. This procedure will be continued until the termination criterion is met.

Because of the efficient and multi-direction searching capability of GA using a population of solutions instead of single solution, its simple and understandable nature, its ability in preventing from getting trapped in local optimum and its successful applications in designing manufacturing systems, it has been used and a bi-level hybrid genetic algorithm (HGA) has been developed here in order to solve the cell formation and routing selection problems simultaneously considering the reliability issues. Information like operation sequence of routes nominated for manufacturing the products, processing time of operations, production volumes, alternative routes of parts, maximum size of cells, number of cells, unit cost of intercellular material handling, unit cost of repair and maintenance and its required labor force are used in the proposed algorithm.

The first level of this algorithm seeks to find the best manufacturing route for the parts, while the second level tries to find the best configuration of the cells based on the manufacturing routes that are selected at the first level. There is a relationship between these two levels which implies that when the first level's search is implemented to get a better combination of the manufacturing routes, after meeting the termination criterion of this level, the search process is transferred to the second level in order to address the best configuration of the cells using the solution obtained from the first level. Meanwhile, the search process is transferred to the first level once the termination criterion of the second level is met. This process will be continued until the termination criterion of the algorithm is met. The proposed algorithm explores the search space adequately using mutation operator, elimination of repetitive chromosomes and acceptance of worse solutions. Besides, it exploits the best solutions using roulette wheel selection, crossover operator and elitism technique. As a result of improved exploration and exploitation, the algorithm can hopefully achieve the optimum or near-optimum solutions. The components of the proposed GA and its operators are explained in the following subsections.

4.1. Notation used in the proposed HGA

$popSize$	Population size
f_i	The objective function of chromosome i
F_i	The fitness of chromosome i
$best_fitt$	The fitness of best chromosome in the current population
$best_partial_fitt$	The best partial fitness found so far
$best_overall_fitt$	The best fitness found so far
$MaxNoIMP1$	Number of consecutive iterations with no fitness improvement for terminating first level of algorithm
$MaxNoIMP2$	Number of consecutive iterations with no fitness improvement for terminating second level of algorithm
$NoIMP$	Number of consecutive iterations with no fitness improvement so far
$MaxTime$	The runtime of algorithm
μ_c	Crossover rate
μ_m	Mutation rate

4.2. Encoding Scheme

Each chromosome is a string of size $N + M$ and indicates a feasible solution of the search space. It is composed of two parts. The first one represents a solution for routing selection problem and contains N genes in which, the allele (value of a characteristic) located in the i^{th} locus (position of a characteristic in the chromosome) demonstrates the selected manufacturing route for part type i . The second part of the chromosome shows a solution for cell formation problem and is composed of M genes, where the allele located in the k^{th} locus indicates the cell selected for machine k . Fig. 1 depicts a chromosome for a problem with 3 parts, 13 machines, 3 cells, 2 process routes for each part and maximum cell size of 5.

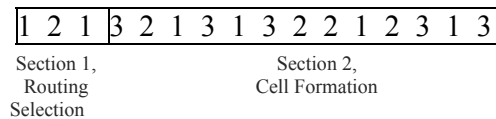


Fig. 1. Configuration of a chromosome

4.3. Fitness Function Evaluation

The fitness value is a measure to determine the quality level of a chromosome. The fitness values are used to select parent chromosomes to create the next generation. A certain transformation is required to use the objective function values as the fitness values. Since the objective function of the considered problem is the minimization of total cost, transformation can be carried out by Eq. (26).

$$F_i = BIGM - f_i \quad (26)$$

where BIGM is a big positive number. Therefore, the better a chromosome's fitness is, the better its quality will be.

4.4. Selection Operator

The roulette wheel selection procedure has been used in the proposed HGA. The purpose of this selection operator is to allow the fittest individuals to be considered more often to reproduce children for the next generation. Each individual is assigned a probability of being selected based on its fitness value. Although better individuals will have a higher selection probability, all individuals in the population will have a chance to be selected.

4.5. Crossover Operator

In the GAs, each pair of selected parents can generate two children using crossover operator. After two parent chromosomes have been selected for recombination, with a probability of μ_c , the parents are recombined and two new chromosomes are generated as off springs and with a probability of $1 - \mu_c$, no crossover occurs and both parents pass unchanged as off springs. In the proposed algorithm, uniform crossover operator is used for the first and the second parts of the chromosomes. For the first part, a binary mask of size N is randomly generated and thereby, the first (second) child will receive its i th gene from the first (second) parent if the i th element of the binary mask is equal to 1 and from the second (first) parent if it is equal to 0. For the second part of chromosomes, a binary mask of size M is randomly generated, and the same procedure is applied. However, applying this crossover to the second part may lead no feasible offsprings because of violating the constraint (7) of the mathematical model. In this situation, modification operator must be applied. Fig. 2 illustrates how the crossover operator works.

Parent 1	1 2 1	3 2 1 3 1 3 2 2 1 2 3 1 3
Parent 2	2 2 2	1 3 2 1 2 3 1 2 3 1 2 1 2
Binary Mask	0 0 1	0 1 1 1 0 1 0 0 1 1 0 1 0
Offspring 1	2 2 1	1 2 1 3 2 3 1 2 1 2 2 1 2
Offspring 2	1 2 2	3 3 2 1 1 3 2 2 3 1 3 1 3

Fig. 2. Crossover operator

4.6. Mutation Operator

For better exploration of the search space and prevention of getting trapped in local optimum, at last one gene of the first part and at last two genes of the second part of the generated off springs are mutated with a probability of μ_m . The mutation operator for the first part of chromosome is such that if a gene is chosen for being mutated, that gene will take a value different from its current value (the manufacturing route is actually changed for the part corresponding to that gene). For the second part of the chromosome assume that the k th gene is selected for mutation. In this case, a gene with different value is randomly selected and then these two genes swap their values (the cell of two machines are actually swapped). It should be noted that, applying the mutation operator never leads to infeasible solutions.

4.7. Modification Operator

As mentioned before, crossover of the second part of chromosomes may lead no feasible offsprings. In this situation, the infeasible offsprings must be modified. In the modification schema, for each cell of each offspring, the number of machines allocated to that cell is counted and if this number exceeds the maximum cell size, the additional machines will be randomly allocated to the cells which have unfilled places. Fig. 3 illustrates how the modification operator works (the maximum cell size is equal to 5 here):

After crossover	Offspring 1	2 2 1	1 2 1 3 2 3 1 2 1 2 2 1 2
	Offspring 2	1 2 2	3 3 2 1 1 3 2 2 3 1 3 1 3
After modification	Offspring 1	2 2 1	1 2 1 3 2 3 1 2 1 3 2 1 2
	Offspring 2	1 2 2	3 3 2 1 1 3 2 2 3 1 3 1 1

Fig. 3. Modification operator

4.8. Replacement Operator and Elitism

In the proposed HGA, the population at a given generation $k+1$ is obtained by replacement operator. This operator selects the best individuals among generation k and its generated offsprings and forms the generation $k+1$. Since all previous and current best population members are included in generation $k+1$, elitism is ensured.

4.9. Acceptance of Worse Solutions

Genetic algorithms often use mutation operator for prevention of getting trapped in local optimum. To ensure further exploration of the search space, a mechanism similar to simulated annealing (SA) algorithm has been used in the proposed algorithm. The only difference is that, unlike SA, in the developed algorithm, after a quarter of algorithm's runtime, the worse solutions are accepted with constant probability of 5%. In other words, the proposed algorithm lists all the chromosomes with worse fitness than $best_fitt$ and then accepts one of them as $best_partial_fitt$. Afterwards, all individuals with better fitness than $best_partial_fitt$ are replaced with random chromosomes.

4.10. Elimination of Repetitive Solutions

Since genetic algorithm is based on population and selection of the chromosomes as the parents is done here based on their fitness values, after some iterations, several repetitive solutions may exist in the current population. This will significantly reinforce the probability of selecting two similar chromosomes as the parents by increasing the number of iterations. Moreover, off springs similar to the previous population will be created and probably, the algorithm will get trapped in local optimum. To avoid this, completely similar chromosomes are eliminated from the population, after each iteration, and they are replaced with random chromosomes.

4.11. Proposed Hybrid Genetic Algorithm

Proper determination of the parameters for each algorithm significantly influences its convergence rate and the quality of the obtained solutions. To determine the value of these parameters, one must always evaluate and compare the quality of the obtained solutions as well as the computational efficiency of the algorithm. In the proposed HGA, some parameters like size of the population, termination criteria, crossover rate and mutation rate have been addressed by implementing several tests and comparing quality of the obtained solutions and convergence rate of the algorithm. The candidate values and the selected value of the parameters in the proposed algorithm are as follows:

$popSize$	20 (candidate options: 20, 30, 40 and 50)
$MaxNoIMP1$	Q (candidate options: $\frac{Q}{2}$, Q , $\frac{3Q}{2}$, $2Q$)
$MaxNoIMP2$	$\frac{3M}{2}$ (candidate options: $\frac{M}{2}$, M , $\frac{3M}{2}$, $2M$)
$MaxTime$	The algorithm converges within 10-1000 seconds according to the size of the problem
μ_c	0.99 (candidate options: 0.90, 0.95, 0.97, 0.99)
μ_m	0.05 (candidate options: 0.01, 0.02, 0.05, 0.10)

Fig. 4 and 5 shows the flowcharts of the proposed bi-level HGA.

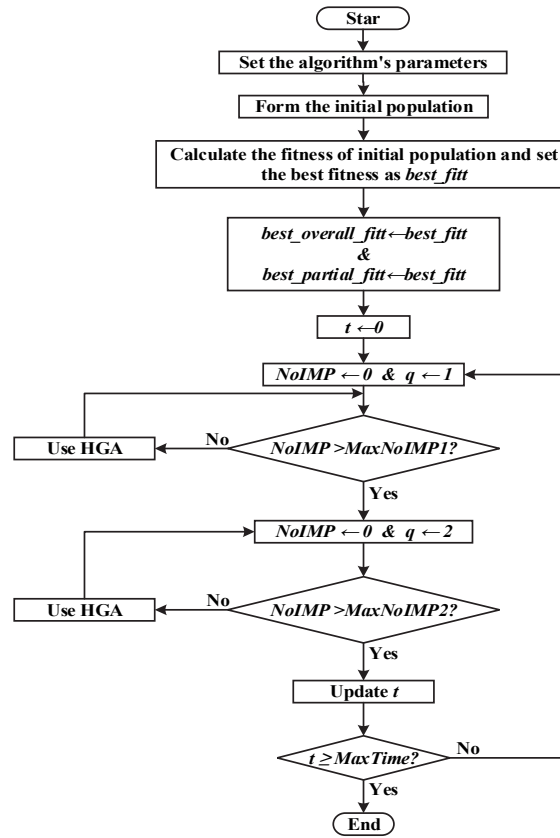


Fig. 4. Flowchart of the bi-level HGA

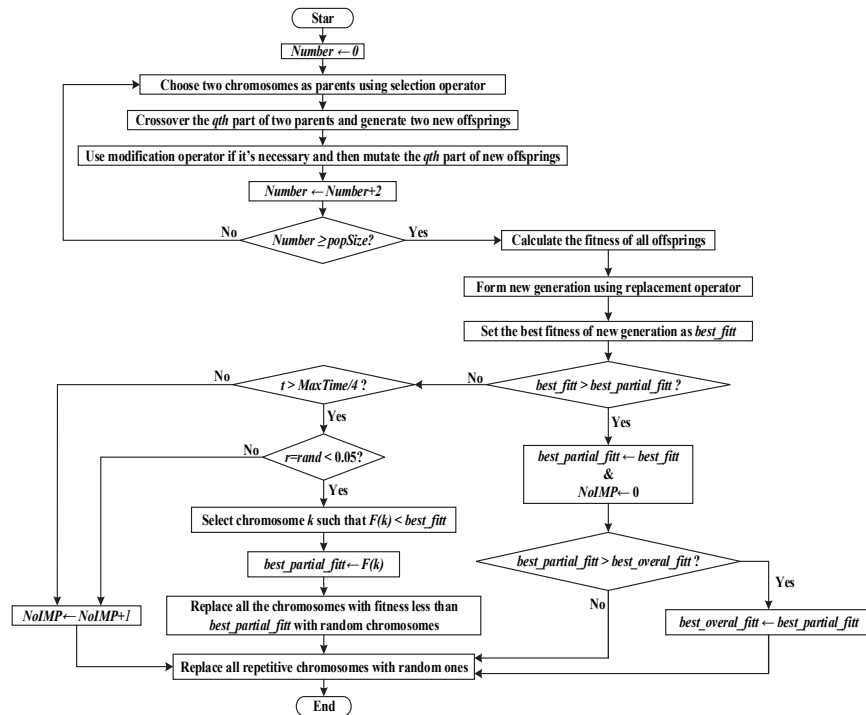


Fig. 5. Flowchart of HGA for the first and second parts of chromosomes

5. Results of Calculations

5.1. Performance Results and Evaluation of Proposed Linear Model

Some 13 numerical examples are used to assess and demonstrate efficiency of the linear model and mixed genetic algorithm developed. All the problems except for problems 2 and 3 are problems with random numbers. Problems 2 and 3 are case studies and their parameters are obtained from space of a mean manufacturing system (parameters and solution for this problem can be found in appendix (A)). Table 2 summarizes the results obtained from solving the numerical examples by the proposed model and also the existing models in the literature through exact method of branch & bound using LINGO 8 software.

Table 2

Solution of numerical examples with the proposed model and existing models

Problem No.	Parameters				INLP		pure ILP (literature)		proposed ILP	
	N	Q	M	C	OFV	time	OFV	time	OFV	time
1	3	5	9	3	22463 *	00:00:05	22463 *	00:00:32	22463.81 *	00:00:02
2	3	9	22	3	22573.13 *	00:02:14	25073.13 *	00:32:27	22573.13 *	00:00:30
3	3	12	22	3	22330.97 *	00:24:55	22330.97 *	01:43:08	22330.97 *	00:01:59
4	3	6	13	4	46867.32 *	00:02:29	46867.32 *	00:51:00	46867.32 *	00:00:42
5	5	10	8	2	83800.6 *	00:00:03	83800.6 *	00:00:11	83800.57 *	00:00:02
6	5	10	15	3	68893.7 *	00:01:53	68893.7 *	01:23:00	68893.7 *	00:00:41
7	5	15	16	3	47465.6 #	>7:00:00	47465.6 #	>7:00:00	39606.7 *	0.002095
8	8	16	15	3	169481.19 #	>7:00:00	170241 #	>7:00:00	166545.47 *	0.004086
9	8	18	20	4	315718.45 #	>7:00:00	114843 \$	>7:00:00	241462.62 *	0.080625
10	8	22	22	4	231471.34 #	>7:00:00	66772.4 \$	>7:00:00	168032.63 *	0.225914
11	10	20	20	4	293536.26 #	>7:00:00	144491 \$	>7:00:00	278148 *	0.090787
12	10	25	27	5	97603.7 \$	>7:00:00	97603.7 \$	>7:00:00	250004 #	>7:00:00
13	10	29	43	6	68023.14 \$	>7:00:00	out of memory		158806 \$	>7:00:00

* optimal solution was found

best objective(not optimal) was found in predetermined time

\$ only best lower bound was found

The obtained results were global optimum answers of the problems, which show that the proposed MILP model is practically more efficient than other existing models. These answers are being used in the next section for evaluation of the developed genetic algorithm.

5.2. Performance Results and Evaluation of Proposed Genetic Algorithm

Table 3 shows the obtained results from solving the numerical examples with the proposed genetic algorithm. The obtained results demonstrate that the proposed algorithm is able to give good answers for the problems with various sizes in a reasonable time. Taking into consideration these results, it can be seen that the proposed algorithm has attained to an optimal answer in 93.64% of the cases. Meanwhile, mean error percentage obtained from optimal value in the worst case was 0.87%, with error content of the worst answer from optimal value being equal to 2.77% in the worst case. Computational cost of the proposed algorithm is also much smaller than that of MILP model suggested here. Therefore, the algorithm would take just a very little time to reach a good answer (the algorithm reaches a good answer at every quarter of its running time). However, most of the time of the algorithm is assigned to implement techniques to escape from local optimum, do more improved search within the search space and try to reach a better possible answer.

Table 3
Solution of numerical examples with the suggested genetic algorithm

Problem number	Parameters				GA (number of runs:10)							
	N	Q	M	C	Optimum objective	best obtained objective	number of optimum solution	mean obtained objective	worst obtained objective	time	Error percentage of mean objective from optimum (%)	Error percentage of worst objective from optimum (%)
1	3	5	9	3	22463.8	22463.8	10	22463.8	22463.8	00:00:10	0.00	0.00
2	3	6	22	3	22573.1	22573.1	9	22736.3	24402.3	00:00:30	0.72	8.10
3	3	12	22	3	22331.0	22331.0	10	22331.0	22331.0	00:00:30	0.00	0.00
4	3	6	13	4	46867.3	46867.3	10	46867.3	46867.3	00:00:20	0.00	0.00
5	5	10	8	2	83800.6	83800.6	10	83800.6	83800.6	00:00:10	0.00	0.00
6	5	10	15	3	68893.7	68893.7	10	68893.7	68893.7	00:00:30	0.00	0.00
7	5	15	16	3	39606.7	39606.7	7	39953.0	40704.1	00:00:30	0.87	2.77
8	8	16	15	3	166545.5	166545.5	10	166545.5	166545.5	00:00:30	0.00	0.00
9	8	18	20	4	241462.6	241462.6	9	241756.6	244402.2	00:00:40	0.12	1.22
10	8	22	22	4	168032.6	168032.6	9	168106.4	168769.7	00:00:40	0.04	0.44
11	10	20	20	4	278148.0	278148.0	9	278195.5	278623.5	00:00:45	0.02	0.17
12	10	25	27	5		247803.7	3	249969.9	253317.6	00:01:00		
13	10	29	43	6		412025.3	7	416035.3	437025.3	00:01:20		

Fig. 6 depicts search process of the algorithm for the problem number 2 with 3 parts, 6 manufacturing routes, 22 machines and 3 cells within 30 seconds. It is observed that the algorithm has reached an optimal answer at 1115th repetition. From 1116th to 3823rd repetition, it is tried to search the answer space more and further ensure about prevention of getting trapped in the local optimum.

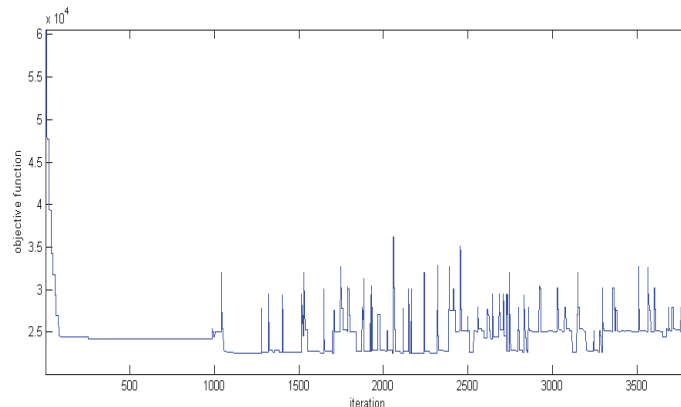


Fig. 6. Search process of the algorithm for the problem number

6. Conclusion

We proposed a mixed linear integer numerical model for designing the manufacturing cells in order to reduce the manufacturing costs, increase quality of the products and enhance reliability of the system. Since this model is mainly developed to deal with a natural issue, it has some characteristics which make them more practical in comparison with the other existing models. For instance, some of the models which consider the reliability in a CF problem have suggested multi-objective models. Thereby, the main purposes are to improve the reliability and reduce the manufacturing costs. Calculating the reliability value is difficult for the system and rather useless for the system, so solving this model will be very difficult. A logical relation is used in this paper such that a cost approach can also enhance the reliability value. However, it is not calculated in this numerical method, since the reliability of this system is improved by decreasing the number of breakdowns during system operation, while the repair and

maintenance costs are reduced as well. The little number of limitations and integer variables, and also linear behavior of the model have all added to its efficiency for both small and the large scale problems. This model has been solved for some ranges of small to large scale problems using LINGO software with its improved efficiency being shown in Table 2. A multi-layer genetic algorithm is recommended to be developed to facilitate solving all kinds of problems and especially the large scale one. Efficiency of this algorithm is illustrated in Table 2 for the both small and large scale problems.

References

- Agarwal, A., & Sarkis, J.A. (1998). Review and analysis of comparative performance studies of functional and cellular manufacturing layouts. *Computers and Industrial Engineering*, 34(1) 77–89.
- Ahkioon, S., Bulgak, A. A., & Bektas, T. (2009). Cellular manufacturing systems design with routing flexibility, machine procurement, production planning and dynamic system reconfiguration. *International Journal of Production Research*, 47(6), 1573-1600.
- Akturk, M. S., & Turkcan, A. (2000). Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research*, 38(10), 2327-2347.
- Ameli, M. S. J., Arkat, J., & Barzinpour, F. (2008). Modelling the effects of machine breakdowns in the generalized cell formation problem. *The International Journal of Advanced Manufacturing Technology*, 39(7-8), 838-850.
- Ameli, M. S. J., & Arkat, J. (2008). Cell formation with alternative process routings and machine reliability consideration. *The International Journal of Advanced Manufacturing Technology*, 35(7-8), 761-768.
- Askin, R.G., & Estrada, S. (1999). Investigation of cellular manufacturing practices, S.A. (Ed.) *Handbook of Cellular Manufacturing Systems (Chapter1)* John Wiley, New York, 25-34.
- Boughton, N.J., & Arokiam, I.C. (2000). The application of the cellular manufacturing: a regional small to medium enterprise perspective. *Proceedings of the Institution of the Mechanical Engineers*, 214(Part B), 751–754.
- Bulgak, A. A., & Bektas, T. (2009). Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration. *European Journal of Operational Research*, 192(2), 414-428.
- Das, K., Lashkari, R. S., & Sengupta, S. (2007). Machine reliability and preventive maintenance planning for cellular manufacturing systems. *European Journal of Operational Research*, 183(1), 162-180.
- Das, K., Lashkari, R. S., & Sengupta, S. (2007). Reliability consideration in the design and analysis of cellular manufacturing systems. *International Journal of Production Economics*, 105(1), 243-262.
- Defersha, F. M., & Chen, M. (2006). A comprehensive mathematical model for the design of cellular manufacturing systems. *International Journal of Production Economics*, 103(2), 767-783.
- Defersha, F. M., & Chen, M. (2006). Machine cell formation using a mathematical model and a genetic-algorithm-based heuristic. *International Journal of Production Research*, 44(12), 2421-2444.
- Defersha, F. M., & Chen, M. (2008). A parallel genetic algorithm for dynamic cell formation in cellular manufacturing systems. *International Journal of Production Research*, 46(22), 6389-6413.
- Defersha, F. M., & Chen, M. (2008). A parallel multiple Markov chain simulated annealing for multi-period manufacturing cell formation problems. *The International Journal of Advanced Manufacturing Technology*, 37(1-2), 140-156.
- Flynn, B. B., & Robert Jacobs, F. (1986). A simulation comparison of group technology with traditional job shop manufacturing. *International Journal of Production Research*, 24(5), 1171-1192.
- Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. Wiley, MA.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search Optimization & Machine Learning*. Addison Wesley.
- Gupta, Y., Gupta, M., Kumar, A., & Sundaram, C. (1996). A genetic algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research*, 34(2), 447-482.

- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.
- Kusiak, A. (1987). The generalized group technology concept. *International Journal of Production Research*, 25(4), 561-569.
- Lokesh, K., & Jain, P.K. (2008). Part-machine group formation with operation sequence, time and production volume. *International Journal of Simulation Modelling*, 7(4), 198–209.
- Lokesh, K., & Jain, P.K. (2009). Part-Machine group formation with ordinal-ratio level data and production volume. *International Journal of Simulation Modelling*, 8(2), 90–101.
- Lokesh, K., & Jain, P.K. (2010). Concurrently part-machine group formation with important production data. *International Journal of Simulation Modelling*, 9(1), 5–16.
- Lokesh, K., & Jain, P.K. (2010). Dynamic cellular manufacturing systems design—a comprehensive model & HHGA. *Advances in Production Engineering & Management Journal*, 5(3), 151–162.
- Man, K.F., Tang, K.S., & Kwong, S. (1999). *Genetic algorithms: concepts and design*. Springer, London.
- Morris, S.J., & Tersine, R.J. (1990). A simulation analysis of factors influencing the attractiveness of group technology cellular layouts. *Management Science*, 36(12), 1567–1578.
- Nsakanda, A. L., Diaby, M., & Price, W. L. (2006). Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research*, 171(3), 1051-1070.
- Onwubolu, G. C., & Mutingi, M. (2001). A genetic algorithm approach to cellular manufacturing systems. *Computers & industrial engineering*, 39(1), 125-144.
- Ouk Kim, C., Baek, J. G., & Baek, J. K. (2004). A two-phase heuristic algorithm for cell formation problems considering alternative part routes and machine sequences. *International Journal of Production Research*, 42(18), 3911-3927.
- Saxena, L. K., & Jain, P. K. (2011). Dynamic cellular manufacturing systems design—a comprehensive model. *The International Journal of Advanced Manufacturing Technology*, 53(1-4), 11-34.
- Seifoddini, H., & Djassemi, M. (2001). The effect of reliability consideration on the application of quality index. *Computers & Industrial Engineering*, 40(1), 65-77.
- Sofianopoulou, S. (1999). Manufacturing cells design with alternative process plans and/or replicate machines. *International Journal of Production Research*, 37(3), 707-720.
- Tompkins, J.A., White, J.A., Bozer, Y.A., & Tanchoco J.M.A. (2003). *Facility planning*. Wiley, New York.
- Wemmerlov, U., 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., & Johnson, D. J. (1997). Cellular manufacturing at 46 user plants: implementation experiences and performance improvements. *International journal of production research*, 35(1), 29-49.
- Zhao, C., & Wu, Z. (2000). A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, 38(2), 385–395.

Appendix (A)

Data for example 3

Machine	Encoding(K parameter)	Machine	Encoding(K parameter)
TC-201	1	MU-202	12
TC-202	2	MU-203	13
TC-203	3	MU-204	14
TC-204	4	MU-205	15
MC-206	5	MU-206	16
MC-208	6	DR-210	17
MC-201	7	DR-201	18
DR-206	8	MU-207	19
GC-202	9	DR-205	20
TU-201	10	MU-101	21
MU-201	11	MU-102	22

P.N	P.V	R.N	P.S	P.T
1	2500	1	5 10 10 2 20 15 14 12 4 7 8 8 9 8 17	24 1.83 1.5 5.75 0.92 2.5 3 1.83 3 5.5 1.5 1.5 2 1.5 3
		2	6 10 10 2 20 19 14 12 4 7 8 8 9 8 17	26 1.83 1.5 5.75 0.92 2.4 3 1.83 3 5.5 1.5 1.5 2 1.5 3
2	2500	1	11 16 16 17 17 20 11 18 18 22 21 21 20	2.5 2 2 2.5 2 1 2 1.5 1.5 2 1.5 3 2.5
		2	11 15 15 17 17 20 11 18 18 22 21 21 20	2.5 2.1 2.1 2.5 2 1 2 1.5 1.5 2 1.5 3 2.5
3	6000	1	6 10 10 1 14 3 13 12 9 8	24 1.2 1.1 5 4.3 3 3.5 1.7 2 1.5
		2	5 10 10 1 13 3 13 12 9 8	26 1.2 1.1 5 4.5 3 3.5 1.7 2 1.5

P.N: part no.

P.V: production volume

R.N: routing no.

P.S: production sequence

P.T: production time

Scheduled production plan

P.N	R.N	P.S	P.T
1	1	5 10 10 2 20 15 14 12 4 7 8 8 9 8 17	24 1.83 1.5 5.75 0.92 2.5 3 1.83 3 5.5 1.5 1.5 2 1.5 3
2	1	11 16 16 17 17 20 11 18 18 22 21 21 20	2.5 2 2 2.5 2 1 2 1.5 1.5 2 1.5 3 2.5
3	2	5 10 10 1 13 3 13 12 9 8	26 1.2 1.1 5 4.5 3 3.5 1.7 2 1.5

Machines in each cell (Design of cells)

Cell 1	Cell 2	Cell 3
TC-202	TC-204	TC-201
MU-201	MC-208	TC-203
MU-206	MC-201	MC-206
DR-210	MU-204	DR-206
DR-201	MU-205	GC-202
DR-205	MU-207	TU-201
MU-101		MU-202
MU-102		MU-203



© 2016 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).