# CREDIBILITY-BASED FUZZY PROGRAMMING MODELS TO SOLVE THE BUDGET-CONSTRAINED FLEXIBLE FLOW LINE PROBLEM

A. GHODRATNAMA, S. A. TORABI AND R. TAVAKKOLI-MOGHADDAM

Abstract. This paper addresses a new version of the flexible flow line problem, i.e., the budget constrained one, in order to determine the required number of processors at each station along with the selection of the most economical process routes for products. Since a number of parameters, such as due dates, the amount of available budgets and the cost of opting particular routes, are imprecise (fuzzy) in practice, they are treated as fuzzy variables. Furthermore, to investigate the model behavior and to validate its attribute, we propose three fuzzy programming models based upon credibility measure, namely expected value model, chance-constrained programming model and dependent chance-constrained programming model, in order to transform the original mathematical model into a fuzzy environment. To solve these fuzzy models, a hybrid meta-heuristic algorithm is proposed in which a genetic algorithm is designed to compute the number of processors at each stage; and a particle swarm optimization (PSO) algorithm is applied to obtain the optimal value of tardiness variables. Finally, computational results and some concluding remarks are provided.

## 1. Introduction

In today's world, high demands for goods and urgency to supply them have caused new and modern types of manufacturing systems to come into focus. A flexible flow line is one of the modern manufacturing systems with several applications in industry, in which the required machines are arranged serially in different work stations (stages) according to the process route of given products where at least one work station has more than one processor. Primary products in the form of raw materials enter the required stations stage by stage in a forward flow manner and are converted into the final products. The main problems arising in a flexible flow line from the scheduling viewpoint consist of sequencing sub-problem (i.e., determining the sequence vector by which the products enter the line) and assignment sub-problem (i.e., assigning the incoming products to the available machines at each stage with parallel processors).

The literature of flexible flow line problem is abundant dealing with different variants of the problem. Various processors (with similar process capabilities like different turning machines) can be assigned to every stage. Obviously, a higher number

Received: September 2010; Revised: April 2011; Accepted: July 2011

*Key words and phrases:* Budget-constrained flexible flow lines, Credibility-based fuzzy programming, Meta-heuristic, Genetic algorithm, Particle swarm optimization.

of processors results in a lesser completion time, because products are distributed among processors uniformly, therefore, the completion time of each product is also reduced. In this paper, a new version of the flexible flow line problem, i.e., the budget constrained one is proposed by which the available budget is also incorporated in the model formulation. In this way, the number of processors at each stage is determined by taking into account the cost of processors. Figure 1 shows a schematic view of concerned system. Notably, the model has only one objective function, i.e., minimization of weighted tardiness times. In this paper, we develop three distinct credibility-based fuzzy programming models based on credibility measure, namely, the expected value, the chance-constrained programming and the dependent chance-constrained programming models. To solve these fuzzy models, a hybrid meta-heuristic algorithm is also proposed. In order to compare the proposed fuzzy models and validate the proposed solution algorithm, an extensive computational experiments conducted and the corresponding numerical results are reported.

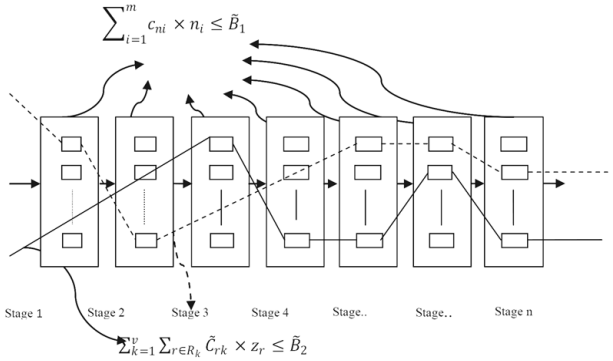



FIGURE 1. Flexible Flow Line and Budget Concept

The remainder of this paper is organized as follows. The literature review is provided in section 2. The proposed fuzzy flexible flow line model is elaborated in section 3. Some required preliminaries on credibility measure are introduced in section 4, and the proposed credibility-based fuzzy programming models as well as their crisp equivalent models are given in section 5. A hybrid meta-heuristic algorithm is developed in section 6 to solve the resulting complex models heuristically. Computational experiments and respective results are reported in section 7. Finally, the concluding remarks are given in section 8.

## 2. **Literature Review**

In a flow line (or flow shop) system, each machine center involves just one machine. If at least one machine center has more than one machine, then the system is called a flexible flow-line system [2, 43]. Therefore, flexible flow lines can be considered as a mixture of flow lines and parallel machines systems. Scheduling a set of jobs in a flexible flow-line is a NP-complete problem [17]. Flexible flow-line problems have received considerable attention from researchers during the last

two decades. However, the scheduling criterion most frequently considered in those papers was the maximum completion time [8, 20]. Furthermore, these studies propose either an exact algorithm, which can solve at most medium-sized problem instances, or different types of heuristic or approximation algorithms [22, 30]. Botta-Genoulaz [5] proposed six heuristic approaches for minimizing the maximum tardiness in a scheduling problem with N jobs and M stages in a flexible flow-line environment with identical parallel machines under time lag and due date constraints. Lin and Liao [23] proposed a near-optimal heuristic approach for minimizing the maximum tardiness in a two-stage flexible flow-line scheduling problem. They propose an integer-linear programming model and apply a genetic algorithm and a lower bound as approximate solution methods, respectively. Vob and Witt [54] studied a 16-stage real-life scheduling problem in a flexible flow-line environment to minimize the tardiness. They develop a limited resource mathematical model based on this scheduling problem, and then solve it using a heuristic approach. Liu and Chang [29] used a Lagrangian Relaxation approach for solving the m-stage flexible flow-line problem that minimizes the sum of setup and preparation times. Kurz and Askin [21] examined the m-stage flexible flow-line problem with sequence-dependent setup times for minimizing the makespan. Pugazhendhi et al [39] applied some heuristic approaches for the m-stage flexible flow-line problem to minimize the total weighted flow-time. Ruiz et al [43] and Ruiz and Maroto [42] used heuristic and meta-heuristic approaches for minimizing the makespan in the m-stage flexible flow-line problem. In the conventional scheduling problems, the input parameters, such as job processing times, ready times and due dates, are often assumed to be deterministic [18]. However, in the real-world situations, these parameters are often tainted with uncertainties. In this regard, scheduling problems can be classified into two categories: deterministic and uncertain (e.g., stochastic or fuzzy) ones. However, when formulating a scheduling problem in the real-world, various parameters are often imprecise in nature. For example, the processing time of a given job is around 3 minutes and its due date is about 10. This is especially true when the sufficient objective data are not available and these factors should be estimated by relying on the knowledge and experience of the decision maker. In these cases, it seems to be more appropriate to consider the parameters as fuzzy ones. There are also several research works in the scheduling literature under fuzzy environment. Konno and Ishii [20] discussed about an open line scheduling problem with the fuzzy allowable time and fuzzy resource constraint when dealing with fuzzy due dates. Shivinian et al [49] considered an optimization problem with linear objective function subject to fuzzy relation inequalities constraints with max-average composition. Safi et al [44] eliminated the difficulties of the Zimmermann method [56] for solving fuzzy linear programming problems. Cadenas and Verdegay [7] surveyed the main models and methods in fuzzy mathematical programming. Also, regarding to fuzzy concept many researchers have considered the use of heuristic and meta-heuristic algorithms in their studies [e.g., see 3,10,11,19,45,35]. Litoiu and Tadei [24] developed some new models for real-time task scheduling with fuzzy deadline and processing times. Abbas-bandy and Alavi [1] introduced a method for solving

fuzzy linear systems. Cadenzas and Verdegay [6] considered a primer on fuzzy optimization models and methods. Mehdizadeh et al [32] solved an optimum problem considering fuzzy clustering criteria by a hybrid PSO and fuzzy c-mean clustering algorithm. To tackle fuzzy scheduling models heuristically, there are some rules, such as the classical dispatching rules [36], fuzzy ranking [31] fuzzy dominance relation methods [4] and so on. Here, we are facing with a fuzzy optimization problem for which the three recently developed fuzzy programming models based upon the credibility measure are proposed, and appropriate strategies are applied to defuzzify these fuzzy models. In this way, the expected value model is first presented [26] which is the most commonly used approach to solve the practical problems where the decision maker wants to make such a decision with the minimum expected cost subject to some expected constraints [40]. The chance-constrained programming model [16,25,28,41] is another commonly used formulation when the decision maker wishes to satisfy some chance constraints in at least some given confidence levels [27]. Dependent chance programming model is the third commonly used approach [27] which is appropriate when the decision maker may prefer to maximize the credibility of satisfying the event when the total cost is no more than a given value, such as capital limit.

## 3. **Proposed Fuzzy Flexible Flow Line Model**

The main assumptions used in the model formulation are as follows:
A fuzzy cost is considered for each of multiple routes as well as each processor type at each stage.
Processing times and due dates of products are considered as fuzzy parameters.
Number of required processors at each stage is endogenously determined through the solution process.
Number of process stages is given.
At each stage, more than one processor can be installed.
No backward movement is allowed.
For each route the successive stages are predefined.
Two fuzzy budgets, including the total cost limit of purchasing of processors and selecting of routes, are considered.

Also, the following notations are used to formulate the problem mathematically.
Indices:

$i$ Index of processing stages, $i \in I = \{1, .., m\}$
$j$ Index of processors at stage $i, j \in J_i = \{1, .., n_i\}$
$k$ Index of products, $k \in K = \{1, .., v\}$
$r$ Index of processing routes, $r \in R = \{l, ..., w\}$

Input Parameters:

$R_k$ Set of processing routes available for product $k, R_k \subset R$.
$I_r$ Ordered set of processing stages for route $r, I_r = \{f_r, ..., (i-l)_r, i_r, ..., l_r\} \subset I$.
$f_r$ First processing stage in route $r$.
$l_r$ Last processing stage in route $r$.
$\widetilde{P}ikr$ Fuzzy processing time in stage $i$ for product $k$ and route $r$.

$\widetilde{B}_1$ Fuzzy dedicated budget to purchase the required machines.
$\widetilde{B}_2$ Fuzzy dedicated budget to consider various routes.
$\widetilde{D}_k$ Fuzzy due date of product $k$.
$Q$ A large number (not less than schedule length).
$C_i$ Unit purchasing cost of machine type $i$.
$\widetilde{C}_{rk}$ Cost of arranging route $r$ for producing the product $k$.
$W_i$ Weight of job $i$ based on management viewpoint.

Decision Variables:

$n_i$ Number of parallel processors in stage $i$ which is equal to the size of $J_i$, representing the processor members of stage $i$.
$C_{max}$ Schedule length.
$C_k$ Completion time of product $k$ in final stage.
$C_{ik}$ Completion time of product $k$ in stage $i$.
$C_{ikr}$ Completion time in stage $i$ for product $k$ and route $r$.
$T_k$ Tardiness time of delivering product $k$ to customer.
$d_{ik}$: Departure time of product $k$ from stage $i$.
$d_{ikr}$ Departure time from stage $i$ for product $k$ and route $r$.
$x_{ijk}$ 1, if product $k$ is assigned to processor $j \in J_i$ in stage $i \in I$; otherwise, $x_{ijk} = 0$.
$x_{ijkr}$ 1, if product $k$ is assigned to processor $j \in J_i$ in stage $i \in I$ and is processed via route $r$; otherwise, $x_{ijkr} = 0$.
$y_{kl}$ l, if product $k$ precedes product $l$; otherwise, $y_{kl} = 0$.
$z_r$ 1, if processing route $r$ is selected.

By using the above notations, the proposed fuzzy mathematical programming model for the concerned flexible flow line problem is as follows:

$$\underset{\text{s.t.}}{Min} \Sigma_{k=1}^{v} w_k \times T_k \tag{1}$$

$$\Sigma_{j \in J_i} x_{ijkr} = z_r, k \in K, r \in R_k, i \in I_r \tag{2}$$

$$\Sigma_{k \in K} \Sigma_{r \in R_k} \widetilde{p}_{ikr} x_{ijkr} \leq \frac{\Sigma_{k \in K} \Sigma_{r \in R_k} \widetilde{p}_{ikr} Z_r}{|J_i|} + min_{k \in K, r \in R_K}(\widetilde{p}_{ikr}), \quad 2i \in I, j \in J_i : |J_i| > 1 \tag{3}$$

$$C_{f_r kr} \geq \widetilde{P}_{f_r kr} z_r, \ k \in K, r \in R_k \tag{4}$$

$$C_{i_r kr} - C_{(i-1)_r kr} \geq \widetilde{p}_{i_r kr} z_r, \ k \in K, r \in R_k, (i-1)_r, i_r \in I_r : i_r > f_r \tag{5}$$

$$C_{ikr} + Q(2 + y_{kl} - z_r - z_s) \geq \ d_{ils} + \widetilde{p}_{ikr} z_r, \ k, l \in K, \ r \in R_k \quad s \in R_l \ i \in I_k \bigcap I_l \ : k < l \ and \ |J_i| = 1, \tag{6}$$

$$C_{ils} + Q(3 - y_{kl} - z_r - z_s) \geq \ d_{ikr} + \widetilde{p}_{ils} z_s, \ k, l \in K, \ r \in R_k \quad s \in R_l \ i \in I_k \bigcap I_l \ j \in J_i : \ k < l \ and \ |J_i| > 1 \tag{7}$$

$$C_{ikr} + Q(2 + y_{kl} - x_{ijkr} - x_{ijls}) \geq \ d_{ils} + \widetilde{p}_{ikr} z_r, \ k, l \in K, \ r \in R_k \quad s \in R_l \ i \in I_k \bigcap I_l \ : j \in J_i : k < l \ and \ |J_i| > 1, \tag{8}$$

$$C_{ils} + Q(3 - y_{kl} - x_{ijkr} - x_{ijls}) \geq \ d_{ikr} + \widetilde{p}_{ils} z_s, \ k, l \in K, \ r \in R_k \tag{9}$$

$$s \in R_l \; i \in I_k \bigcap I_l \; j \in J_i : \; k < l \; and \; |J_i| > 1,$$

$$C_{i_r kr} = d_{(i-1)_r} + p_{i_r kr} z_r \; k \in K, r \in R_k, (i-1)_r, \; i_r \in I_r : i_r > f_r \tag{10}$$

$$C_{ils} = d_{(i-1)r} + p_{i_r kr} z_r \; k \in K, \; r \in R_k, (i-1)_r, \; i_r \in I_r : \; i_r > f_r \tag{11}$$

$$d_{ikr} \leq Q z_r, \; k \in K, \; r \in R_k, i \in I_r \tag{12}$$

$$C_{l_r kr} = d_{l_r kr}, \; k \in K, \; r \in R_k \tag{13}$$

$$C_{l_r kr} \leq C_k, \; k \in K, \; r \in R_k \tag{14}$$

$$C_{l_r kr} \leq C_{max}, \; k \in K, \; r \in R_k \tag{15}$$

$$C_{max} \geq \sum_{k \in K} \sum_{r \in R_k} \leq \frac{\widetilde{p}_{ikr} z_r}{|J_i|} + \sum_{h \in I : h \neq i} (min_{k \in K, \; r \in R_k} \; (\widetilde{p}_{hkr})) \; i \in I; \tag{16}$$

$$T_k = max_k \{0, C_k \widetilde{D}_k\} \tag{17}$$

$$\sum_{r \in R_k} z_r = 1 \; ; \; k \in K; \tag{18}$$

$$\sum_{i=1}^{v} c_i \times \; n_i \leq \widetilde{B}_1 \tag{19}$$

$$\sum_{k=1} \sum_{r \in R_k} \widetilde{C}_{rk} \times z_r \leq \widetilde{B}_2 \tag{20}$$

$$x_{ijkr} = 0, \; k \in K, \; r \in R_k, \; i \in I_r, \; j \in J_i : \; |J_i| = 1 \tag{21}$$

$$y_{kl} = 0, \; k, l \in K : k \geq l \tag{22}$$

$$C_{ikr} \geq 0, C_k \geq 0, T_k \geq 0, k \in K, r \in R_k, i \in I_r \tag{23),(24),(25}$$

$$d_{ikr} \geq 0, k \in K, r \in R_k, i \in I_r \tag{26}$$

$$x_{ijkr} \in \{0,1\}, z_r \in \{0,1\}, k \in K, r \in R_k, i \in I_r, j \in J_i \tag{27),(28}$$

$$y_{kl} \in \{0,1\} \; k, l \in K \tag{29}$$

The objective function (1) minimizes the sum of the weighted tardiness times. Constraint (2) guarantees that in every stage with parallel processors each product is assigned to exactly one processor, if the stage is in the selected processing route; otherwise, the product is not assigned to this stage. Constraint (3) equalizes in every stage the workload assigned to each parallel processor. Constraint (4) ensures that each product is processed in the first stage of its selected route, and constraint (5) guarantees that it is also processed in all the downstream stages. Constraints (6) and (7) show product non-interference for single processors. For parallel processors, constraints (6) and (7) related to product non-interference use up additional assignment variables $X_{ijkr}$ and $X_{ijls}$. For a given sequence of products and selected processing routes, only one of constraints (8) and (9) is active when both products k and l are assigned to the same processor. Equation (10) indicates that processing of each product in every stage starts immediately after its departure from the previous stage of the selected route. Constraints (11) and (12) enforce variable upper bounds on completion and departure times. Equation (13) ensures that each product leaves the system as soon as it is completed in the last stage of its selected route. Constraint (14) defines the completion time of product K .Constraint (15) defines the maximum completion time, and Constraint (16) imposes a lower bound

on it. Equation (17) computes the tardiness time of each product. Equation (18) ensures that exactly one processing route is selected for each product. Constraint (19) defines an upper bound for the budget of machine purchasing allocated to each stage. Similarly, Constraint (20) defines an upper bound for the budget of adopting the specific route for producing each product. Finally, Constraints (21)-(29) show the type of different decision variables.

## 4. **Preliminaries**

The fuzzy set theory introduced by Zadeh [55] is increasingly applied to a wide variety of practical problems. In the fuzzy decision making environment, there are three prominent fuzzy measures, i.e., the possibility, necessity and credibility measures. Let $\xi$ be a fuzzy variable with membership function $\mu$, and $u$ and $r$ be two real numbers. Then, the possibility degree of occurrence of a fuzzy event, characterized by $\xi \leq r$, is defined by:

$$Pos\{\xi \leq r\} = sup_{u \leq r} \mu(u) \tag{30}$$

While the necessity degree of occurrence of this fuzzy event is defined by:

$$Nec\{\xi \leq r\} = 1 - Pos\{\xi > r\} = 1 - sup_{u\ >r} \mu(u). \tag{31}$$

Also, the credibility measure $C_r$ is defined as an average of possibility and necessity measures, i.e.,

$$Cr\{\xi \leq r\} = \frac{1}{2}(Pos\{\xi \leq r\} + Nec\{\xi \leq r\}) \tag{32}$$

Note that a fuzzy event may fail even though its possibility achieves 1, and hold even though its necessity is 0. However, the fuzzy event must hold if its credibility is 1, and fail if its credibility is 0.

In this paper, the fuzzy parameters are treated by trapezoidal membership functions. Noteworthy, there are different possible forms for presenting the membership function of fuzzy parameters such as: linear, piece-wise linear, exponential, hyperbolic, etc. (see Peidro and Vasant [37] for a comparison of the main types of membership functions). However, the literature review indicates that among the various types of membership functions, the most feasible form for constructing a membership function when formulating fuzzy mathematical programming models especially for practical problems is the linear one in the form of triangular or trapezoidal membership functions, although there may be preferences for other patterns with some applications [50,56]. The main advantage of these linear membership functions is that they generate equivalent, efficient and computationally linear models. Moreover, they are the most common tool for modeling the ambiguous parameters due to their computational efficiency and simplicity in data acquisition (for more justifications, see [34,38,51-53]).

Figures 2 and 3 show these well-known membership functions. Notably, the triangular membership function is a special case of the trapezoidal membership function.

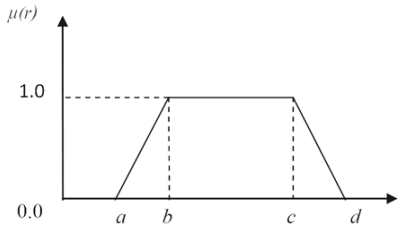



FIGURE 2. A Trapezoidal Membership Function

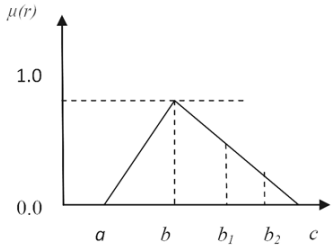



FIGURE 3. A Triangular Membership Function

As such, by considering a trapezoidal membership function, we have:

$$\mu(r) = \begin{cases} \frac{r-a}{b-a} & a \leq r \leq b \\ 1 & b \leq r \leq c \\ \frac{d-r}{d-c} & c \leq r \leq d \\ 0 & otherwise \end{cases} \tag{33}$$

Accordingly, the possibility, necessity and credibility degrees of the event $\xi \leq r$, respectively, are as follows:

$$Pos\{\xi \leq r\} = \begin{cases} 1 & b \leq r \\ \frac{r-a}{b-a} & a \leq r \leq b \\ 0 & otherwise \end{cases} \tag{34}$$

$$Nec\{\xi \leq r\} = \begin{cases} 1 & d \leq r \\ \frac{r-c}{d-c} & c \leq r \leq d \\ 0 & otherwise \end{cases} \tag{35}$$

$$Cr\{\xi \leq r\} = \begin{cases} 1 & d \leq r \\ \frac{d-2c}{2(d-c)} & c \leq r \leq d \\ \frac{1}{2} & b \leq r \leq c \\ \frac{r-a}{2(b-a)} & a \leq r \leq b \\ 0 & otherwise \end{cases} \tag{36}$$

## 5. **Credibility-based Models and Their Crisp Equivalent Ones**

In this section, the original fuzzy model proposed for the problem is first converted to three distinct credibility-based fuzzy programming models. Then, appropriate strategies are applied to transform these credibility-based models into their crisp equivalent ones.

## 6. **The Expected Value Model (EWM)**

Herein, the concept of the expected value for a fuzzy variable and its related crisp equivalent decision model are reviewed. Let $\xi$ be a fuzzy variable. Then, the expected value of $\xi$ is defined by:

$$E[\xi] = \int_0^{\infty} Cr\{\xi \geq r\}dr - \int_{-\infty}^{0} Cr\{\xi \leq r\}dr \tag{37}$$

In this way, for the fuzzy variable $xi = (a, b, c, d)$ with the trapezoidal membership function where $a < b < c < d$, the following equation shows how its expected value can be obtained [26].

$$E[\xi] = \frac{b+c}{2} + \frac{1}{2}\int_{\frac{b+c}{2}}^{c} 1dx + \frac{1}{2}\int_{c}^{d} \frac{x-d}{c-d}dx - \frac{1}{2}\int_{b}^{\frac{b+c}{2}} 1dx = \frac{a+b+c+d}{4} \tag{38}$$

Consequently, by taking into account the aforementioned definitions, the original fuzzy mathematical model (1)-(29) can be reduced to the following expected value model.

Minimize the crisp objective function given in (1)

$$\Sigma_{k\in K}\Sigma_{r\in R_k} E[\widetilde{p}_{ikr}]x_{ijkr} \leq \frac{\Sigma_{k\in K}\Sigma_{r\in R_k} E[\widetilde{p}_{ikr}]Z_r}{|J_i|} + min_{k\in K, r\in R_K}(E[\widetilde{p}_{ikr}]), \tag{39}$$

$$i \in I, j \in J_i : |J_i| > 1$$

$$C_{f_r kr} \geq E[\widetilde{p}_{f_r kr}]z_r,\ k \in K, r \in R_k \tag{40}$$

$$C_{i_r kr} - C_{(i-1)_r kr} \geq E[\widetilde{p}_{i_r kr}]z_r,\ k \in K, r \in R_k, (i-1)_r, i_r \in I_r : i_r > f_r \tag{41}$$

$$C_{ikr} + Q(2 + y_{kl} - z_r - z_s) \geq\ d_{ils} + E[\widetilde{p}_{ikr}]z_r,\ k, l \in K,\ r \in R_k \tag{42}$$

$$s \in R_l\ i \in I_k \bigcap I_l\ : k < l\ and\ |J_i| = 1,$$

$$C_{ils} + Q(3 - y_{kl} - z_r - z_s) \geq\ d_{ikr} + E[\widetilde{p}_{ils}]z_s,\ k, l \in K,\ r \in R_k \tag{43}$$

$$s \in R_l \; i \in I_k \bigcap I_l \;\; : k < l \; and \; |J_i| = 1,$$

$$C_{ikr} + Q(2 + y_{kl} - x_{ijkr} - x_{ijls}) \geq \; d_{ils} + E[\widetilde{p}_{ikr}]z_r, \; k, l \in K, \; r \in R_k \tag{44}$$

$$s \in R_l \; i \in I_k \bigcap I_l \;\; : j \in J_i : k < l \; and \; |J_i| > 1$$

$$C_{ils} + Q(3 - y_{kl} - x_{ijkr} - x_{ijls}) \geq \; d_{ikr} + E[\widetilde{p}_{ils}]z_s, \; k, l \in K, \; r \in R_k \tag{45}$$

$$s \in R_l \; i \in I_k \bigcap I_l \; j \in J_i : \; k < l \; and \; |J_i| > 1$$

$$C_{i_r kr} = d_{(i-1)r} + E[\widetilde{p}_{i_r kr}]z_r \; k \in K, \; r \in R_k, (i-1)_r, \; i_r \in I_r : \; i_r > f_r \tag{46}$$

$$C_{max} \geq \sum_{k \in K} \sum_{r \in R_k} \leq \frac{E[\widetilde{p}_{ikr}]z_r}{|J_i|} + \sum_{h \in I : h \neq i} (min_{k \in K, \; r \in R_k} \; (E[\widetilde{p}_{hkr}])) \; i \in I; \tag{47}$$

$$T_k = max_k\{0, C_k - E[\widetilde{D}_k]\} \tag{48}$$

$$\sum_{i=1}^{m} \; c_i \times \; n_i \leq E[\widetilde{B}_1] \tag{49}$$

$$\sum_{k=1}^{v} \sum_{r \in R_k} \widetilde{C}_{rk} \times z_r \leq E[\widetilde{B}_2] \tag{50}$$

Constraints(2), (10)-(15), (18), (21)-(29).

## 7. The Chance-Constrained Programming Model (CCPM)

Chance-constrained programming (CCP) can also be applied to solve practical optimization problems with the requirement that the chance constraints should hold with at least some given confidence levels provided as an appropriate safety margin by the decision maker [25, 28]. The standard fuzzy chance-constrained programming model can be expressed as follows.

$$\begin{aligned} &\text{Min} \\ &\overline{f} \\ &\text{s.t.} \end{aligned} \tag{51}$$

$$Cr\{f(x, \xi) \leq \overline{f}\} \geq \delta \tag{52}$$

$$Cr\{g_j(x, \xi) \leq 0\} \geq \beta_j, \; j = 1, 2, ..., m \tag{53}$$

In this mathematical model, $x$ is an $n$-dimensional decision vector and $\xi$ is a fuzzy vector. Also, $f$ and $g_j$ denote the objective function and $j$-th constraint, respectively. The aim of this model is to minimize $\overline{f}$. As the constraint (53) indicates, the least required credibility of the objective function $f$ is $\delta$. Similarly, for the $j$-th chance constraint $\beta_j$ shows the least required credibility level. It is noteworthy that in this model, $\delta$ and $\beta_j$ are predetermined confidence levels given by the decision maker [40]. In this regard, for any $\alpha \in (0, 1]$, the $\alpha$-critical value is defined by:

$$\xi_{inf}(\alpha) = inf\{r | Cr\{\xi \leq r\} \geq \alpha\} \tag{54}$$

Considering a trapezoidal membership function, the *alpha*-critical value of $\xi$ is computed by:

$$\xi_{\inf}(\alpha) = \begin{cases} (1-2\alpha)a + 2\alpha b & if \alpha \leq 0.5 \\ (2-2\alpha)c + (2\alpha-1)d & if \alpha > 0.5 \end{cases} \$$

It should be also noted that the weighted sum of some fuzzy variables, including both crisp and fuzzy parameters, is also a fuzzy variable. By using the above definitions, the chance-constrained programming model as the second crisp equivalent one can be formulated as follows [40].

$$\text{Min}$$
$$\overline{f} \tag{55}$$
$$\text{s.t.}$$
$$Cr\{\sum_{k=1}^{v} w_k \times \leq (\overline{f})\} \geq \alpha \tag{56}$$

$$Cr\{\Sigma_{k\in K}\Sigma_{r\in R_k}\widetilde{P}_{ikr}x_{ijkr} \leq \frac{\Sigma_{k\in K}\Sigma_{r\in R_k}\widetilde{p}_{ikr}Z_r}{|J_i|} + min_{k\in K, r\in R_K}(\widetilde{p}_{ikr})\} \geq \beta_{ij}, \tag{57}$$
$$i \in I, j \in J_i : |J_i| > 1$$
$$Cr\{C_{f_rkr} \geq \widetilde{p}_{f_rkr}z_r\} \geq \eta_{kr},\ k \in K, r \in R_k \tag{58}$$

$$Cr\{C_{i_rkr} - C_{(i-1)_rkr} \geq \widetilde{p}_{i_rkr}z_r\} \geq \mu_{ikr},\ k \in K, r \in R_k, (i-1)_r, \tag{59}$$
$$i_r \in I_r : i_r > f_r$$

$$Cr\{C_{ikr} + Q(2 + y_{kl} - z_r - z_s) \geq\ d_{ils} + \widetilde{p}_{ikr}z_r\} \geq \lambda_{klr},\ k,l \in K,\ r \in R_k \tag{60}$$
$$s \in R_l\ i \in I_k \bigcap I_l\ : k < l\ and\ |J_i| = 1,$$

$$Cr\{C_{ils} + Q(3 - y_{kl} - z_r - z_s) \geq\ d_{ikr} + \widetilde{p}_{ils}z_s\} \geq \Omega_{klr},\ k,l \in K,\ r \in R_k \tag{61}$$
$$s \in R_l\ i \in I_k \bigcap I_l\ : k < l\ and\ |J_i| = 1,$$

$$Cr\{C_{ikr} + Q(2 + y_{kl} - x_{ijkr} - x_{ijls}) \geq\ d_{ils} + \widetilde{p}_{ikr}z_r\} \geq \Lambda_{klr}, \tag{62}$$
$$k,l \in K,\ r \in R_k, s \in R_l\ i \in I_k \bigcap I_l\ : j \in J_i : k < l\ and\ |J_i| > 1$$

$$Cr\{C_{ils} + Q(3 - y_{kl} - x_{ijkr} - x_{ijls}) \geq\ d_{ikr} + \widetilde{p}_{ils}z_s\} \geq \hbar_{klr}, \tag{63}$$
$$k,l \in K, r \in R_k, s \in R_l\ i \in I_k \bigcap I_l\ j \in J_i :\ k < l\ and\ |J_i| > 1$$

$$Cr\{C_{i_rkr} = d_{(i-1)r} + \widetilde{p}_{i_rkr}z_r\} \geq \Psi_{ikr}\ k \in K,\ r \in R_k, (i-1)_r, \tag{64}$$
$$i_r \in I_r :\ i_r > f_r$$

$$Cr\{C_{max} \geq \sum_{k\in K}\sum_{r\in R_k} \leq \frac{\widetilde{p}_{ikr}z_r}{|J_i|} + \sum_{h\in I: h\neq i} (min_{k\in K,\ r\in R_k}\ (\widetilde{p}_{hkr}))\} \geq \varphi_i\ i \in I; \tag{65}$$

$$Cr\{T_k = max_k\{0, C_k - \widetilde{D}_k\}\} \geq \omega_k\ k \in K \tag{66}$$

$$Cr\{\sum_{i=1}^{m} c_i \times \ n_i \leq \widetilde{B}_1\} \geq \kappa_1 \tag{67}$$

$$Cr\{\sum_{k=1}^{v} \sum_{r \in R_k} \widetilde{C}_{rk} \times z_r \leq \widetilde{B}_2\} \geq \kappa_2 \tag{68}$$

Constraints (2), (10)-(15), (18), (21)-(29).

In the CCPM formulation, equations (55), (56) are applied in order to convert the chance constraints into their crisp equivalent ones.

## 8. **Dependent Chance-Constrained Programming Model (DCPM)**

Sometimes the decision maker may hope that the total weighted tardiness time does not exceed a given level $TWT$ for the sake of customer satisfaction. That is, the credibility of fuzzy event should be maximized. In order to formulate this type of a decision system, Liu [27-28] provided a theoretical framework of fuzzy dependent-chance programming (DCP), where the underlying philosophy is based on selecting the decision with maximal credibility to meet the event. From this view, we can present a credibility maximization model for our problem as follows: Constraint

$$\begin{aligned} &\text{Max} \\ &Cr\{f(x,\xi) \leq f_0\}. \\ &\text{s.t} \end{aligned} \tag{69}$$

Constraint (54).

In this approach maximizing the credibility level of objective function $f$ is the main aim even if the objective function increases in much more value. The respective constraints like the CCP model try to satisfy the lease required credibility level of each chance constraint [40].

In this way, the definition of $\alpha$-critical value for this type of problem is also applied, except here ? is equal to one. With the aforementioned comment, the third crisp equivalent model of original fuzzy model is as follows. Note that in this mathematical model, all of the lower bounds of credibility measures are fixed to uppermost values. Again, for converting the chance constraints to their crisp equivalent ones, equations (55) and (56) are applied.

$$\begin{aligned} &\text{Max} \\ &Cr\{\sum_{k=1}^{v} W_k \times T_k \leq f_0\} \end{aligned} \tag{70}$$

Constraints (2), (10)-(15), (18), (21)-(29), (59)-(70).

## 9. **Proposed Hybrid Meta-Heuristic Algorithm**

To solve these three crisp models, a hybrid meta-heuristic algorithm is proposed, by which a genetic algorithm is designed to compute the number of processors in each stage. Next, a particle swarm optimization (PSO) algorithm is applied to obtain the optimal value of tardiness variables. The process of using the designed

GA and PSO is continued interactively until the termination criterion is satisfied. Briefly, the main mechanisms of these algorithms are presented below.

## 10. **Particle Swarm Algorithm**

Particle swarm algorithm (PSO) is an evolutionary computation technique developed by Eberhart and Kennedy [9]. Two years later Salerno [46] used PSO on a number of neural model architectures solving the XOR problem and then applied that to a real problem. Shi and Eberhart [48] carried out the developments, applications, and resources related to PSO in the area of engineering and computer science. The PSO structure is similar to the genetic algorithm (GA), which is initialized with a population of random solutions and searched for the optima by adapting successive generations. For the optimization process, each particle is stimulated with a random position ($X_{id}$), velocity ($V_{id}$), and a fitness value evaluated with a predefined fitness function. Then, the fitness value of each particle is updated based on the local best value ($P_{id}$) and global best value ($P_{gd}$). The PSO method is formulated as follows. The initial position of $x_{id}$ is obtained by:

$$x_{id} = min\ x_{id} + r \times (max\ x_{id} - min\ x_{id}) \tag{71}$$

where $r$ is a uniform random variable between [0, 1] and min and mix represent the maximum and minimum limits of $x_{id}$.

This equilibrium or relation assures the covering of all feasible solution.

$$v_{id}^{new} = v_{id}^{old} + C_1 \times (rand_1) \times (p_{id} - x_{id}) + C_2 \times (rand_2) \times (p_{gd} - x_{id}) \tag{72}$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \tag{73}$$

$$If\ V_{id} > V_{max},\ V_{id} = V_{max}\ else\ V_{id} < V_{max},\ V_{id} = -Vmax \tag{74}$$

where $C_1$ and $C_2$ are two positive constants, $i$ represent the number of particles and $d$ is the dimension in the solution space. $V_{max}$ is a parameter called the maximum velocity specified by users. Particles' velocities on each dimension are fixed to the $V_{max}$. When the sum of accelerations causes the velocity on that dimension to exceed the $V_{max}$, the velocity on that dimension is limited to the $V_{max}$. Therefore, when $V_{max}$ is set too high, particles may raise past the excellent solution; when $V_{max}$ is set too low, particles may not explore sufficiently beyond locally superior regions [41]. In this paper, our proposed PSO is used for attaining the optimal sequence of jobs to be processed on the machines located in various stages, and also for selecting the optimal route considered for each product simultaneously. Figure 4 shows the structure of one particle used in PSO algorithm.

In this regard, both factors are dependent on the number of jobs. For routes, the minimum and maximum values are fixed to one and four values, respectively. Furthermore, for continuity, $r_{min}$ and $r_{max}$ are considered as follows.

$$r_{min} = 0.5 \tag{75}$$

$$r_{max} = 4 + 0.5 - epsilon \tag{76}$$

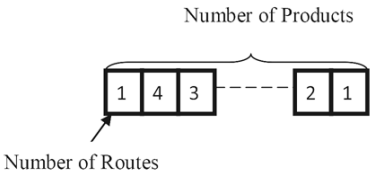



FIGURE 4. The Structure of One Particle Used in the PSO Algorithm

Also, $v_{max}$ is considered equal to the following equation:

$$v_{max} = \frac{r_{max} - r_{min}}{10} \tag{77}$$

To procreate the initial solution, the primary sequence is regarded as a rearrangement, which is equal to the number of jobs. The initial route is set for each job equal to the following term as the same as equation (73).

$$r = r_{min} + (r_{max} - r_{min}) * rand(1, nvar) \tag{78}$$

Learning rates, C1 and C2, which are the indicators of individual learning and society learning, are, set to 2. Noteworthy, the above values for PSO parameters have been selected based upon a number of initial tests conducted on the problem. Regarding the above-mentioned comments the main loop of this algorithm is as follows. In iteration 1, the velocity associated with the routes is set to zero and the cost associated with each particle is calculated. The best memory of each particle is set equal to the current particle cost and particle position. When the cost of one particle is lower than the previous one, the best particle solution memory is updated. Also, when the best cost of all particles is to be lower than the previous one, the best global solution memory is updated. Using equation (76), the particle velocity related to the production routes is updated. Also, for controlling the considered velocity, the following alternative equation, equation (78), is used.

$$particle(i).velocity.r = min(max(particle(i).velocity.r, -vr_{max}), vr_{max}) \tag{79}$$

Using equation (75), the particle position related to the production routes is updated. Furthermore, for controlling the considered position, the following equation is utilized.

$$particle(i).position.r = min(max(particle(i).position.r_{min}), r_{max}) \tag{80}$$

With respect to the aforementioned points, the cost associated with each particle containing the memories of the best sequence and the best route is calculated. In these circumstances, if the cost associated with each particle is reduced, the best memory of this particle is updated. Also, if the global best cost associated with all of these particles is reduced, the best global memory of all of those particles is updated. This process continues iteratively until the termination criterion is met.

## 11. **Genetic Algorithm**

Genetic algorithm (GA) was first proposed by John Holland in the 1960s [14] and developed by Holland and his students as well as some colleagues at the University of Michigan in the 1960s and the 1970s [14, 13]. Holland [15] presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA [12, 33]. Like any other optimization algorithm, the GA begins by defining the solution string and objective (cost) function and it ends by testing for convergence. However, this algorithm is quite different from other algorithms. The components path of the GA is depicted in Figure 5. A cost

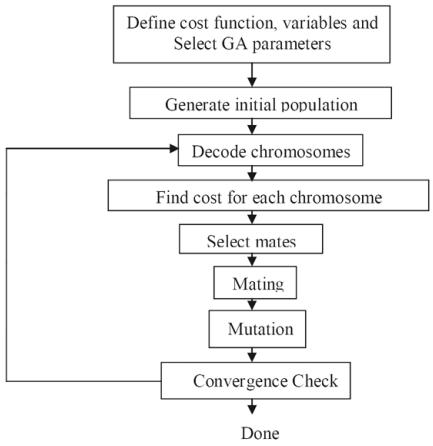



FIGURE 5. The Steps of the Proposed Genetic Algorithm

function generates an output from a set of input variables. This function may be a mathematical function, an experiment or a game. The object is to modify the output in some desirable fashion by finding the appropriate values for the input variables. So it can be observed that determining an appropriate cost function and variables to use are intimately related. The term fitness is extensively used to designate the output of the objective function in the GA literature. Fitness denotes a maximization problem. Although fitness has a closer association with biology than the term cost, the GA begins by definition of a chromosome or an array of variable values to be optimized. If the chromosome has $N_{var}$ variables (an $N_{var}$-dimensional optimization problem) given by $p_1, p_2, ..., P_{Nvar}$ , then the chromosome is written as an $N_{var}$ element row vector. In this paper, the genetic algorithm is used to calculate the number of processors located in each stage. The method is used to calculate the high and low levels of the number of the processors. In this regards, it is assumed that the number of processors per stage is at least one and at most equal to the maximum of the total investment divided by each processor cost. To produce the random values of processors, the following equation is used.

$$x = x_{min} + (x_{max} - x_{min}) \times rand(npop, nvar) \tag{81}$$

Figure 6 shows the string structure used in GA algorithm.

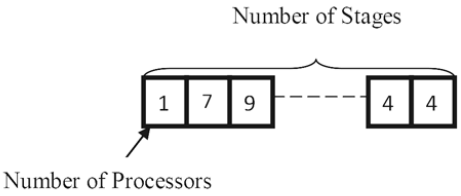



FIGURE 6. The Structure of One String Used in the GA Algorithm

Regarding to equation, $x_min$ and $x_max$ values are set to the values defined above, and the random function generates values between zero and one with dimensions, *npop* rows and *nvar* columns. In this way, the related values always fall in the range between the maximum and minimum number of processors. To become more realistic these values are rounded. Thus, with these defined values, the main loop of the algorithm begins. Accordingly, the cost per repetition is regularly calculated and the top members of the population with the appropriate values are selected, so that at each iteration, better values are replaced with the previous ones. Also, to calculate the amount of fitness and adaptation, the following equation is used.

$$f = exp(-beta * costs) \tag{82}$$

The cost values used in this relation are normal values and beta is a constant value. Also, in order to calculate the probabilities of selections, the following equation is used.

$$p = \frac{f}{sum(f)} \tag{83}$$

These values are used to rotate the roulette wheel in the selection process of the crossover operator. Regarding aforementioned comments, two parents are selected randomly and the crossover operator is performed. The following equations define the crossover operator to generate the new children to be added to the population.

$$child1 = alpha \times p_1 + (1 - alpha) \times p_2 \tag{84}$$

$$child2 = alpha \times p_2 + (1 - alpha) \times p_1 \tag{85}$$

In this equation, *alpha* is a random number between 0 and 1, and *pi* is a representative parent. Also, in the mutation operator, one chromosome and one gene are selected at random. The random values between the minimum and maximum amounts are produced, and this new chromosome is added to the original population. Now, the new population consists of the main population, and the new children generated by the mutation and crossover operators. Now, after observing the fitness value (cost) of each member of the new population, the superior members with a predefined size are selected. This process continues repeatedly until the termination criterion is met.

## 12. **Computational Results**

In this section, the three crisp equivalent models are solved and compared with each other in terms of both objective function value and required computational time. Moreover, to show the behavior of these three approaches interacted by the proposed meta-heuristic algorithm, their convergence are presented by three figures separately. The MATLAB software is used to run the proposed meta-heuristic algorithm. After running some initial tests, the termination criteria (i.e., the maximum iterations) of GA and PSO meta-heuristics are set as 120 and 10, respectively. The general attributes of the given problems are shown in Table 1. Also, the fuzzy parameters are shown in the Tables 2 to 4.

| | |
|---|---|
| Number of stages | 6 |
| Number of products | 20 |
| Number of routes | 4 |

TABLE 1. Overall Attributes of Test Problem

The computational results of the three crisp equivalent models, such as tardiness time, completion times and so on, have been represented through Tables 5 to 10. Notably, for the chance-constrained programming model, the lower bounds of credibility measures are fixed to equal values. In this test problem for all constraints that has fuzzy parameters, a confidence level between [0, 1] can be incorporated into our computation. In this study an arbitrary value of 0.3 was used to examine the solutions.

As shown in the tables presented above, it can clearly be observed that the number of processors is the same for the two methods. In other words, the genetic algorithm has similar behavior for all three methods. Performing the PSO algorithm, compared to the expected value and chance-constrained programming methods, the dependent chance-constrained programming produces a lower value of tardiness and the expected value and chance-constrained programming methods come next, respectively.

As for the selected routes for processing of parts, different methods give different results, each route yielding different cost values. Also succession of parts entering into the processors is different for the methods concerned.

## 13. **Sensitivity Analysis**

In this section, some sensitivity analyses are carried out. Four important factors, i.e., processing time, route cost, purchasing budget and selection budget, are considered and their impact on the objective function value (O.F.V) is analyzed. In addition, the number of population and the maximum iteration for running the genetic algorithm are fixed to 40 and 20, respectively. Figures 7 to 9 show that with increasing the processing time, the objective function value is also increased, and each crisp model has its specific optimum objective point. The solution deviation may be a result of randomness environment.

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Product** | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| **Weight** | | | 1 | 2 | 3 | 2 | 2 | 5 | 5 | 3 | 1 | 1 | 1 | 2 | 3 | 4 | 6 | 7 | 3 | 2 | 1 | 1 |
| **Due Date** | **DU.a** | | 12 | 13 | 18 | 1 | 4 | 3 | 17 | 15 | 9 | 2 | 1 | 10 | 8 | 4 | 4 | 5 | 1 | 12 | 1 | 16 |
| | **DU.b** | | 27 | 19 | 20 | 21 | 5 | 3 | 27 | 17 | 13 | 3 | 31 | 4 | 14 | 34 | 7 | 10 | 8 | 15 | 5 | 17 |
| | **DU.c** | | 36 | 23 | 24 | 27 | 19 | 34 | 30 | 28 | 27 | 33 | 19 | 32 | 20 | 25 | 25 | 16 | 21 | 15 | 27 | 19 |
| | **DU.d** | | 38 | 23 | 36 | 40 | 30 | 37 | 36 | 38 | 31 | 36 | 30 | 35 | 38 | 37 | 18 | 32 | 26 | 31 | 33 | 31 |
| **Processing Time** | **Stage** | **P.a** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 0 | 2 | 1 | 1 | 0 | 3 | 0 | 1 | 4 | 2 | 0 | 0 | 1 | 1 | 0 | 4 | 4 | 0 | 5 | 0 |
| | **2** | | 6 | 3 | 0 | 1 | 1 | 4 | 5 | 2 | 1 | 1 | 3 | 0 | 0 | 7 | 0 | 4 | 3 | 5 | 1 | 2 |
| | **3** | | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 4 | 0 | 1 | 2 | 1 | 2 | 0 | 0 | 2 | 1 | 2 | 1 | 1 |
| | **4** | | 6 | 1 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 1 | 2 | 0 |
| | **5** | | 1 | 3 | 2 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 4 | 0 | 3 | 1 | 2 | 1 | 5 | 2 | 6 | 5 |
| | **6** | | 7 | 0 | 2 | 6 | 1 | 0 | 2 | 0 | 1 | 0 | 4 | 3 | 1 | 0 | 1 | 1 | 0 | 2 | 2 | 0 |
| | **Stage** | **P.b** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 0 | 6 | 2 | 1 | 6 | 5 | 3 | 1 | 6 | 7 | 3 | 5 | 2 | 7 | 1 | 5 | 5 | 3 | 6 | 2 |
| | **2** | | 7 | 6 | 2 | 1 | 6 | 5 | 8 | 3 | 4 | 5 | 5 | 2 | 8 | 7 | 0 | 5 | 4 | 6 | 1 | 3 |
| | **3** | | 0 | 1 | 0 | 1 | 2 | 6 | 8 | 4 | 3 | 4 | 3 | 3 | 3 | 0 | 1 | 4 | 4 | 2 | 5 | 2 |
| | **4** | | 7 | 3 | 1 | 1 | 6 | 6 | 3 | 2 | 3 | 6 | 1 | 1 | 4 | 1 | 6 | 4 | 0 | 3 | 3 | 4 |
| | **5** | | 3 | 4 | 4 | 3 | 7 | 1 | 6 | 2 | 5 | 3 | 5 | 5 | 3 | 2 | 4 | 2 | 6 | 3 | 6 | 5 |
| | **6** | | 8 | 4 | 4 | 8 | 2 | 4 | 6 | 3 | 2 | 5 | 4 | 6 | 2 | 4 | 3 | 3 | 0 | 3 | 3 | 2 |
| | **Stage** | **P.c** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 2 | 7 | 6 | 2 | 9 | 8 | 5 | 3 | 9 | 8 | 5 | 6 | 4 | 7 | 5 | 6 | 7 | 8 | 7 | 4 |
| | **2** | | 7 | 7 | 2 | 7 | 8 | 7 | 9 | 7 | 5 | 6 | 6 | 3 | 8 | 8 | 1 | 9 | 8 | 7 | 3 | 7 |
| | **3** | | 6 | 9 | 5 | 6 | 5 | 8 | 8 | 5 | 7 | 5 | 3 | 4 | 8 | 3 | 6 | 8 | 6 | 4 | 5 | 8 |
| | **4** | | 8 | 5 | 1 | 5 | 7 | 8 | 5 | 3 | 4 | 7 | 4 | 2 | 6 | 4 | 8 | 6 | 1 | 4 | 3 | 8 |
| | **5** | | 4 | 7 | 4 | 8 | 8 | 2 | 6 | 6 | 9 | 5 | 9 | 5 | 8 | 6 | 6 | 3 | 7 | 9 | 8 | 8 |
| | **6** | | 8 | 5 | 5 | 9 | 5 | 5 | 7 | 5 | 8 | 8 | 6 | 8 | 4 | 7 | 4 | 8 | 2 | 5 | 4 | 4 |
| | **Stage** | **P.d** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 4 | 8 | 7 | 3 | 9 | 9 | 7 | 4 | 9 | 9 | 7 | 9 | 6 | 7 | 5 | 6 | 8 | 9 | 7 | 8 |
| | **2** | | 9 | 8 | 4 | 9 | 9 | 9 | 9 | 8 | 5 | 6 | 6 | 4 | 9 | 9 | 4 | 9 | 8 | 9 | 8 | 9 |
| | **3** | | 9 | 9 | 9 | 7 | 7 | 8 | 8 | 6 | 8 | 5 | 8 | 7 | 9 | 8 | 7 | 9 | 7 | 5 | 7 | 8 |
| | **4** | | 9 | 6 | 1 | 6 | 8 | 8 | 7 | 5 | 9 | 8 | 6 | 7 | 9 | 6 | 9 | 8 | 2 | 8 | 8 | 9 |
| | **5** | | 9 | 8 | 6 | 8 | 9 | 5 | 9 | 9 | 9 | 7 | 9 | 7 | 8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| | **6** | | 9 | 5 | 9 | 9 | 8 | 9 | 9 | 6 | 9 | 8 | 9 | 9 | 9 | 7 | 4 | 8 | 2 | 6 | 4 | 5 |
| **Routes** | **1** | | | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | | | 1 | | 1 | | | 1 | 1 | 1 |
| | **2** | | | 1 | 1 | 1 | | 1 | | 1 | 1 | | 1 | | | | 1 | 1 | | 1 | | |
| | **3** | | 1 | 1 | | | 1 | 1 | | 1 | 1 | | | 1 | 1 | | 1 | | 1 | 1 | 1 | |
| | **4** | | 1 | 1 | 1 | | 1 | | 1 | 1 | | | | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 |
| **Route Cost** | **Route** | **RC.a** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 1 | 3 | 5 | 4 | 3 | 1 | 2 | 6 | 4 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 4 | 1 | 1 | 1 |
| | **2** | | 3 | 2 | 2 | 2 | 3 | 3 | 5 | 3 | 2 | 1 | 2 | 1 | 2 | 2 | 7 | 1 | 5 | 6 | 1 | 4 |
| | **3** | | 4 | 1 | 4 | 3 | 3 | 3 | 5 | 3 | 3 | 4 | 1 | 2 | 2 | 5 | 3 | 1 | 1 | 2 | 2 | 2 |
| | **4** | | 3 | 2 | 1 | 2 | 3 | 4 | 2 | 7 | 1 | 2 | 2 | 2 | 2 | 4 | 2 | 1 | 2 | 2 | 2 | 2 |
| | **Route** | **RC.b** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 1 | 7 | 7 | 6 | 3 | 4 | 3 | 7 | 8 | 5 | 3 | 1 | 6 | 6 | 4 | 4 | 6 | 3 | 2 | 6 |
| | **2** | | 7 | 7 | 4 | 3 | 4 | 6 | 5 | 8 | 3 | 4 | 2 | 1 | 3 | 3 | 8 | 4 | 5 | 7 | 5 | 5 |
| | **3** | | 7 | 2 | 6 | 4 | 3 | 5 | 8 | 3 | 4 | 7 | 4 | 5 | 5 | 7 | 7 | 4 | 1 | 2 | 3 | 3 |
| | **4** | | 3 | 6 | 5 | 2 | 3 | 4 | 3 | 7 | 5 | 3 | 5 | 4 | 4 | 5 | 3 | 2 | 7 | 4 | 7 | 3 |
| | **Route** | **RC.c** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 8 | 8 | 7 | 8 | 4 | 5 | 5 | 7 | 9 | 5 | 4 | 3 | 7 | 7 | 5 | 5 | 6 | 3 | 3 | 9 |
| | **2** | | 7 | 8 | 6 | 7 | 5 | 8 | 7 | 8 | 6 | 7 | 8 | 1 | 5 | 5 | 9 | 7 | 6 | 8 | 7 | 5 |
| | **3** | | 7 | 7 | 6 | 4 | 5 | 6 | 8 | 8 | 6 | 8 | 7 | 7 | 6 | 7 | 8 | 8 | 5 | 8 | 3 | 5 |
| | **4** | | 5 | 7 | 6 | 3 | 5 | 7 | 4 | 7 | 6 | 6 | 8 | 6 | 7 | 5 | 6 | 3 | 7 | 6 | 7 | 5 |
| | **Route** | **RC.d** | | | | | | | | | | | | | | | | | | | | |
| | **1** | | 9 | 9 | 8 | 8 | 9 | 6 | 9 | 9 | 9 | 6 | 9 | 4 | 9 | 7 | 9 | 6 | 7 | 4 | 9 | 9 |
| | **2** | | 9 | 9 | 8 | 8 | 9 | 6 | 9 | 9 | 9 | 6 | 9 | 4 | 9 | 7 | 9 | 6 | 7 | 4 | 9 | 9 |
| | **3** | | 7 | 9 | 7 | 9 | 6 | 7 | 8 | 9 | 9 | 8 | 8 | 9 | 6 | 7 | 8 | 9 | 5 | 9 | 8 | 5 |
| | **4** | | 6 | 9 | 6 | 4 | 9 | 8 | 7 | 9 | 7 | 8 | 8 | 7 | 7 | 9 | 9 | 5 | 9 | 7 | 9 | 9 |

TABLE 2. Parameters VS. Product

| **Stage** | | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| **Cost of Processor** | | 10 | 20 | 15 | 20 | 30 | 20 |
| **Routes** | **1** | 1 | | | 1 | | |
| | **2** | 1 | | | | 1 | 1 |
| | **3** | | 1 | 1 | 1 | | |
| | **4** | | 1 | 1 | | 1 | 1 |

TABLE 3. Parameters VS. Stage

Also, Figures 10 to 12 show the route cost impact on the objective function value. It seems that there is no important sensitivity, and the reason can be found in amount of this parameter. The solution deviation may be a result of randomness environment. Also, dependent chance-constrained programming has the minimum

| | | | |
|---|---|---|---|
| **B1.a** | 180 | **B2.a** | 80 |
| **B1.b** | 190 | **B2.b** | 90 |
| **B1.c** | 210 | **B2.c** | 100 |
| **B1.d** | 220 | **B2.d** | 110 |

TABLE 4. Budgets for Purchasing the Processor and Selecting Route

| **Stage** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **Number of Processors** | 7 | 10 | 7 | 6 | 4 | 4 |

TABLE 5. Number of the Obtained Processors

| **Product** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Scheduling Positions** | | 15 | 8 | 12 | 3 | 17 | 14 | 18 | 19 | 4 | 9 | 5 | 20 | 16 | 10 | 11 | 1 | 6 | 2 | 13 | 7 |
| **Tardiness Time** | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Completion Time** | **Stage** | | | | | | | | | | | | | | | | | | | | |
| | **1** | 0 | 0 | 0 | 1.8 | 7.8 | 0 | 0 | 0 | 7 | 6.5 | 3.8 | 0 | 0 | 0 | 0 | 5.3 | 0 | 5 | 6.3 | 0 |
| | **2** | 7.3 | 6 | 2 | 1.8 | 7.8 | 6.3 | 7.8 | 6.25 | 7 | 6.5 | 3.8 | 4.3 | 6.3 | 7.8 | 1.3 | 5.3 | 5.8 | 5 | 6.3 | 5.3 |
| | **3** | 14.3 | 10.8 | 5.5 | 1.8 | 7.8 | 12.3 | 15 | 15 | 7 | 6.5 | 3.8 | 13.8 | 11.8 | 10.5 | 4.8 | 5.3 | 10.3 | 5 | 6.3 | 10 |
| | **4** | 14.3 | 14.5 | 5.5 | 1.8 | 14.5 | 17.8 | 15 | 17.5 | 7 | 12.8 | 3.8 | 20.3 | 19.3 | 10.5 | 11.5 | 5.3 | 11 | 9 | 10.3 | 10 |
| | **5** | 18.5 | 14.5 | 16.8 | 6.8 | 14.5 | 17.8 | 20.5 | 17.5 | 12.8 | 12.8 | 10.5 | 20.3 | 19.3 | 15 | 11.5 | 9 | 11 | 9 | 10.3 | 16.8 |
| | **6** | 27.5 | 14.5 | 21.8 | 14.8 | 14.5 | 17.8 | 26.5 | 17.5 | 17.8 | 12.8 | 16.3 | 20.3 | 19.3 | 22.3 | 11.5 | 14 | 11 | 9 | 10.3 | 19.5 |
| **Routes** | **1** | | | | | 1 | | | | | 1 | | | | | | | | 1 | 1 | |
| | **2** | | | | 1 | | | | | 1 | | 1 | | | | | 1 | | | | |
| | **3** | | 1 | | | | 1 | | 1 | | | | 1 | 1 | | 1 | | 1 | | | |
| | **4** | 1 | | 1 | | | | 1 | | | | | | | 1 | | | | | | 1 |
| **Route Costs** | | 4.3 | 4.8 | 4.5 | 5.3 | 4.8 | 5.3 | 4 | 5.8 | 4.8 | 4.5 | 5 | 5.8 | 4.8 | 5.8 | 6.5 | 5.3 | 3 | 2.8 | 3.8 | 4.8 |

TABLE 6. Solutions Related to Products

| **Stage** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Number of Processors** | 7 | 10 | 7 | 6 | 4 | 4 |

TABLE 7. Number of the Obtained Processors

objective function value, because in this approach the maximum level has been generated. It can be anticipated that interaction between all of the parameters has large impact in this approach.

The impact of purchasing budget on O.F.V has been depicted through Figures 13 to 15. In the expected value and chance-constrained models, by increasing this impact, the objective function increases to some points then decreases. The solution deviation may be the result of a randomness environment. Also, dependent chance-constrained programming has the minimum objective function value.

Finally, the impact of selection budget on O.F.V has been depicted in Figures 16 to 18. It seems that there is no important sensitivity, and the reason can be found in amount of this parameter and interaction of this parameter with others.

| Product | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scheduling Positions | | 16 | 15 | 11 | 7 | 4 | 2 | 13 | 10 | 8 | 20 | 19 | 9 | 3 | 14 | 1 | 5 | 6 | 17 | 18 | 12 |
| Tardiness Time | | 1.2 | 3.6 | 0.0 | 1.2 | 4.2 | 4.6 | 0.0 | 0.0 | 0.6 | 21.2 | 17.2 | 0.0 | 0.0 | 2.0 | 0.0 | 2.2 | 1.2 | 3.0 | 15.2 | 1.2 |
| Completion Time | Stage 1 | 0 | 8.8 | 4.4 | 2.2 | 3.6 | 4.2 | 5.6 | 3.2 | 0 | 10.6 | 6.8 | 0 | 1.6 | 0 | 0 | 4.6 | 0 | 6.2 | 0 | 5 |
| | Stage 2 | 11 | 8.8 | 4.4 | 2.2 | 3.6 | 4.2 | 5.6 | 3.2 | 2.8 | 10.6 | 6.8 | 4 | 1.6 | 9.8 | 0 | 4.6 | 3.6 | 6.2 | 10.8 | 5 |
| | Stage 3 | 12 | 8.8 | 4.4 | 2.2 | 3.6 | 4.2 | 5.6 | 3.2 | 8.2 | 10.6 | 6.8 | 10 | 1.6 | 12 | 0.6 | 4.6 | 6.4 | 6.2 | 15.4 | 5 |
| | Stage 4 | 22 | 8.8 | 4.4 | 7 | 8 | 4.2 | 13 | 10 | 10 | 23.8 | 6.8 | 11 | 4 | 12 | 5.8 | 4.6 | 6.4 | 15 | 18.6 | 13 |
| | Stage 5 | 22 | 17 | 10 | 7 | 8 | 5.2 | 13 | 10 | 10 | 23.8 | 21.4 | 11 | 4 | 13 | 5.8 | 6.8 | 6.4 | 15 | 18.6 | 13 |
| | Stage 6 | 22 | 20 | 14 | 7 | 8 | 7.6 | 13 | 10 | 10 | 23.8 | 25.4 | 11 | 4 | 16 | 5.8 | 10 | 6.4 | 15 | 18.6 | 13 |
| Routes | 1 | | | | 1 | 1 | | 1 | 1 | | 1 | | | 1 | | | | | 1 | | 1 |
| | 2 | | 1 | 1 | | | 1 | | | | | 1 | | | | | 1 | | | | |
| | 3 | 1 | | | | | | | | 1 | | | 1 | | | 1 | | 1 | | 1 | |
| | 4 | | | | | | | | | | | | | | 1 | | | | | | |
| Route Costs | | 5.8 | 5 | 3.2 | 5.2 | 3 | 4.8 | 2.6 | 6.6 | 3.6 | 3.8 | 2 | 3.8 | 4.4 | 4.6 | 5.4 | 2.8 | 1 | 2.2 | 2.6 | 4 |

TABLE 8. Solutions Related to Products

| Stage | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number of Processors | 4 | 4 | 5 | 5 | 3 | 3 |

TABLE 9. Number of the Obtained Processors

| Product | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scheduling Positions | | 17 | 2 | 5 | 14 | 12 | 19 | 18 | 13 | 7 | 16 | 4 | 10 | 20 | 8 | 1 | 3 | 9 | 15 | 6 | 11 |
| Tardiness Time | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Completion Time | Stage 1 | 0 | 8 | 0 | 11 | 0 | 0 | 22 | 11 | 15 | 20 | 7 | 0 | 0 | 0 | 0 | 6 | 0 | 20 | 7 | 15 |
| | Stage 2 | 17 | 8 | 4 | 11 | 13 | 17 | 22 | 11 | 15 | 20 | 7 | 8 | 18 | 9 | 4 | 6 | 8 | 20 | 7 | 15 |
| | Stage 3 | 26 | 8 | 13 | 11 | 20 | 25 | 22 | 11 | 15 | 20 | 7 | 15 | 27 | 17 | 11 | 6 | 15 | 20 | 7 | 15 |
| | Stage 4 | 35 | 8 | 13 | 21 | 28 | 33 | 35 | 11 | 24 | 32 | 7 | 15 | 37 | 17 | 20 | 6 | 17 | 28 | 15 | 24 |
| | Stage 5 | 35 | 16 | 22 | 21 | 28 | 33 | 35 | 31 | 24 | 32 | 16 | 23 | 37 | 26 | 20 | 15 | 17 | 28 | 15 | 24 |
| | Stage 6 | 35 | 21 | 31 | 21 | 28 | 33 | 35 | 37 | 24 | 32 | 25 | 34 | 37 | 33 | 20 | 23 | 17 | 28 | 15 | 24 |
| Routes | 1 | | | | 1 | | | 1 | | 1 | 1 | | | | | | | | 1 | 1 | 1 |
| | 2 | | 1 | | | | | | 1 | | | 1 | | | | | 1 | | | | |
| | 3 | 1 | | | | 1 | 1 | | | | | | | 1 | | 1 | | 1 | | | |
| | 4 | | | 1 | | | | | | | | | 1 | | 1 | | | | | | |
| Route Costs | | 7 | 8 | 6 | 8 | 6 | 7 | 9 | 9 | 9 | 6 | 8 | 7 | 6 | 9 | 8 | 9 | 5 | 4 | 9 | 9 |

TABLE 10. Solutions Related to Products

The solution deviation may be a result of randomness environment. Also, dependent chance-constrained programming has the minimum objective function value.

Related to dependent chance-constrained programming, it can be interpreted that in this method a level of maximum credibility is taken into consideration. Thus, the levels of credibility for all constraints are taken as 1.

In this manner, those constraints more influential on other constraints are considered as important factors in the final results. In other words, the effects of the parameters and constraints values, act as important factors.

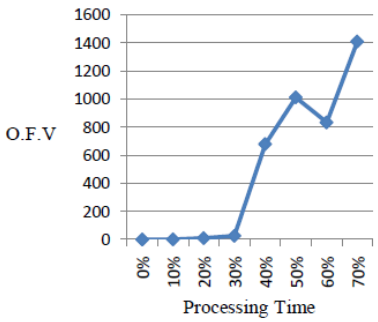



FIGURE 7. The Expected Value Approach Considering the Processing Time Impact

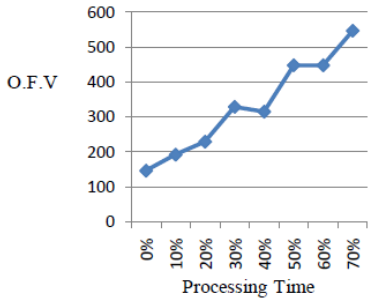



FIGURE 8. The Chance Constrained Approach Considering the Processing Time Impact

## 14. **Convergence Graphs**

The convergence rates of three equivalent crisp models have been shown in Figures 19-21, respectively. It is evident that the chance-constrained programming approach with the *alpha*-critical value equal to 0.3 has the best solution. As depicted in the convergence figures, the genetic algorithm arrives to a convergence after 120 iterations. All the three methods show similar behavior in this respect. As for the converged values, the chance-constrained programming method produces the least value. The reason is that the capital represented by a trapezoidal membership function, gets limited at a certain bound, thus limiting the used up capital at a certain amount. As for the costs of purchasing, the expected value and dependent chance-constrained programming introduce the second and third rank respectively. It can be said that, in the expected value method, the trapezoidal membership

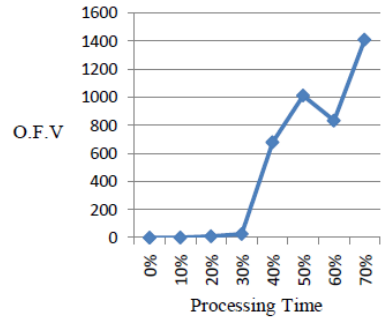



FIGURE 9. The Dependent Chance Constrained Approach Considering the Processing Time Impact

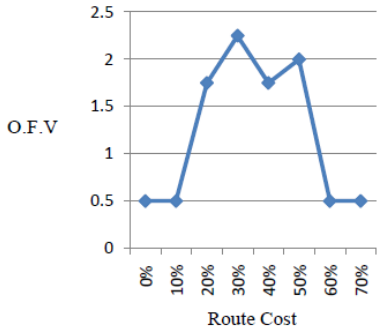



FIGURE 10. The Expected Value Approach Considering the Route Cost Impact

function gets it's bound at the middle of the trapezoidal, while for the dependent chance-constrained programming method the whole of the trapezoidal membership function is being used. Thus, it is clearly observed that in dependent chance-constrained programming method, we access to the full amount of capital, in the expected value method to the one half of the capital, and in the chance-constrained programming method to the 0.3 of the capital.

## 15. **Conclusions**

This paper deals with a new model for a flexible flow line problem with the budget factor. Because in the real world, some parameters (e.g., processing times and due dates) are imprecise in nature, they are treated as fuzzy parameters. In this way, a fuzzy mathematical model is first developed, and then, three crisp

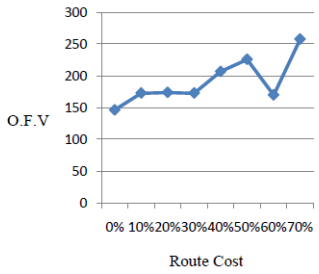



FIGURE 11. The Chance Constrained Approach Considering

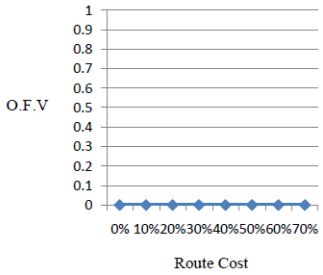



FIGURE 12. The Dependent Chance Constrained Approach Considering the Route Cost Impact

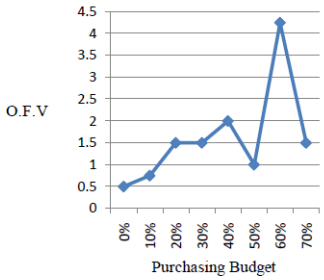



FIGURE 13. The Expected Value Approach Considering the Purchasing Budget Impact

equivalent models, i.e., the expected value, chance-constrained programming and dependent chance-constrained programming models are introduced. Furthermore, to solve these crisp models, a hybrid meta-heuristic algorithm integrating a genetic algorithm with a particle swarm optimization is proposed. The computational results show that dependent chance-constrained programming can obtain the lowest

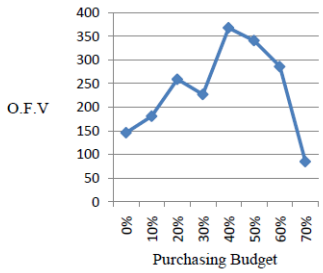



FIGURE 14. The Chance Constrained Approach Considering the Purchasing Budget Impact

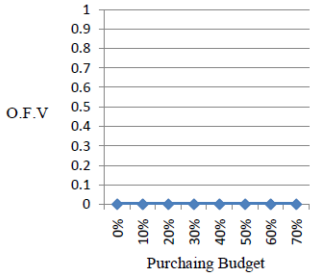



FIGURE 15. The Dependent Chance Constrained Approach Considering the Purchasing Budget Impact

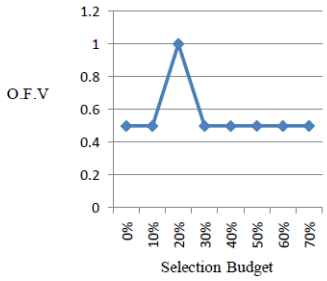



FIGURE 16. Expected Value Approach Considering the Selection Budget Impact

objective function value. The reason behind this fact is that when we select the dependent chance-constrained programming model, we actually select the uppermost level of fuzzy parameters. It can be anticipated that if we set parameters in other different values, we can expect different results because of interaction between paprameters in the mathematical model. After all, Concerning the value of

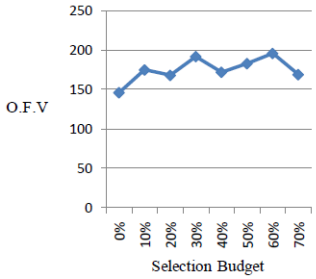



FIGURE 17. The Chance Constrained Approach Considering the Selection Budget Impact

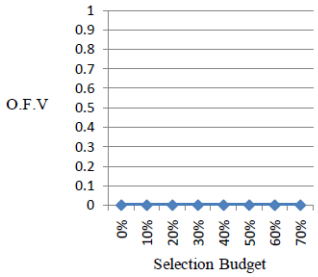



FIGURE 18. The Dependent Chance Constrained Approach Considering The Selection Budget Impact

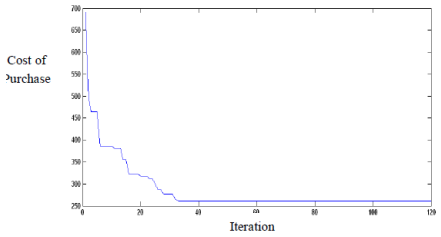



FIGURE 19. The Expected Value Approach

the objective function, one can use the dependent chance-constrained method in the first rank, the expected value approach and chance-constrained method in the second and third rank respectively.

It can be implied that if we allocate the higher credibility to the fuzzy constraints we concentrate on upper points of a trapezoidal membership function. In other words, if we set $\alpha = 1$, the dependent chance-constrained programming is arisen; otherwise, the chance-constrained programming is formed. It is obvious

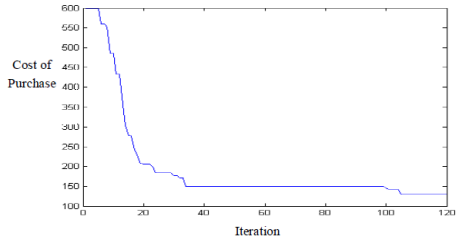



FIGURE 20. The Chance Constrained Programming Approach

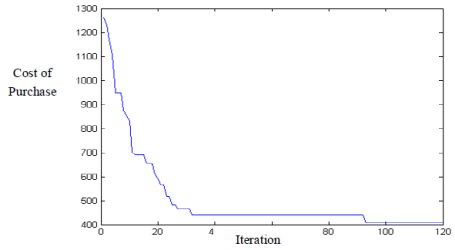



FIGURE 21. The Dependent Chance Constrained Programming Approach

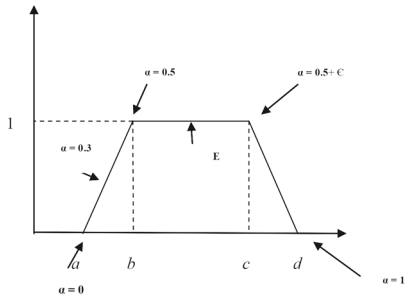



FIGURE 22. Relationship Between the Lower Limit of Credibility Measure and Points of the Trapezoidal Membership Function

that if we consider dependent chance-constrained programming, we minimize the upper limit of the fuzzy tardiness variable; therefore, we expect a higher objective function. However, due to the interaction of fuzzy processing times and due dates, the fuzzy tardiness time results in a lower objective function value in the expected value approach. Figure 22 clarifies these points. The convergence rates related to the GA, which computes the number of processors in each stage with the lowest

cost (i.e., purchasing cost), imply that the proposed meta-heuristic algorithm has the capability to come closer to the optimal solution. In this figure, the minimum objective function value is reported for the expected value approach; however, the minimum computational time is reported for dependent chance-constrained programming although with a higher objective function. Also, for further research, considering other membership functions and comparing the new results with those of trapezoidal membership function found here will be another challenging direction.

## References


[1] S. Abbasbandy and M. Alavi, *A method for solving fuzzy linear systems*, Iranian Journal of Fuzzy Systems, **2(2)** (2005),37-43.

[2] H. Allaoui and A. Artiba, *Scheduling two-stage hybrid flow shop with availability constraints*, Computers and Operations Research, **33(5)** (2006),1399-1419.

[3] V. A. Amentano and D. P. Ronconi, *Tabu search for total tardiness minimization in flow shop scheduling problems*, Computers and Operations Research, **26(3)** (1999), 219-235.

[4] M. Asano and H. Ohta, *Single machine scheduling using dominance relation to minimize earliness subject to ready and due times*, Production Economics, **64(1-3)** (1996), 101-111.

[5] V. Botta-Genoulaz, *Hybrid flow-shop scheduling with precedence constrains and time legs to minimize maximum lateness*, Production Economics, **64(1-3)** (2000), 101-111.

[6] J. M. Cadenas and J. L. Verdegay, *A primer on fuzzy optimization models and methods*, Iranian Journal of Fuzzy Systems, **3(1)** (2006), 1-21.

[7] J. Chang, W. Yan and H. Shao, *Scheduling a two stage no wait hybrid flow shop with separated setup and removal times*, Proceedings of the American Control Conference, Boston, MA, United states, **2(2)** (2004), 1412-1416.

[8] R. C. Eberhart and J. Kennedy, *A new optimizer using particle swarm theory*, Proceedings of the sixth International Symposium on Micromachine and Human Science, (1995), 9-43.

[9] M. B. Fakhrzad and M. Heydari, *Flexible flow-lines model at m machine centers with fuzzy total costs*, Applied Sciences, **8(11)** (2008), 2059-2066.

[10] M. Hapke and R. Slowinski, *Fuzzy priority heuristic for project Scheduling*, Fuzzy Sets and Systems, **83(3)** (1996),291-299.

[11] J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, (1975), Second Edition: MIT Press, 1992.

[12] J. Holland, *Hierarchical descriptions, universal spaces, and adaptive systems*, In: Arthur W. Burks, Editor. Essays on Cellular Automata. University of Illinois Press, 1970.

[13] J. Holland, *Iterative circuit computers*, In: Proc. Western Joint Comp. Conf, (1960), 259-265.

[14] J. Holland, *Outline for a logical theory of adaptive systems*, JACM, **9(3)** (1962), 279-314.

[15] X. Huang, *Chance-constrained programming models for capital budgeting with NPV as fuzzy parameters*, Computational and Applied Mathematics, **198** (2007), 149-159.

[16] X. Huang, *Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion*, Production Economics, **105(2)** (2007), 407-424.

[17] Z. Jin, Z. Yang and T. Ito, *Methaheuristics algorithms for the multi stage hybrid flow shop scheduling problem*, Production Economics, **100(2)** (2006), 322-334.

[18] M. Kolonko, *Methaheuristics algorithms for the multi stage hybrid flow shop scheduling problem*, European Journal of Operational Research, **113(1)** (1999), 123-136.

[19] T. Konno and H. Ishii, *An open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint*, Fuzzy Sets and Systems, **109(1)** (2000), 141-147.

[20] M. E. Kurz and R. G. Askin, *Comparing scheduling rules for flexible flow lines*, Production Economics, **85(3)** (2003), 371-388.

[21] G. J. Kyparisis and C. Koulamas, *A note on weighted completion time minimization in a flexible flow shop*, Operation Research Letter, **29(1)** (2001), 5-11.

[22] H. T. Lin and C. Liao, *A case study in a two-stage hybrid flow-shop with set up time and dedicated machines*, Production Economics, **86(2)** (2003), 133-143.
[23] M. Litoiu and R. Tadei, *Real-time task scheduling with fuzzy deadlines and processing times*, Fuzzy Sets and Systems, **117(1)** (2001), 34-45.
[24] B. Liu and K. Iwamura, *Chance-constrained programming with fuzzy parameters*, Fuzzy Sets and Systems, **94** (1998), 227-237.
[25] B. Liu and Y. K. Liu, *Expected value of fuzzy variable and fuzzy expected value models*, IEEE Transactions on Fuzzy Systems, **10** (2002), 445-450.
[26] B. Liu, *Dependent-chance programming with fuzzy decisions*, IEEE Transactions on Fuzzy Systems, **7(3)** (1999), 354-360.
[27] B. Liu, *Uncertainty Theory: An Introduction to its Axiomatic Foundations*, Springer, Berlin, 2004.
[28] C. Y. Liu and S. C. Chang, *Scheduling flexible flow Shops with sequence-dependent set up effects*, IEEE Transactions. Robotics Automation, **16(1)** (2000), 408-419.
[29] R. Logendran, S. Carson and E. Hanson, *Group scheduling in flexible flow shops*, Production Economics, **96(2)** (2005), 143-155.
[30] C. S. McCahon and E. S Lee, *Job sequencing with fuzzy processing times*, Computers Mathematics Application, **19(7)** (1990), 31-41.
[31] E. Mehdizadeh, S. Sadi-Nezhad and R. Tavakkoli-Moghaddam, *Optimization of fuzzy clustering criteria by a hybrid pso and fuzzy c-means clustering algorithm*, Iranian Journal of Fuzzy Systems, **5(3)** (2008), 1-14.
[32] H. Nezamabadi-Pour, S. Yazdani-Sharbabaki, M. M. Farsangi and M. Neyestani, *A solution to an economic dispatch problem by a fuzzy adaptive genetic algorithm*, Iranian Journal of Fuzzy Systems, **8(3)** (2011), 1-21.
[33] E. C. Ozelkan and L. Duckstein, *Optimal fuzzy counterparts of scheduling rules*, European Journal of Operational Research, **113(3)** (1999), 593-609.
[34] D. Peidro and P. Vasant, *Fuzzy Multi-Objective Transportation Planning with Modified SCurve Membership Function*, In Proceedings of Global Conference on Power Control and Optimization, **35** (2009), 101-110.
[35] S. Pugazhendhi, S. Thiagarajan, C. Rejendran and N. Anantharaman, *Generating non-permutation schedules in flow line based manufacturing systems with sequence-dependent setup times of jobs: a heuristic approach*, Applied Management and Technology, **23(1)** (2004), 64-78.
[36] Z. Qin and X. Ji, *Logistics network design for product recovery in fuzzy environment*, European Journal of Operational Research, **202(2)** (2010), 479-490.
[37] S. Ramezanzadeh and A. Heydari, *Optimal control with fuzzy chance constraints*, Iranian Journal of Fuzzy Systems, In press.
[38] R. Ruiz and C. Maroto, *A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility*, European Journal of Operational Research, **169(3)** (2006), 781-800.
[39] R. Ruiz, F. Svirikaya-Serifoglu and T. Urlings, *Technical report, polytechnic university of valencia*, Department of Applied Statistics and Operating Research, Spain, 2006.
[40] M. R. Safi, H. R. Maleki and E. Zaeimazad, *A note on zimmermann method for solving fuzzy linear programming problems*, Iranian Journal of Fuzzy Systems, **4(2)** (2007), 31-45.
[41] M. Sakawa and R. Kubota, *Fuzzy programming for multi objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms*, European Journal of Operational Research, **120(2)** (2000), 393-407.
[42] J. Salerno, *Using the particle swarm optimization technique to train a recurrent neural model*, IEEE Transactions, International Conference on Tools with Artificial Intelligence, (1997), 45-49.
[43] T. Sawik, *Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers*, Mathematical and Computer Modelling, **31** (2000), 39-52.
[44] Y. Shi and R. Eberhart, *Particle swarm optimization: development, applications and resources*, IEEE Transaction, **3(1)** (2001), 81-86.

[45] E. Shivanian, E. Khorram and A. Ghodousian, *Optimization of linear objective function subject to fuzzy relation inequalities constraints with max-average composition*, Iranian Journal of Fuzzy Systems, **4(2)** (2007), 15-29.
[46] H. Tanaka, H. Ichihashi and K. Asai, *A formulation of fuzzy linear programming problem bases on comparison of fuzzy numbers*, Control and Cybernetics, **13** (1984), 185-194.
[47] S. A. Torabi and E. Hassini, *An interactive possibilistic programming approach for multiple objective supply chain master planning*, Fuzzy Sets and Systems, **159** (2008), 193-214.
[48] S. A. Torabi and E. Hassini, *Multi-site production planning integrating procurement and distribution plans in multi-echelon supply chains: an interactive fuzzy goal programming approach*, International Journal of Production Research, **47(19)** (2009), 5475-5499.
[49] S. A. Torabi, M. Ebadian and R. Tanha, *Fuzzy hierarchical production planning (with a case study)*, Fuzzy Sets and Systems, **161** (2010), 1511-1529.
[50] S. Vob and A. Witt, *Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: a real-world application*, International Journal of Production Economics, **105(2)** (2007), 445-458.
[51] L. Zadeh,*Fuzzy sets*, Information and Control, **8** (1965), 338-353.
[52] H. Zimmermann, *Fuzzy programming and linear programming with several objective functions*, Fuzzy Sets and Systems, **1(1)** (1978), 45-56.

Ali Ghodratnama, Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
*E-mail address*: ghodratn@ut.ac.ir

Seyed Ali Torabi*, Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
*E-mail address*: satorabi@ut.ac.ir

Raza Tavakkoli-Moghaddam, Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
*E-mail address*: tavakoli@ut.ac.ir

*Corresponding author