

Integrating Strategic and Tactical Decisions to Robust Designing of Cellular Manufacturing under Uncertainty: Fixed Suppliers in Supply Chain

V.R Ghezavati*

Industrial Engineering Faculty Board Member, Islamic Azad University, South Tehran Branch, Tehran, Iran

S.J. Sadjadi

Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

M. Dehghan Nayeri

University of Tehran, Tehran, Iran

Received 17 April 2010

Accepted 30 March 2011

Abstract

Cellular manufacturing (CM) has been identified as an innovative practice for manufacturer to achieve efficiency as well as flexibility under an uncertain environment. This study addresses a new mathematical robust model for a cellular manufacturing problem integrated with tactical aspects under supply chain network characteristics in the presence of uncertain internal parameter (processing times) and external parameter (demands). The model aims to minimize total cost consisting expected value and variance of tactical cost (scheduling), strategic cost (employment of suppliers in supply chain network) to produce exceptional elements as well as the cost of resource underutilization. The model can give up a series of the solutions that are less sensitive to the changes in input data. To solve the robust model, an efficient hybrid method based on new combination of genetic algorithm (GA), simulated annealing (SA) in a parallel structure are proposed under an optimization rule. The effectiveness of the proposed model is demonstrated by numerical results and the trade-off between solution robustness and model robustness is also analyzed.

Keywords: Cellular manufacturing, group scheduling, Robust optimization, Uncertain demand and processing time, Parallel solution method.

1. Introduction and literature survey

Group technology (GT) is a management concept that aims to group products with similar process or manufacturing characteristics, or both. Cellular manufacturing system (CMS) can be proposed as a practical application of GT that determines the groups of machines based on similarity of the products processed by them. The traditional purpose of CMS is to

identify machine cells and part families concurrently and to assign part families to machine cells in order to minimize the intercellular and intracellular costs of parts. Aggregating traditional considerations with newly ones such as scheduling, stochastic approaches, processing time, variable demand, sequencing, and layout consideration to the traditional aspects can be more practical. Scheduling parts in individual cells is an

* Corresponding author: Industrial Engineering Faculty Board Member, Islamic Azad University, South Tehran Branch, Tehran, Iran email: V_ghezavati@azad.ac.ir and vrghezavati@gmail.com

operational feature that should be determined at the design stage.

In any manufacturing environment, suppliers or supply chain network structures have a vital role in order to produce different products. A manufacturer can decide whether the operations are completed inside the manufacturing system or they have to be outsourced to the suppliers in a supply chain network. Thus, suppliers can affect on the characteristics of operation processes which are not completed inside the system. Once there are multi suppliers for each product, decision maker have to select which supplier is more suitable or is the best one for subcontracting products based on the different criterions such as transportation, inventory policies and location of warehouses. Thus, for each product one of the two types of costs will be arisen: tactical cost such as scheduling or production planning due to produce inside the system and subcontracting cost due to employment suppliers under a supply chain network. By optimizing this problem, in fact supplier network design will be optimized across a supply chain. Therefore, integration of tactical and strategic decisions in a single problem must have high priority for researchers.

In this paper, we aim to optimize formation of work cells according to the optimization of tactical decisions (scheduling) and strategic decisions (supplier considerations) under an uncertain environment.

This survey highlights studies that are relevant to the uncertainty planning of CMS problems; however, a survey of certain conditions will be presented.

In the literature correspondence to CMS problems, uncertainty has been considered in different forms. We have classified previous researches into different groups, which would be discussed later.

Group 1) Uncertainty appears in products' demand and products' mix. In this group, there are two approaches employed in order to resolve uncertainty, which are stochastic and fuzzy theory. There are also some cases where demand is aggregated with tactical aspects such as production planning in stochastic form [1], layout problem [2] or dynamic and multi period conditions [3]. There are also cases where, in other studies, uncertainty in products' demand is treated using fuzzy numbers [4].

Group 2) The other group is associated with the formulation of CMS problems using fuzzy coefficients in the objective function and constraints [5].

Group 3) Another group considers uncertainty with processing times of products where there are some efforts applied for mathematical programming and using fuzzy approaches to obtain expected results. There are

also cases where significant attempts are devoted to implement heuristic approaches [6] and [7]. Other types of studies formulated the problem as a queue network and then analyzed it by queuing theory [8].

Group 4) Uncertainty occurs for different reasons such as fluctuations in design aspects during production process. Since, fluctuations in design aspects are not certain events, so uncertainty can be formulated by a set of possible scenarios in the future. In this way, some studies applied interval coefficient to resolve uncertainty [9].

Group 5) There are also some other cases where uncertainty is considered in resource availability of production equipments where cell formation (CF) problems can be formulated by applying probability theory [10] and [11]. In addition, multi processing routes is another alternative to be substituted once a machine is faced with failure [12, 13].

Group 6) Uncertainty can also be recognized in similarity coefficients. For example, a new similarity coefficient has been introduced where applied fuzzy theory and then transformed it to a binary matrix [14].

Group 7) Capacity level of machines is considered to be uncertain. Since this critical parameter plays an important role to determine bottleneck machine, thus it is vital to make flexible decisions under any realization of this parameter [15].

Group 8) Finally, uncertainty in CMS problem can also be detected in products arrival time to cells. Classical models assume that all products are available at the beginning of the production planning while in real application we may have some cases where products arrive to cell with unknown time which could be studied using queuing theory [16]. The cases where a CMS is involved with cell formation integrated with scheduling ([17], [18]), exceptional elements in CF ([19], [20]), the implementation of meta-heuristics and heuristics to solve large-scale problems are more practical and appealing real-case problems ([21], [22]).

In order to indicate strengths and comparison of our work against previous researches, the following descriptions are discussed.

1- A huge portion of investigations regarded to the CMS problem in uncertain situations can be divided into three groups: (a) fuzzy theory, (b) stochastic optimization, and (c) heuristic procedures.

1a) The most applied approach to resolve uncertainty in CMS problems in the literature can be introduced fuzzy approach [5], [23], [9].

1b) Also, most of the stochastic papers analyzed problem actually in certain condition where cases are transformed into the certain problems by replacing uncertain parameter by its expected value. Then, models

are reduced to certain models and solved by certain methods. In actuality, many of the previous researches have explored the problems in certain conditions. Additionally, researchers usually have considered just demand of parts to be stochastic, [2], [24], [3].

1c) In the last group, some papers have been developed to introduce heuristic methods to solve CMS problem in uncertain situations, [13], [6], [7].

Based on the above discussions, a little research applied stochastic programming (2b) with the standard structure to solve integrated cell formation problem and tactical decisions. Thus, from the view point of authors, there is a considerable gap in the literature to research and this can indicate strengths of the current research via previous works.

In the structure of any stochastic programming (SP) problem, one must decide which decision variables are first stage and which are second stage; in other words, which variables must be determined first and which may be determined after the uncertainty has been realized.

2- Cellular manufacturing decisions are strategic decisions that are affected by operational decisions such as scheduling, production planning, layout consideration, utilities, productivity and etc. Thus, in order to enhancing decision making regarded to formation of cells, it is essential to incorporate strategic decisions with operational decisions in a unique problem. Recently, researchers have had some efforts in order to integrate two types of decisions. But the lack of literature is that most of them are done in certain situations while in real-world most of the operational parameters are uncertain; and thus, integrated problems must be more studied in uncertain situations [17], [18], [25].

In our paper, both mentioned drawbacks are solved concurrently where cellular manufacturing problem is incorporated with group scheduling problem under a supply chain design framework in stochastic solution space and these can show dominance of our work rather than the others.

2. Robust Optimization Methodology

Mulvey et al. [26] proposed robust optimization where the method was able to attempt decision makers' errand risk aversion or service level task. Also, their method could give up a series of solutions that are less sensitive to realizations of the data in a scenario set. The optimal solution supplied by a robust optimization model is called robust if it stays "close" to the optimal whenever

the input data vary called solution robustness. A solution is called robust if it is "almost" feasible for small changes in the input data called as model robustness.

Structural and control constraints are the two types of robust optimization constraints. Structural constraints are modeled based on the concept of linear programming and its input data, which are free of any noise. On the other hand, control constraints are applied as an auxiliary constraint fluctuated by noisy data. Furthermore, in any robust optimization problem, one must decide which decision variables are treated in the first stage and which ones are considered for the second stage; that is, which variables must be set now and which may be set after the uncertainty is resolved [27, 28 and 34]. In robust CMS problem, cell formation (CF) decisions must be made now (design variable), before it is known which scenario will come to pass due to its strategic impact, while scheduling decisions are under uncertainty since they are determined in future after uncertainty in processing times have been realized (control variable).

In this section, a robust optimization framework will be applied where details of this are discussed by [29]. Let X be a set of design variable and Y be a set of control variable. Also, there are two types of constraints: structural constraint and control constraint. Since, the coefficients used in the control constraint are subject to the noise. Therefore, they depend on the number of scenarios and they are indexed by s .

There are some situations where it is impossible to achieve a solution for the model that is both feasible and optimal for all scenarios. Consequently, the trade-off between solution robustness and model robustness should be resolved through application of the decision making process. The robust optimization model is defined to measure this trade-off. For this purpose, a vector error ξ_s that can measure the infeasibility permitted in the control constraint under a scenario s will be introduced. Thus, robust model will be formulated as follows:

$$\text{Min } Z = f(x, y_1, y_2, \dots, y_s) + \omega \cdot g(\xi_1, \xi_2, \dots, \xi_s) \quad (1)$$

$$\text{ST: } Ax = b \quad (2)$$

$$B_s x + C_s y_s + \xi_s = e_s \quad \forall s \quad (3)$$

$$x \geq 0, \quad y_s \geq 0 \quad \forall s \quad (4)$$

In the above model, variable vector y_s is control variable and X is design variable where (2) is structural constraint and (3) is control constraint.

The first term of the objective function computes an expected value of $p_s \times f_s$ where p_s is the probability of occurring scenario s and f_s is objective function value under scenario s . In this research like [29], we

focus on the application of stochastic non-linear programming model called robust optimization, in which higher moment of the distribution of f_s is proposed in the robust model. Therefore, the optimal solution is less sensitive to uncertain and noisy data in which the variables or parameters reflect the probability of occurrence. The second term of the objective function is the infeasibility penalty function where is applied to penalize fluctuations of the control constraint (3) under some scenarios. When a solution is infeasible under some scenarios then fluctuation occurs.

Weight w is used due to compute the trade-off between solution robustness measured by the first term and model robustness measured by the penalty term. For example, if w is assumed zero, then the objective function tries to minimize only the first term and solutions may be infeasible under some scenarios. On the other hand, if w is assumed a large positive number then the second term will dominate the objective and solutions may have a large cost.

In order to model the objective function, we apply the objective function proposed by [26] and [29] where consists of the mean value with an addition of a constant μ times the variance of fluctuations.

$$f(x, y_1, y_2, \dots, y_s) = \sum_s p_s \times f_s + \mu \sum_s p_s \left(f_s - \sum_{s'} p_{s'} \times f_{s'} \right)^2 \quad (5)$$

As μ increases, the solution is less sensitive to the changes in the data under all scenarios. The objective function (5) has a nonlinear and quadratic term which needs a great computation time to reach optimal solution. To overcome this problem, [29] and [30] introduced the following function instead of the previous version to reach a linear model:

$$f(x, y_1, y_2, \dots, y_s) = \sum_s p_s \times f_s + \mu \sum_s p_s \left[\left(f_s - \sum_{s'} p_{s'} \times f_{s'} \right) + 2\theta_s \right] \quad (6)$$

$$ST: f_s - \sum_{s'} p_{s'} f_{s'} + \theta_s \geq 0 \quad \theta_s \geq 0 \quad (7)$$

Proof) [26]

3. Robust Optimization to formulate CMS problem

3.1. Real world cases contain uncertain processing time

I) In this paper, uncertainty is considered in internal parameter processing times for parts and external parameter demands. Note that uncertainty in internal and external parameters can be described differently,

discrete or continues form, based on the properties of the real applications. For instance, sometimes it may be possible that design of parts is changed during production process (group 4 in the literature surveys) due to improvement in technology, production equipments, customers' expectations, etc. These changes can happen in various industries such as computer and mobile phone sets where products' life cycles are short and products' design have to be changed for many times to satisfy new customers' expectations in a long planning period. Therefore, these changes can vary design aspects and finally can fluctuate processing time and required amount of demands in production plan. Note that these changes are not certain events in the future. Hence, in order to design cellular manufacturing effectively, changes must be predicted as some discrete scenarios by given probabilities at the beginning of the planning phase.

II) Also, in some cases such as condition-based maintenance system, where after each inspection based on the unknown degree of deterioration maintenance is performed; the time required for maintenance and also, amount of failures are not predictable and will be uncertain [11].

III) For another application, consider a recycling system planning for second-hand products at their end-of-life phase where products have reverse cycle from end to start to be processed such as in reverse supply chains. The recycling and reverse process of a product, however, is characterized by uncertainties due to unknown usage influences on the product. In such case, the time needed for recycling products and amount of reversed products are uncertain at the beginning planning period.

Above discussions illustrates situations in which internal and external parameters (processing time and demand) are uncertain at the beginning of the planning horizon and therefore the rest analysis must be applied.

3.2. Model description

The robust optimization technique introduced in the previous section can now be used to study the effects of uncertainty in integrated CMS and group scheduling problem under a supply chain design. We assume that processing time of parts either completed inside the systems or by suppliers and also products' demands are uncertain and are described by discrete scenarios. We have set of scenarios, that each of them may be occurred with probability p_s where $\sum_s p_s = 1$. Due to multiple

scenarios in the problem; group scheduling decisions in cells must be determined under each scenario. Scheduling problem in a cellular manufacturing environment is treated as group scheduling problem,

which assumes that all parts in a part family are processed in the same cell and no inter-cellular transfer is needed. Thus, for a bottleneck cell with higher cycle time, there are two alternatives to decrease cycle time: duplicating bottleneck machines and outsourcing exceptional parts to the suppliers in a supply chain network. In this formulation, we minimize total expected cost included expected and variance of tactical costs (waiting time leads to holding cost) in cells, cost of solution infeasibility, total strategic cost (sub-contracting costs regarded to the employment of suppliers in a network) in order to outsource exceptional elements and the cost of resource underutilization. This cost component occurs when the parts, which have no need to be operated on a machine, placed together in a same cell. We consider the following assumptions in formulation process:

1. The problem will be formulated by robust optimization technique.
2. There is a single supplier for outsourcing each product.
3. Outsourcing cost included operation cost, transportation cost and inventory cost.
4. The model will have trade-off between solution robustness and model robustness.
5. All parts are available at the beginning of the planning period.
6. While an operation starts on a machine, it cannot be interrupted before completion of the process.
7. Set-up time for parts is sequence independent and it is a portion of the processing time.
8. Machines are available during the planning period and they are not failed.
9. Processing times for each part on each machine are uncertain and described by set of discrete scenarios.
10. Cost of sub-contracting to the suppliers and underutilization is known and deterministic.
11. Batch size is constant for all productions.
12. Machines are always available at the beginning of planning period.
13. Holding cost for all parts per unit time is constant and the same.
14. The maximum number of cells that can be formed is specified in advance.
15. Maximum cell size is known in advance.
16. Number of operations for each part is deterministic and known.

Notations

Indexes

- i Part index: $\{i=1, \dots, P\}$
- j Machine index: $\{j=1, \dots, M\}$
- k Cell index: $\{k=1, \dots, C\}$
- s Scenario index: $\{s=1, \dots, S\}$
- r Sequence order index $\{r=1, \dots, R\}$

Parameters:

- D_i^s Demand for part i under scenario s
- PT_{ij}^s Processing time of part i on machine j in scenario s when an operation is performed in cells
- PTO_{ij}^s The time needed for outsourcing operation j to the suppliers for part i under scenario s
- H Holding cost per unit time in which a part have to wait in cells
- NM_{Max} Maximum number of machines permitted to be located in each cell
- NC Maximum number of cells to be formed in problem
- P^s Probability of scenario s
- U_{ij} Cost of underutilization if part i has no need to be processed on machine j and placed together in a same cell
- CO_i The cost of out scouring each operation of part i to the suppliers.
- DD_i Due date for part i
- $a_{ij}=1$ If part i needs to be processed on machine j , otherwise 0
- μ The penalty cost of the violation in scheduling costs under all scenarios
- ω The penalty cost of the infeasibility for due date restrictions

Decision Variables

- $X_{ik} = 1$ If part i is assigned to cell k , otherwise, 0
- $Y_{jk} = 1$ If machine j is assigned to cell k , otherwise, 0
- $RS_{i[r]}^s = 1$ If sequence $[r]$ is assigned to part i under scenario s , otherwise, 0
- TPT_{ik}^s Total processing time of part i needs to be processed in cell k under scenario s
- $TPR_{[r]k}^s$ Total processing time of a part with sequence $[r]$ in cell k under scenario s
- $FT_{[r]k}^s$ Total flow time of a part assigned to the sequence $[r]$ in cell k under scenario s including waiting time and processing time.
- $DP_{[r]k}^s$ Demand for a part assigned to the sequence order $[r]$ in cell k under scenario s .
- $FlowTime_{ik}^s$ Total flow time of specific part i assigned to cell k under scenario s .
- ξ_{ik}^s The Tardiness of part i assigned to cell k under scenario s (infeasibility measurement for due date restrictions).

CF decisions are scenario – independent: they must be made before occurring scenarios and they are made based on similarity of processing parts and are independent to quantity of processing time. Scheduling decisions are scenario – dependent, thus $RS_{i[r]}^s$, TPT_{ik}^s , $TPR_{[r]k}^s$, $FT_{[r]k}^s$, $FlowTime_{ik}^s$, ξ_{ik}^s variables are indexed by scenario since they should be made after we realized scenario.

Objective Function. Model (8) finds the optimal solution for the robust problem to be used for different scenarios and it consists of different terms.

$$\begin{aligned}
 Min\ TC &= \sum_{s=1}^S p_s \times \left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^s \times FT_{[r]k}^s \right) \\
 &+ \mu \sum_{s=1}^S p_s \left[\left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^s \times FT_{[r]k}^s \right) - \right. \\
 &\left. \sum_{s'=1}^S p_{s'} \left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^{s'} \times FT_{[r]k}^{s'} \right) + 2\theta_s \right] \\
 &+ \omega \sum_{s=1}^S \sum_{k=1}^c \sum_{r=1}^R p_s \times DP_{[r]k}^s \times \xi_{[r]k}^s \\
 &+ \sum_{s=1}^S \sum_{k=1}^c \sum_{i=1}^P \sum_{j=1}^M P_s \times CO_i \times a_{ij} \times D_i^s \times X_{ik} \times (1 - Y_{jk}) \\
 &+ \sum_{s=1}^S \sum_{k=1}^c \sum_{i=1}^P \sum_{j=1}^M P_s \times U_{ij} (1 - a_{ij}) \times D_i^s \times X_{ik} \times Y_{jk}
 \end{aligned}$$

The first term indicates the expected value of scheduling cost (tactical cost). The second term is the variance of the scheduling cost and measures solution robustness like [29] described in (6). The third term measures the model's robustness with respect to infeasibility associated with control constraints (18) under scenario s . Finally, the fourth and fifth terms compute expected sub-contracting cost (strategic cost because of suppliers) as well as the expected cost of resource underutilization that are dependent to scenarios since are weighted by products' demand.

Constraints

$$\sum_{k=1}^C X_{ik} = 1 \quad \forall i \tag{9}$$

$$\sum_{k=1}^C Y_{jk} = 1 \quad \forall j \tag{10}$$

$$TPT_{ik}^s - \sum_{j=1}^M PT_{ij}^s \times X_{ik} \times Y_{jk} = 0 \quad \forall i, k, s \tag{11}$$

$$\sum_{r=1}^R RS_{i[r]}^s = 1 \quad \forall i, s \tag{12}$$

$$\sum_{i=1}^P X_{ik} \times RS_{i[r+1]}^s \leq \sum_{i=1}^P X_{ik} \times RS_{i[r]}^s \quad \forall k, s, r \tag{13}$$

$$\sum_{i=1}^P X_{ik} \times RS_{i[r]}^s \leq 1 \quad \forall k, s, r \tag{14}$$

$$TPR_{[r]k}^s - \sum_{i=1}^P RS_{i[r]}^s \times TPT_{ik}^s = 0 \quad \forall r, k, s \tag{15}$$

$$DP_{[r]k}^s - \sum_{i=1}^p D_i^s \times RS_{i[r]}^s = 0 \quad \forall r, k, s \tag{16}$$

$$FT_{[r]k}^s - \sum_{w=1}^r TPR_{[w]k}^s = 0 \quad \forall r, k, s \tag{17}$$

$$FlowTime_{ik}^s - \sum_{r=1}^R FT_{[r]k}^s \times RS_{i[r]}^s = 0 \quad \forall i, k, s \tag{18}$$

$$FlowTime_{ik}^s + \sum_{j=1}^M a_{ij} \times X_{ik} \times (1 - Y_{jk}) \times PTO_{ij}^s - \xi_{ik}^s \leq DD \quad \forall i, k, s \tag{19}$$

$$\sum_{j=1}^M Y_{jk} \leq NM_{Max} \quad \forall k \tag{20}$$

$$\left[\left(\sum_{k=1}^c \sum_{r=1}^R H \times DP_{[r]k}^s \times FT_{[r]k}^s \right) - \sum_{s'=1}^S p_{s'} \left(\sum_{k=1}^c \sum_{r=1}^R H \times DP_{[r]k}^{s'} \times FT_{[r]k}^{s'} \right) + \theta_s \right] \geq 0 \quad \forall s \tag{21}$$

$$X_{ik}, Y_{jk}, RS_{i[r]}^s \in \{0, 1\}$$

$$TPT_{ik}^s, TPR_{[r]k}^s, FT_{[r]k}^s, FlowTime_{ik}^s, \xi_{ik}^s, DP_{[r]k}^s, \theta_s \geq 0$$

In the proposed model, in order to simplify modeling process, at first model's information are converted from parts-based to the sequences-based by equations (15) and (16) and then information are returned to parts-based by equation (18). This is the first time that such innovative approach is applied in mathematical modeling enables decision makers to solve the problem with the least computations.

Set constraint (9) and (10) indicate that each part and each machine must be assigned to exactly one cell, respectively. Set constraint (11) computes total processing time for each part in a cell for all scenarios. Set constraint (12) guarantees that each part must be assigned to single sequence order in each scenario. Set constraint (13) states that parts must be ordered

sequentially in each cell and each scenario. Therefore, filling sequence orders must have a decreasing trend and sequence $[r]$ cannot be assigned to any part until sequence order $[r-1]$ is assigned to the other part. Thus, in each cell, one part will have sequence order 1 and the other sequences order will be filled by the other parts, orderly. Set constraint (14) guarantees that in each cell at most one part can be assigned to each sequence order. Set constraint (15) computes total processing time for a part assigned to the sequence $[r]$ and cell k in scenario s . In other words, this equation indicates that which processing time is assigned to sequence order $[r]$. Set constraint (16) computes amount of assigned demand to sequence order $[r]$ in cell k under scenario s based on the part which is located to that sequence order. Set constraint (17) calculates total flow time including waiting time and processing time for a part assigned to sequence order $[r]$ in cell k under scenario s . For a part

assigned to sequence order $[r]$, amount of $\sum_{\psi=1}^{r-1} TPR_{[\psi]k}^s$

shows waiting time and $TPR_{[r]k}^s$ indicates processing time. Therefore, the time in which the process of a part assigned to sequence $[r]$ ends will be sum of the two mentioned terms. Note that since in group scheduling processing, the amount of time for all operations of each part are summed to make a single process, so process of a part assigned to sequence $[r]$ cannot be started unless grouped operation of a part assigned to sequence $[r-1]$ is finished. Set constraint (18) determines total flow time for a specific part i which is assigned to cell k under scenario s based on the sequence order of the part. Set constraint (19) is a control constraint and used to guarantee that parts under each scenario have no tardiness. The left side of this constraint has two terms. The first term shows total flow time of parts in manufacturing cells and the second term explains total time for outsourced operations to the suppliers which are not completed in cells. It is assumed that if operation j of part i needs to be outsourced, the time required for completion will be PTO_{ij}^s instead of PT_{ij}^s . It is noted that for each part if total time for completion all operation is less than its due date then under minimization the deviation $\xi_{ik}^s = 0$, whereas if the part has lateness then $\xi_{ik}^s > 0$ and the infeasible solution is reached and all infeasibilities are minimized in the objective function. Set constraint (20) specifies maximum number of machines allowed in each cell. Set constraint (21) is an auxiliary constraint similar to (7) and is applied to linearize objective function discussed in equation (5). The last set constraint indicates type of decision variables.

3.3. Model Linearization

Since, the proposed model is nonlinear one; we reformulate the model as a mixed-integer linear programming model by introducing new sets of variables and auxiliary constraints to improve computational efficiency. In some terms such as objective function, set constraints (11), (13), (14) and (19), pure quadratic 0-1 term is composed where two binary variables are multiplied to each other. Also, in some terms such as set constraints (15) and (18), quadratic mixed 0-1 term is composed where a continuous variable and a binary variable are multiplied to each other. In this section, we will try to linearize these terms applying linearization methods. In the first group, linearization method for pure 0-1 problems proposed by [31] will be applied. Also, linearization method for mixed 0-1 problems proposed by [32] will be applied for the second group.

To linearize the proposed model, above techniques are applied as the following types:

Type 1) $LP1_{ijk}$ will be substituted with $X_{ik} \times Y_{jk}$ using in objective function and set constraints (11) and (19) ([28]).

Type 2) $LP2_{ik[r]}^s$ will be substituted with $X_{ik} \times RS_{i[r]}^s$ using in set constraints (13) and (14) ([28]).

Type 3) $LP3_{ik[r]}^s$ will be substituted with $RS_{i[r]}^s \times TPT_{ik}^s$ using in set constraint (15) ([29]).

Type 4) $LP4_{ik[r]}^s$ will be substituted with $FT_{[r]k}^s \times RS_{i[r]}^s$ using in set constraint (18) ([29]).

3.3.3 Linear Model

$$\begin{aligned} \text{Min } TC = & \sum_{s=1}^S p_s \left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^s \times FT_{[r]k}^s \right) \\ & + \mu \sum_{s=1}^S p_s \left[\left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^s \times FT_{[r]k}^s \right) \right. \\ & \left. - \sum_{s'=1}^S p_{s'} \left(H \times \sum_{k=1}^c \sum_{r=1}^R DP_{[r]k}^{s'} \times FT_{[r]k}^{s'} \right) + 2\theta_s \right] \\ & + \omega \sum_{s=1}^S \sum_{k=1}^c \sum_{r=1}^R p_s \times DP_{[r]k}^s \times \xi_{[r]k}^s \end{aligned}$$

$$\begin{aligned}
 &+ \sum_{s=1}^S \sum_{k=1}^c \sum_{i=1}^P \sum_{j=1}^M P_s \times CO_i \times a_{ij} \times D_i^s \times (X_{ik} - LP1_{ijk}) \\
 &+ \sum_{s=1}^S \sum_{k=1}^c \sum_{i=1}^P \sum_{j=1}^M P_s \times U_{ij} (1 - a_{ij}) \times D_i^s \times LP1_{ijk} \\
 (22)
 \end{aligned}$$

Constraints: Set constraints (9)- (10)- (12)- (16)- (17)- (20) and (21) without change
 Set constraint (11) altered as:

$$TPT_{ik}^s - \sum_{j=1}^M PT_{ij}^s \times LP1_{ijk} = 0 \quad \forall i, k, s \quad (23)$$

Set constraint (13) altered as:

$$\sum_{i=1}^P LP2_{ik[r+1]}^s \leq \sum_{i=1}^P LP2_{ik[r]}^s \quad \forall k, s, r \quad (24)$$

Set constraint (14) altered as:

$$\sum_{i=1}^P LP2_{ik[r]}^s \leq 1 \quad \forall k, s, r \quad (25)$$

Set constraint (15) altered as:

$$TPR_{[r]k}^s - \sum_{i=1}^P LP3_{ik[r]}^s = 0 \quad \forall r, k, s \quad (26)$$

Set constraint (18) altered as:

$$FlowTime_{ik}^s - \sum_{r=1}^R LP4_{ik[r]}^s = 0 \quad \forall i, k, s \quad (27)$$

Set constraint (19) altered as:

$$FlowTime_{ik}^s + \sum_{j=1}^M a_{ij} \times (X_{ik} - LP1_{ijk}) \times PTO_{ij}^s - \xi_{ik}^s \leq DD_i \quad \forall i, k, s \quad (28)$$

$$LP1_{ijk} \leq X_{ik} \quad \forall i, j, k \quad (29)$$

$$LP1_{ijk} \leq Y_{jk} \quad \forall i, j, k \quad (30)$$

$$LP1_{ijk} \geq X_{ik} + Y_{jk} - 1 \quad \forall i, j, k \quad (31)$$

$$LP2_{ik[r]}^s \leq X_{ik} \quad \forall i, k, r, s \quad (32)$$

$$LP2_{ik[r]}^s \leq RS_{i[r]}^s \quad \forall i, k, r, s \quad (33)$$

$$LP2_{ik[r]}^s \geq X_{ik} + RS_{i[r]}^s - 1 \quad \forall i, k, r, s \quad (34)$$

$$LP3_{ik[r]}^s \leq TPT_{ik}^s \quad \forall i, k, r, s \quad (35)$$

$$LP3_{ik[r]}^s \leq M \times RS_{i[r]}^s \quad \forall i, k, r, s \quad (36)$$

$$LP3_{ik[r]}^s \geq TPT_{ik}^s - M \times (1 - RS_{i[r]}^s) \quad \forall i, k, r, s \quad (37)$$

$$LP4_{ik[r]}^s \leq FT_{[r]k}^s \quad \forall i, k, r, s \quad (38)$$

$$LP4_{ik[r]}^s \leq M \times RS_{i[r]}^s \quad \forall i, k, r, s \quad (39)$$

$$LP4_{ik[r]}^s \geq FT_{[r]k}^s - M \times (1 - RS_{i[r]}^s) \quad \forall i, k, r, s \quad (40)$$

In above formulation, auxiliary constraints (29), (30) and (31) guarantee linearization type 1. Auxiliary constraints (32), (33) and (34) ensure linearization type 2. Auxiliary constraints (35), (36) and (37) indicate linearization type 3. Finally, auxiliary constraints (38), (39) and (40) guarantee linearization type 4. Also, M is a large positive number.

The proposed linear model will be discussed analytically and sensitivity analysis will be performed respect to the robust optimization terminology in the section 5. However, the proposed linear model is NP-hard problem, and thus applying it for real world applications requires an effective solution procedure that can solve it in large- scales. Therefore, in the next section, we introduce an effective solution approach based on the combination of two solution methods under an optimization rule and then detailed computational results will be presented in section 5.

4. The Parallel Solution Procedure

There exist many innovative combinations of different types of metaheuristics methods in order to solve problems regarded to the manufacturing environments [33], [34] and [35]. In this section, we propose a parallel structured solution approach to solve introduced stochastic and nonlinear model in large-scale, efficiently. The proposed algorithm involves new combination of genetic algorithm (GA) and simulated annealing (SA) under an optimization rule where both algorithms try to find sub optimal solution in a parallel structure considering exchanging data and information between each other. In addition, Fig 1 illustrates the conceptual framework of our method.

In this process, GA and SA algorithms attempt to obtain sub optimal solution and also share their information about the best solution, simultaneously. In this way, each algorithm sends information about sub optimal solution to the other algorithm through some designed information channels. By these channels, SA receives the best solution from all solutions generated by GA and then creates the best neighborhood solution by its own procedure based on the solution received. Furthermore, when SA forms the best neighborhood solution, and then sends it to the set of solution created by GA process. Thus, SA solution has a chance to be selected for the next generation and participation in GA process. This interaction between GA and SA transfers information between the best neighborhood solution found by SA and also the set found by GA. This cycle continues until both algorithms reach high quality

solution, simultaneously. In other words, each algorithm has two main duties. The first is to find the best sub optimal solution by its own procedure and the second is to assist the other algorithm to reach solutions by sending its information about solution space. By this way, performance of the both algorithm will be improved, significantly and they are able to reach the best solution, concurrently. For more features, each box in Fig 1 will be discussed at the rest of this section.

Since the structure of finding the best solution in GA and SA has many similarities such as neighborhood solution, solution representation, fitness function, next solution and mutation process, thus we have decided to combine these algorithms to enhance strengthen of both them. In addition, because other algorithms such as tabu search and ant colony or swarm optimization have different procedures to find solution, so they cannot be combined with GA or SA and cover weaknesses. In other words, SA can be completely a sub-ordinate section of GA because it can be adjusted as a mutation process to find neighborhood solution but other methods due to many differences with GA cannot be combined. Finally, in our opinion the best combination is constructed by GA and SA.

4.1. Optimizing method: Shortest Processing Time rule

Since the aim of scheduling information is to minimize expected waiting time of parts in cells, thus an optimization rule for scheduling is considered as a subordinate part of the algorithm. This method is based on shortest processing time (SPT) rule which is approved by theory of neighbor pairs that it can minimize total waiting time for parts assigned to a same cell under each scenario as follows. This can leads to improvement of solution procedure performance:

In order to minimize waiting time for parts under each scenario, sequence of parts should be planned so that the part with shorter processing time to be processed earlier.

4.2. Parallel algorithm phases

4.2.1 GA boxes

- GA Initialization. Initialization phase consist of defining chromosome structure, population size, cross over and mutation rates and number of generation.

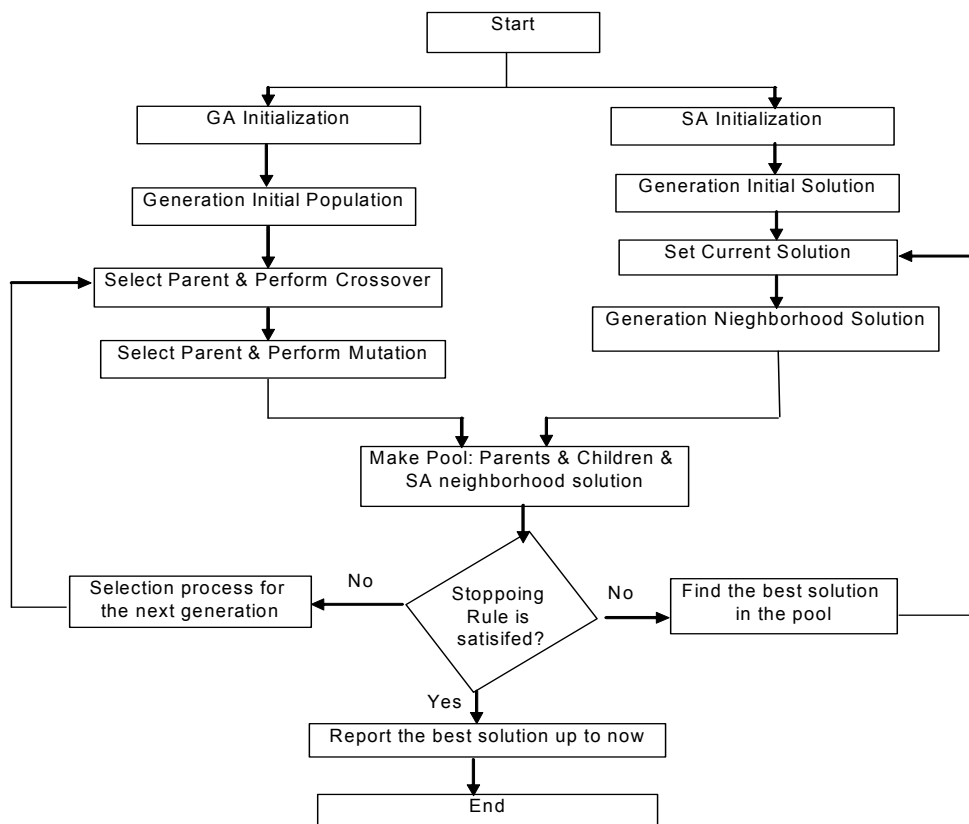


FIGURE 1. Conceptual framework of the proposed algorithm

- Hierarchical solution structure. In order to encode CF and scheduling information for both parts and machines, a two-layer hierarchical schema is proposed. In the first layer, a string of integer numbers is used to encode the CF results for machine and then part genes. In the second layer with s rows, scheduling information denoting sequence order of each part under each scenario will be specified. The genes regarded to the first layer were used to control the genes in the second layer under a hierarchical manner. For illustration, consider a data set included 7 parts and 5 machines to be classified into two manufacturing cells under two scenarios. Table 1 indicates a typical chromosome structure. The allele of each gene in the first layer represents the cell number to which the machine or part belongs. For instance, machine 1 is assigned to cell #1 and so on. As such, cell #1 contains machines 1, 2 and 5 and also parts 1, 2, and 4. In the second layer, representation contains two rows with 7 genes denotes sequence order of each part in a cell under each scenario. For example, sequence order of part 3 is 3 under scenario 1 and is 2 under scenario 2.

Other initial parameters will be set using a trail and error method during numerical procedures to find the best value for them.

TABLE 1. Sample of solution structure

		P1	P2	P3	P4	P5	P6	P7	M1	M2	M3	M4	M5
First Layer	Cell Assignment	1	1	2	1	2	2	2	1	1	2	2	1
Second layer	Sch. Under S=1	2	1	3	3	1	4	2	-	-	-	-	-
	Sch. Under S=2	1	2	2	3	1	3	4	-	-	-	-	-

Step IV: When total processing time for all parts in cells and under all scenarios is found, based on the SPT rule, scheduling decisions under all scenarios will be made. Thus, in each cell and under each scenario a part with a shorter processing time has priority and is processed earlier that leads to minimization of waiting time under all scenarios. At the end of this stage, for some parts it may be possible that due date restriction is not satisfied and then penalty cost is arisen in the objective function.

Step V: In this step, in order to decode chromosome structure and determine value of the other scheduling variables, total processing time of each sequence order $[r]$ based on the part assigned to it in cell k and scenario s is determined by using eq. (20).

Step VI: Once sequence order of each part under each scenario has been determined, then it is possible to compute tardiness of each part in each scenario by comparing the time that processing time ends against due date restrictions. In other words, infeasibility index

- Initial Population. The first generation is created by initializing the population of chromosomes, $X^k = [x_i^k]$, $k = 1, 2, \dots$, pop size from the feasible region $\{(X) | g_i(x_i) \leq 0, x_i = 1, 2, \dots, n\}$, randomly. Six steps will be performed to achieve each feasible solution in initial population:

Step I: Assignment of parts

In this step, parts are assigned to the cells, randomly. In this way, a random number from 1 to # of cells for each part is generated, and then the part will be assigned to the selected cell.

Step II: Assignment of machines

In this step, machines are assigned to cells according to capacity of cells. In this procedure, for each machine, a random number from 1 to # of cells is generated and a cell is generated by chance. If number of machines assigned to the selected cell is less than its capacity, the machine will be assigned to the selected cell. Otherwise, this step is repeated until a feasible solution is reached.

Step III: In each cell and under each scenario total processing time for each part needs to be processed in cell will be computed using eq. (16).

for each part under each scenario can be determined and penalty cost in objective function will be found.

- Crossover Operator. Selected parents V'_1, V'_2, V'_3, \dots are grouped to the pairs $(V'_1, V'_2), (V'_3, V'_4), \dots$ without loss of generality to be combined. For all parts and machines, a random real number λ from the open interval $(0, 1)$ is generated. Since each part / machine is assigned to different cells in a pair of solutions (parents), thus in order to produce new solution (child), part / machine will be assigned to one cell with probability 0.5 based on the value of parameter λ . The crossover operator on V'_2 and V'_1 produce one child X with assignment decisions are illustrated in figure 2. Once assignment decisions of new child are found then steps III to VI described in initialization phase will be performed to decide scheduling information.
- Mutation operators. Mutation is designed to prevent premature convergence and to explore a new solution space. In order to have the most efficient process two types of mutation operators have been

designed. In the first type, a conventional group exchange mutation operator is used in which exchanges all genes' alleles between two cut points for layer 1 at one time for parts or machines [29]. Sample of this type is illustrated in Figure 3.

Parent 1		P.1	P.2	P.3	P.4	P.5	P.6	P.7	M.1	M.2	M.3	M.4	M.5
First layer	Cell Assignment	1	1	2	1	2	2	2	1	1	2	2	1
Second layer	Sch. Under S=1	2	1	3	3	1	4	2	-	-	-	-	-
	Sch. Under S=2	1	2	2	3	1	3	4	-	-	-	-	-

Parent 2		P.1	P.2	P.3	P.4	P.5	P.6	P.7	M.1	M.2	M.3	M.4	M.5
First layer	Cell Assignment	2	1	1	1	2	2	1	2	1	2	1	2
Second layer	Sch. Under S=1	1	3	2	4	3	2	1	-	-	-	-	-
	Sch. Under S=2	3	3	4	2	1	2	1	-	-	-	-	-

Offspring		P.1	P.2	P.3	P.4	P.5	P.6	P.7	M.1	M.2	M.3	M.4	M.5
First layer	Cell Assignment	1	1	2	1	2	2	2	1	1	2	1	2
Second layer	Sch. Under S=1	Scheduling decisions will be made using steps III to VI							-	-	-	-	-
	Sch. Under S=2	Scheduling decisions will be made using steps III to VI							-	-	-	-	-

FIGURE 2. Sample of crossover operation

		P.1	P.2	P.3	P.4	P.5	P.6	P.7	M.1	M.2	M.3	M.4	M.5
First Layer	Cell Assignment	1	1	2	1	2	2	2	1	1	2	2	1
Second Layer	Sch. Under S=1	For this new assignment scheduling will be updated using SPT rule							-	-	-	-	-
Layer	Sch. Under S=2	For this new assignment scheduling will be updated using SPT rule							-	-	-	-	-

FIGURE 3. Sample of mutation type 1

After this process, due to modification of assignment decisions, all cells must be rescheduled again based on the steps III to VI.

In the second type, the algorithm tries to decrease amount of infeasibility index regarded to due date limitations in order to generate robust solutions. As it was discussed earlier employing step VI of initialization phase enables us to find parts in which has tardiness or has infeasibility in due date restriction. After this step, for a selected solution to be motivated number of parts with infeasible index will be found. Then, 25% of parts will be selected by chance among all infeasible parts. Sequence order for each selected part will be changed to an earlier order and the part will be processed earlier rather than its previous sequence order by generating a random new sequence order.

- Selection Process for the next generation in GA phase. The selection process is based on selecting 50% from the best chromosomes and other 50% randomly.
- *Stopping criterion.* The genetic algorithm process is terminated once the process performs fixed number of iterations.

4.2.2 Simulated annealing algorithm boxes

- *Representation of solution.* Each solution represents a feasible solution made of hierarchical structure such as a solution in GA population. Thus, all characteristics described in GA solution representation are applied in this section.

- *Initialization.* In this phase, we describe how to procedure initial parameters in SA. Initial and final values for the control parameter temperature, named as T_0 and T_f respectively, are determined as follows.

Initial temperature: Due to significant influence of the objective function on the initial temperature, some illustrative examples are given. Also, its value should be large enough so that probability of acceptance of new solutions at the initial temperature T_0 reaches at least 80%. In this way, we generate 100 solutions randomly and compute their objective functions defined for each of them as g_i . We compute the difference between two sequential answers denoted by Δf_i . So, we compute an initial temperature like [36]:

$$T_0 = \frac{\Delta f_{\max}}{-Ln(0.8)} \tag{23}$$

Pseudo code of initial temperature is shown in Figure 4.

✓ *Final temperature:* this parameter is set to be:

$$T_f = 0.08 \times T_0 \tag{24}$$

The cooling rate (α) is considered to be a constant. Also, an initial solution is produced similar to initial random population generation in GA employing steps I to VI.

Phase 2) *Generate feasible neighborhood solution.* When an initial solution is created, the objective function is computed. Then to generate neighborhood solution we modify current solution by one of the following move types:

Move type 1: selection of a part randomly, and change assignment of the selected part to the new cell by chance.

Move type 2: selection of a machine by chance, and finding number of the other cells with free capacity so that the selected machine can be assigned to them. Then, a new cell with free capacity is found randomly and the machine is assigned to it.

After each type, since assignment decisions are changed thus all cells must be rescheduled again by steps III to VI.

Move type 3: Similar to the second mutation type in GA process, to improve infeasibility measurement sequence order of parts with unsatisfied due date constraints will

be altered. By this way, penalty cost for unsatisfied due dates will be decreased and the final solution will be robust, too.

Flowchart of this procedure is shown in figure 5.

- *Examine acceptance condition.* Determine cost difference between current solution and neighborhood solution. The metropolis condition is used to determine the probability of acceptance neighborhood solution. This probability is computed as $e^{-\frac{\Delta Cost}{T_i}}$ where T_i is current temperature. The neighborhood will be accepted if a random number from interval (0, 1) be less than the probability of acceptance and then neighborhood solution is replaced with current solution.
- *Update counters.* Increase counter value by one. If the iteration counter value is less than to maximum for the current temperature. Otherwise go to stop.
- *Adjust temperature.* Adjusting temperature is done by cooling rate. Thus, decrease current temperature with rate α . In other words, temperature in the next process is computed as $T_{i+1} = T_i \times \alpha$ where T_i is current temperature. If new temperature is greater than final temperature, then set counter value as zero, otherwise stop.
- *Stopping criterion.* The simulated annealing process is terminated once the system is frozen. In other words, the value of cost function is not

improved after a certain number of iteration has been executed. The SA process is stopped if cooling process is iterated n times and we achieve final temperature.

Since the other parts of SA algorithm are very well known, thus, it's sufficient to present only the title of the other aspects in SA; however, they are applied in algorithm coding:

- *Evaluation of current solution and neighborhood solution,* - *Examination of acceptance condition,* - *Update counters,* - *Adjusting temperature* ($T_{i+1} = \alpha \times T_i$), - *Stopping criteria in each temperature,*

4.3 Benchmark heuristic procedure

To demonstrate efficiency of the proposed method, we compare proposed method against branch and bound algorithm and also a benchmark heuristic procedure (introduced by [37]) with a decomposition structure. This algorithm has two phases where in phase I part families based on maximization of similarity coefficients are determined and in phase II scheduling and the other tactical decisions based on previous phase are made. For the proposed model, we apply this heuristic decomposition procedure using same structure and solve the model in two steps.

```

Sub initial temperature
For i = 1 to 100
  Do
    Generate random solution
    Let  $g_i$  = fitness function of new solution
  End Do
Next i
For j = 2 to 100
  Do
    Let  $\Delta f_j = |g_j - g_{j-1}|$ 
  End Do
Next j
Let  $\Delta f_{max} = \max(\Delta f_2, \dots, \Delta f_{100})$ 
Let  $\Delta = \Delta f_{max}$ 
Prob(Acceptance at  $T_0$ )  $\geq 0.8 \Rightarrow$ 
 $e^{-\frac{\Delta}{T_0}} \geq 0.8 \Rightarrow -\frac{\Delta}{T_0} \geq \ln(0.8) \Rightarrow \frac{T_0}{\Delta} \geq \frac{1}{-\ln(0.8)}$ 
 $\Rightarrow T_0 \geq \frac{\Delta}{-\ln(0.8)}$ 
End sub
    
```

FIGURE 4. Pseudo code of initial temperature

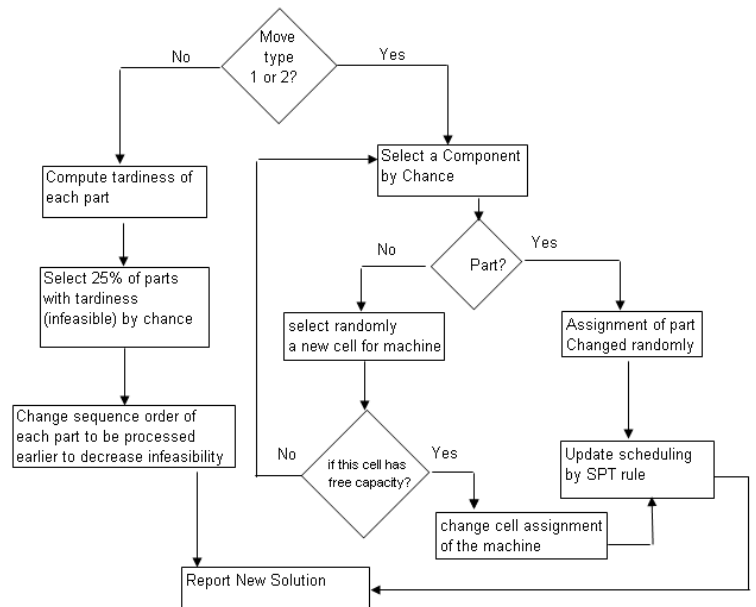


FIGURE 5. Flowchart of neighborhood structure

5. Experimental design and research hypotheses

5.1 Trade-off between solution robustness and model robustness

In the first section of computational results, we illustrate analytical performance of the proposed linear model through sensitivity analysis. The importance of weight w in the objective function (8) is to find a trade-off between solution robustness (near to optimal solution) and model robustness (near to feasible space). Robust optimization terminology permits infeasibility in the control constraints by means of penalty costs. In the proposed research, once $w=0$, ξ_{ik}^s in constraint (18) will be equaled to

$$FlowTime_{ik}^s + \sum_{j=1}^M a_{ij} \times X_{ik} \times (1 - Y_{jk}) \times PTO_{ij}^s - DD_i$$

minimize objective function and then the model wants

to have the highest delay in delivering parts due to have not penalties in tardiness events. Therefore, each part will have maximum possible tardiness with no penalty cost. Thus, total tardiness cost attains the highest value. It is obvious that this scheduling plan cannot be adapted. Hence, it is critical to analysis the introduced robust optimization model with different value of w . Fig. 6 points up the trade-off between feasibility costs and optimality costs. As the weight increases, the expected total cost representing solution robustness increases significantly, and the expected tardiness cost representing model robustness drops. This implies that for larger values of w , the solution obtained is approaching “almost” feasible for any realization of scenario s through the payment of more total costs. The expected tardiness will eventually drop to zero with an increase in value of w . Based on the decision maker preferences; tardiness is specified to be allowable for parts up to 15%. This has determined the value of w to be 100. The results are consistent with those reached by [29] and [39].

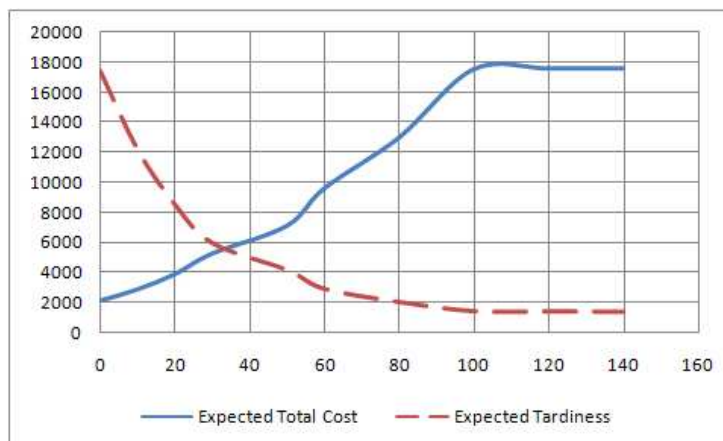


FIGURE 6. Correlation between expected total cost and expected tardiness considering variable penalty cost

5.2 Effectiveness of the proposed solution method

In this section, we evaluate the quality of parallel structured algorithm by means of a computational study. We test the proposed algorithm on random problems, and compare its performance with the solutions reached by branch-and-bound (B&B) algorithm by using the Lingo 8.0 software and also a heuristic algorithm. 22 test problems are provided and solved by this method. This algorithm is coded in Visual Basic 6.0 and runs on a 3.2 GHz Pentium IV computer. To find medium-sized problems scale, we have started from a small-sized problem and increased gently the size of problem by a

specific rule until the exact approach could not reach the optimum solution within a predetermined run time. In a similar fashion, for large-sized problems, we have started from the largest medium-sized problem and increased the size of the problem by a specific rule until the branch and bound algorithm could not reach the feasible solution within a run time. These problems are generated randomly based on consideration of similar data in the literature.

In the first section, some small cases are generated in order to demonstrate the performance of the proposed algorithm. We solve them by employing three methods: branch and bound algorithm, proposed parallel method and benchmark heuristic procedure.

In the second section, we generate some medium sized problems and since optimal solver is unable to reach optimal solutions in a predetermined run time, to evaluate efficiency of the proposed method, we consider a run time restriction for Lingo solver and compare results of parallel method with the best solutions found by Lingo as an upper bound. In this way, each example is allowed to be solved within 5400 seconds (1.5 hours). Thus, to solve the medium sized problems, we consider a possible interval for optimum value of objective function (F^*) that constructed by the F^{bound} and F^{Best} values that are proposed by Lingo software where $F^{bound} \leq F^* \leq F^{Best}$. Based on the Lingo software's documents, the F^{Best} determines the best feasible solution found so far. Also, F^{bound} defines the bound on the objective function. This bound is a limit on how far the solver will be able to improve the

objective. At some points, these two values may become very close. Given that the best objective value can never exceed the bound, the fact that these two values are close determines that Lingo's current best solution is either the optimal solution, or very closes to it. At such a point, the user can interrupt the solver and accept with the current best solution in order to save additional computation time. As we said before, we interrupt the solver within 5400 seconds (this procedure is similar to [36] and [38]). It is noted that CPU time unit is based on the second.

In the third branch, we generate some large sized problems and due to inability of Lingo software to find feasible solution within 5400 seconds, we just use the proposed benchmark heuristic algorithm to evaluate performance of the proposed method in large scale problems.

TABLE 2. Parameter setting of the total process

Parameters	Level	Values
Crossover rate (Pc)	2	0.85 0.95
Mutation rate (Pm)	2	0.10 0.20
Population size (Ps)	2	20 30
Generations	2	250 300
α : Cooling rate	2	0.85 0.95
K : number of iterations in each temperature	2	150 200
T_0	1	$\frac{Max\{\Delta f_i\}}{-Ln(0.8)}$
T_f	1	$0.08 \times T_0$

5.2.1 Numerical experiments in small sized problems

From Table 3, the presented algorithm is able to find and report the optimal and promising solutions in a reasonable computational time. This indicates the success of our proposed algorithms in small sized problems. Also, it is concluded that there is no gap between quality of solutions reached by both algorithms and optimal solution. It implies that both proposed

algorithm and heuristic procedures are effective to solve the presented model in small sized problems. Table 3 can show us other information. As mentioned earlier, a problem which cannot be solved optimally within 5400 seconds is in class of medium sized problems. So, problems with higher scale rather than example S10 will be placed in the class of medium or large sized problems.

TABLE 3. Numerical Experiments for small-sized test problems

Test Problem	P	M	C	S	Lingo Optimal Solution		Our proposed solution		Heuristic Solution
					OFV	CPU time	OFV	CPU time	OFV
S1	3	2	2	1	114.3	1	114.3	<1	114.3
S2	3	2	2	2	121.1	1	121.1	<1	121.1
S3	4	2	2	2	125.8	1	125.8	<1	125.8
S4	5	3	2	1	127.4	26	127.4	<3	127.4
S5	6	4	2	1	133.2	108	133.2	<3	133.2
S6	6	4	2	2	139.4	422	139.4	<3	139.4
S7	7	4	2	2	144.8	1688	144.8	<7	144.8
S8	7	5	3	2	152.9	2412	152.9	<7	152.9
S9	8	6	3	1	154.7	3856	154.7	<7	154.7
S10	8	6	3	2	160.2	5248	160.2	<7	160.2

OFV: Objective Function Value

CPU time unit = second

TABLE 4. Numerical Experiments for medium-sized test problems

Test Problem	P	M	C	S	Lingo's the best solution		Our proposed solution		Heuristic solution	Gap between (1) & (2)	Gap between (2) & (3)
					OFV (1)	CPU time	OFV (2)	CPU time			
M1	9	6	3	1	176.2	>5400	164	<10	164	7.44%	0.00%
M2	9	6	3	2	197.8	>5400	179	<10	179	10.50%	0.00%
M3	10	7	3	1	224.6	>5400	192	<10	196	16.98%	2.08%
M4	10	7	3	2	279.2	>5400	229	<15	235	21.92%	2.62%
M5	11	8	3	1	307.3	>5400	264	<15	278	16.40%	5.30%
M6	11	8	3	2	354.7	>5400	302	<15	312	17.45%	3.31%
L1	12	9	3	1	Not Feasible		Average:			15.12%	2.22%

5.2.2 Effectiveness of the proposed method in the medium sized problems

In this section, some medium instances are generated and then solved by both *B&B* algorithm and benchmark heuristic procedure. As it was discussed earlier, Lingo solver cannot reach global optimum solutions in maximum run time (5400 seconds). Therefore, solutions achieved by proposed method are compared with the best objective function reached by branch-and-bound method as an upper bound for optimal solution. Also, we solve these problems by benchmark heuristic procedure to validate efficiency of this algorithm in solving problems. Table 4 points up results for these problems.

As shown in the last row of Table 4, the average value of $F_{Best}(B \& B) - Our(solution) / Our(solution) \times 100$ is 15.12% which implicates to a better performance of proposed algorithm rather than the best solution of *B&B* algorithm in limited run time. Indeed, from table 4, when scale of problems is increased, solutions obtained by benchmark heuristic procedure is to be lie between the solutions of *B&B* and proposed solutions

($Z_{B\&B}^{Best} \geq Z_{Heuristic}^{Best} \geq Z_{proposed\ Method}^{Best}$) and therefore, our algorithm has also a better performance rather than benchmark heuristic algorithm in medium sized problems, too (2.22% on average).

5.2.3 Efficiency of the proposed method in large sized problems

To estimate quality and efficiency of the solutions obtained by our proposed algorithm in large sized scales, performance validation is according to the solutions reached only by benchmark algorithm. From Table 5, summary of this comparison and the other attributes of sets are shown. In this way, we define a measurement named it 'quality percent' computed by $(\text{heuristic OFV} - \text{our OFV}) / \text{our OFV} \times 100$ (Objective Function Value). The solution results presented here and also the last row show that average value of quality percent with value 6.08% associates to a better performance of our algorithm rather than heuristic procedure in large sized scales. Also, the final cells configuration is illustrated in table 6 for problem M3.

TABLE 5. Numerical Experiments for large-sized test problems

Test Problem	P	M	C	S	Our	Heuristic	Gap between (1) & (2)
					propose OFV (1)	solution OFV (2)	
L1	15	10	4	1	877	905.2	3.21%
L2	20	15	5	2	1869	1976.3	5.74%
L3	25	18	5	2	11243	12254.9	9.00%
L4	30	20	5	1	16052	17366.7	8.19%
L5	35	25	6	2	21854	23058.2	5.51%
L6	40	30	6	1	29120	30529.4	4.84%
Average:							6.08%

TABLE 6. Solution results for problem M3

Machines		Parts									
		5	6	9	10	1	2	4	3	7	8
Cell 1	G	1	1	1	1						
	B	1	1	1							
Cell 2	C		1			1	1	1		1	
	A					1	1	1			
Cell 3	E								1	1	1
	F	1							1	1	1
	D								1	1	

5.3 Sensitivity analysis on amount of uncertainty

In order to illustration of results, the following numerical examples are provided.

In this section, a comprehensive example is provided in order to illustrate effectiveness of the proposed model. For this purpose, two numerical examples are designed and then solved. All parameters in both examples are the same except one parameter and it is "processing time for parts" once they are outsourced to suppliers to do operations (PTO_{ij}^s). Generally, in each example

there are two choices in order to operate processes of parts: operating processes inside the manufacturing system or outsourcing them to the suppliers. It is assumed that processing time for operations will be uncertain and described by discrete scenarios either they are completed inside the manufacturing system or by the suppliers. In both examples, there are 2 scenarios where probability of the first one is p and the second one is $1-p$. In the first example, it is permitted that processing time in both situations (doing inside manufacturing system or by suppliers) under scenario 2 has up to 100% violation compared to the first scenario. In other words, the second scenario can have 100% violation rather than the first scenario.

But, in the second example, only processing time for operations which are completed inside of the manufacturing system have 100% violation. In addition, operations which are completed by the suppliers have only 40% violation.

We have solved both examples and final configurations of the cells are as follows. Figure 7 points up part-machine matrix. Figure 8 illustrates work cells formed in example 1 and also, figure 9 indicates work cells formed in example 2.

Figure 7. part-machine matrix

	M1	M2	M3	M4	M5	M6	M7	M8
P1	1	0	1	0	1	0	1	0
P2	0	0	1	0	0	0	1	0
P3	0	1	0	0	0	1	0	1
P4	0	0	1	0	0	0	1	0
P5	1	0	1	1	1	1	0	1
P6	1	0	0	1	1	0	0	0
P7	0	1	1	0	0	1	0	1
P8	0	1	0	0	0	1	0	0
P9	1	0	0	0	1	0	0	0
P10	1	0	0	1	0	0	0	0

Figure 8. Configuration of cells in the first example

	M1	M4	M5	M2	M6	M8	M3	M7
Cell 1	P5	0	1	1	0	1	1	0
	P6	1	1	1	0	0	0	0
	P10	1	1	0	0	0	0	0
	P9	1	0	1	0	0	1	0
Cell 2	P3	0	0	0	1	1	1	0
	P7	0	0	0	1	1	1	1
	P8	0	0	0	1	1	0	0
Cell 3	P1	1	0	1	0	0	0	1
	P2	0	0	0	0	0	0	1
	P4	0	0	0	0	0	0	1

Figure 9. Configuration of cells in the second example

	M1	M4	M5	M8	M2	M6	M3	M7
P5	0	1	1	1	0	1	0	0
P6	1	1	1	0	0	0	0	0
P10	1	1	0	0	0	0	0	0
P9	1	0	1	1	0	0	0	0
P3	0	0	0	1	1	1	0	0
P7	0	0	0	1	1	1	1	0
P8	0	0	0	0	1	1	0	0
P1	1	0	1	0	0	0	1	1
P2	0	0	0	0	0	0	1	1
P4	0	0	0	0	0	0	1	1

By comparing two above figures a very interesting result will be discovered. As it can be found, number of inter-cell transportation in the first example is 5 and in the second example is 6. In other words, 5 suppliers are required for operating processes in the first example and 6 suppliers are required in the second example. This result can be justified through the following discussions. The concept of above result is that in the second example more suppliers are applied in order to do operations. It is because of the fact that amount of uncertainty (40%) for suppliers' characteristics in the second example is less than amount of uncertainty

(100%) for suppliers in the first example. Since, robust optimization terminology aims to minimize violation of solution or uncertainty in the final result, thus in the second example, the proposed model tried to find a solution with lower uncertainty and violation. So, the model selected more number of suppliers with lower uncertainty level (40%) in order to minimize total uncertainty of the problem. Therefore, the achieved numerical result conforms to the expected results of a robust model and this can validate effectiveness of the proposed model.

6. Conclusion and Discussions

In this study, a robust optimization model for cellular manufacturing system integrated with group scheduling aspects was developed. Assuming a future economic scenario that has associated probability, an optimal cell configuration is less sensitive to the change in the noisy and uncertain data. Manufacturing management can obtain a robust solution by a tradeoff between optimality and infeasibility measurements. Then mathematical method applied to linear the proposed model. A new combination of GA and SA is proposed where both algorithms aim to reach sub optimal solution, concurrently through exchanging data between each other. We divided our computational experiments to four sections. At first, the trade-off between solution robustness and model robustness is analyzed. Also, effectiveness of the proposed method was validated through three classes: small, medium and large sizes. Our contributions research field consists of: considering uncertain processing time described by robust optimization theory which yields to more flexibility and practical aspects in real world cases, integrating cell formation problem with scheduling aspects, linearization of the model and presenting a parallel algorithm which had successful performance in any size of problem.

For future research, we suggest three directions:

- ✓ Development of the model under more and the other stochastic parameters such as costs, processing routes and machine availability.
- ✓ Considering this problem as a multi objective model which considers CF decisions in one objective and scheduling in the other objective.
- ✓ Aggregating proposed model with the other production aspects like layout problem considerations.
- ✓ Formulating the problem as a chanced constrained programming where the probability of tardiness event for parts is less that service level can be another novel developments.

These remain critical issues for future study.

Acknowledgment

This research is supported by Grant of Islamic Azad University South Tehran branch by Faculty Research Support Fund from the Faculty of Engineering, Tehran, Iran.

References

- [1] Hurley, S.F., Clay Whybark, D.: Inventory and capacity trade-off in a manufacturing cell, *International Journal of Production Economics*, Vol.59, No.1, PP. 203-212, 1999.
- [2] Song, S.-J., Hitomi, K.: Determining the planning horizon and group part family for flexible cellular manufacturing, *Production Planning and Control*, Vol.7, No.6, PP. 585-593, 1996.
- [3] Balakrishnan, J., Cheng, C.H.: Dynamic Cellular Manufacturing under multi-period planning horizon, *European Journal of Operational Research*, Vol.177, No.1, PP. 281-309, 2007.
- [4] Safaei, N., Saidi-Mehrabad, M., Tavakkoli-Moghaddam, R., Sassani, F.: A fuzzy programming approach for cell formation problem with dynamic & uncertain conditions, *Fuzzy Sets and Systems*, Vol.159, No.2, PP. 215-236, 2008.
- [5] Papaioannou, G., Wilson, J.M. Fuzzy extensions to integer programming of cell formation problem in machine scheduling, *Annals of Operations Research*, Vol.166, No.1, PP. 1-19, 2009.
- [6] Sun, Y.-L., Yih, Y.: An intelligent controller for manufacturing cell, *International Journal of Production Research*, Vol.34, No.8, PP. 2353-2373, 1996.
- [7] Andres, C., Lozano, S., Adenso-Diaz, B: Disassembly sequence planning in a Disassembly cell, *Robotics and Computer-Integrated Manufacturing*, Vol.23, No.6, PP. 690-695, 2007.
- [8] Xinlu Ma, Lindu Zhao and Lothar Schulze, Performance of Automated Storage/Retrieval Systems under Stochastic Demand Using Queueing Theory, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4469-4478, 2009.
- [9] Shanker, R., Vrat, P.: Post design modeling for cellular manufacturing system with cost uncertainty, *International Journal of Production Economics*, Vol.55, No.1, PP. 97-109, 1998.
- [10] Kuroda, M., Tomita, T: Robust design of a cellular-line production system with unreliable facilities, *Computers and Industrial Engineering*, Vol.48, No.3, PP. 537-551, 2005.
- [11] Hosseini, M.M.: An inspection model with minimal and major maintenance for a system with deterioration and Poisson failures, *IEEE Transactions on Reliability*, Vol.49, No.1, PP. 88-98, 2000.
- [12] Siemiatkowski, M., Przybylski, W.: Modeling and simulation analysis of process alternative in cellular manufacturing of axially symmetric parts, *International Journal of Advanced Manufacturing Technology*, Vol.32, No.5-6, PP. 516-530, 2007.
- [13] Asgharpour, M.J., Javadian, N.: Solving a stochastic cellular manufacturing model using genetic algorithm, *International Journal of Engineering, Transactions A: Basics*, Vol.17, No.2, PP. 145-156, 2004.
- [14] Ravichandran, K.S., Chandra Sekhara Rao, K.: A new approach to fuzzy part family formation in cellular manufacturing system, *International Journal of Advanced Manufacturing Technology*, Vol.18, No.8, PP. 591-597, 2001.
- [15] Szwarc, D., Rajamani, D., Bector, C.R: Cell formation considering fuzzy demand and machine capacity, *International Journal of Advanced Manufacturing Technology*, Vol.13, No.2, PP. 134-147, 1997.
- [16] Yang, J., Deane, R.H: Setup time reduction and competitive advantage in a closed manufacturing cell, *European Journal of Operational Research*, Vol.69, No.3, PP. 413-423, 1993.

- [17] Solimanpur M., Prem Vrat, Ravi Shankar: A heuristic to minimize makespan of cell scheduling problem, *Int. J. Production Economics*, Vol.88, No.3, 231–241, 2004.
- [18] Ahmadizar, F., Barzinpour, F., (2010): A hybrid algorithm to minimize makespan for the permutation flow shop scheduling problem, *International Journal of Computational Intelligence Systems*, 3(6), 853 - 861
- [19] Vakharia, A. J., and Kaku, B. K.. Redesigning a cellular manufacturing system to handle long-term demand changes: A methodology and investigation. *Decision Sciences*, Vol.24, No.5, 909–917, 1993.
- [20] Mahdavi I., Javadi B., Fallah-Alipour K., Slomp J.: Designing a new mathematical model for cellular manufacturing system based on cell utilization, *Applied Mathematics and Computation*, Vol.190, No.4, PP.662–670, 2007.
- [21] Kahraman, C., Engin, O., Yilmaz, M. K. (2009), A New Artificial Immune System Algorithm for Multi-objective Fuzzy Flow Shop Problems, *International Journal of Computational Intelligence Systems*, 2(3), 236 – 247
- [22] Kahraman, C., Engin, O., Yilmaz, M. K. (2008), An application of effective genetic algorithms for Solving Hybrid Flow Shop Scheduling Problems, *International Journal of Computational Intelligence Systems*, 1(2), 134 – 147
- [23] Shanker, R., Vrat, P.: Some design issues in C.M. using the fuzzy programming approach *International Journal of Production Research*, Volume 37, Issue 11, 1999, Pages 2545-2563
- [24] Tavakkoli-Moghaddam, R, Javadian, N., Javadi, B., Safaei, N.: Design of a facility layout problem in CMS with stochastic demand *Applied Mathematics and Computation* Volume 184, Issue 2, 15 January 2007, Pages 721-728
- [25] Aneja Y.P., Kamoun H., (1999), Scheduling of parts and robot activities in a two machine robotic cell, *Computers & Operations Research* 26, 297-312
- [26] Mulvey, J.M., Vanderbei, R.J., Zenios, S.A., (1995). Robust optimization of large-scale systems. *Operations Research* 43, 264–281.
- [27] Snyder L.V., (2006): Facility location under uncertainty: a review, *IIE Transactions*, Vol.38, No.2, 537–554, 1995.
- [28] Ghezavati V.R., Jabal-Ameli M.S., Makui A., : A new heuristic method for distribution networks considering service level constraint and coverage radius, *Expert Systems with Applications*, Vol. 36, No. 3, PP. 5620-5629, 2009.
- [29] Stephen C.H. Leung, Sally O.S. Tsang, W.L. Ng, Yue Wu: A robust optimization model for multi-site production planning problem in an uncertain environment, *European Journal of Operational Research*, Vol.181, No.1, PP. 224–238, 2007.
- [30] Yu, C.S., Li, H.L.,. A robust optimization model for stochastic logistic problems. *International Journal of Production Economics*, Vol. 64, No.2, PP.385–397, 2000.
- [31] Glover F., Woolsey L.: Converting the 0-1 polynomial programming problem to a 0-1 linear program, *Operations Research*, Vol.22, PP. 180–182, 1974.
- [32] Ghezavati, V.R. and Saidi-Mehranad, M. (2011): An efficient linearization technique for mixed 0-1 polynomial problems, *Journal of computational & applied mathematics*, 235(6), pp. 1730-1738.
- [33] Jiangang Yi and Yao Li, A RBF Neural Network Based Optimization Model and Its Application in Mechanical Manufacturing Production Plan, *ICIC Express Letters*, vol.4, no.3(B), pp.1033 -1038, 2010.
- [34] Feng-Tse Lin and Tzong-Ru Tsai, A Two-stage Genetic Algorithm for Solving the Transportation Problem with Fuzzy Demands and Fuzzy Supplies, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4775-4786, 2009.
- [35] Ghosh, T., Sengupta, S., Chattopadhyay, M, and Dan, P K (2011), Meta-heuristics in cellular manufacturing: A state-of-the-art review, *International Journal of Industrial Engineering Computations*, 2:pp. 87-122.
- [36] Safaei N., Saidi-Mehrabad M., Jabal-Ameli M.S., A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, *European Journal of Operational Research*, Vol.185, PP.563–592, 2008.
- [37] Geonwook Jeon, Herman R. Leep , Forming part families by using genetic algorithm and designing machine cells under demand changes, *Computers & Operations Research*, Vol.33, PP.263–283, 2006.
- [38] Ghezavati, V. R. and Saidi-Mehrabad, M.: Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time, *International Journal of advanced manufacturing technology*, Article in press, DOI: 10.1007/s00170-009-2322-2.
- [39] Mulvey, J.M., Ruszczyński, A. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research* Vol. 43, PP.477–490, 1995.