Mazandaran University of Science and Technology

13th International Conference on Industrial Engineering (IIEC 2017)

Available online at www.iiec2017.com

Iran Institute of Industrial Engineering

Fuzzy bi-objective mathematical model for a hybrid flow shop scheduling problem

Fatemeh Pourdehghan Golneshini[1], Hamed Fazlollahtabar[2]

[1] Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran
E-mail: fatemeh.poordehghan@ustmb.ac.ir

[2]Faculty of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
E-mail: hfazl@iust.ac.ir

Abstract

Nowadays, flow shop is the most common production environment in industry. However, the need to increase or balance the capacity in different production stages has led to having more than one machine at some stages. The flow shops are usually categorized as hybrid flow shop, multi-processor flow shop, flexible flow shop or flow shop with parallel machines. This article deals with the hybrid flow shop scheduling problem with unrelated and eligible machines along with fuzzy processing time and fuzzy due date. The objective is to minimize a linear combination of total completion time and maximum lateness. A mixed integer mathematical model is presented for the problem. A numerical example is worked out and various randomly generated test problems are used to investigate the efficiency of the proposed mathematical model.

Keywords:
Hybrid flow shop, mathematical model, fuzzy numbers, bi-objective programming.

Introduction

Scheduling is a form of decision- making process that plays a pivotal role in manufacturing and service systems. It is a process of allotting available and limited resources, such as machines, material handling equipment, operators, and tools, over a specific time to perform a collection of required tasks in order to achieve certain objectives. The first studies on scheduling appeared in the 1950s with the paper of Johnson [1]. Since then, scheduling has attracted a vast body of interest.

Different types of scheduling problems were addressed in the literature with different performance measures. Single machine scheduling, flow shop scheduling, parallel machine scheduling, job shop scheduling, HFS scheduling and project scheduling are some of the important types of scheduling environments [2].

The flow shop scheduling problem has attracted many researchers since Johnson's seminal paper. In the classical flow shop problem, a set of jobs flow through multiple stages in the same machine order, where each stage consists of only one machine. Nowadays, in some companies, the need to increase the capacity of the shop floor or to balance the capacity between different stages has led to the duplication of some machines at some stages. In other companies, the increasing demand of customized products, i.e. special packaging, sizes or features has triggered the need to purchase additional machines at some stages in order to dedicate some of them to these special products [3]. This extended layout is usually addressed as hybrid flow shop, flow shop with multiple machines (processors), flexible flow shop, multiprocessor flow shop, or flow shop with parallel machines. In this study, we will refer to this shop floor configuration as hybrid flow shop (HFS).

The HFS scheduling problem, which is one of the most applied and well-known production scheduling problem, is an extension of two particular types of scheduling problems: the parallel machine scheduling (PMS) problem and the flow shop scheduling (FSS) problem [3]. This type of environment is relatively common and has a variety of applications in real-world industries such as chemical, steel, automobile, food, textile, paper, pharmaceutical, semiconductor and electronics manufacturing, printed circuit board etc.

In hybrid flow shop scheduling problems, it is assumed that a set of n jobs must be sequentially processed on a set of k production stages or machine centers, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must include multiple machines, which introduces the additional flexibility to the production process. Machines at each stage can be identical, uniform or unrelated and each machine within each stage can process one job at a time. Moreover, the job must be processed by exactly one machine at every stage. There is no specific assumption on the similarity of the machines at each stage. Each job is processed first at stage one, then at stage two, and finally at stage k. Each

stage has m_i $(i = 1,2,...k)$ number of parallel processors. It is convenient to view a job as a sequence of k tasks – one task for each stage – where the processing of any task can commence only after the completion of the preceding task. The purpose of HFS scheduling is to determine the jobs and their processing order on each machine. The layout of a k stage hybrid flowshop scheduling is shown in Figure 1 where M_{mk} represents the machine in the m th parallel and k th stage [2] This definition is very general and is the basis of the papers reviewed.

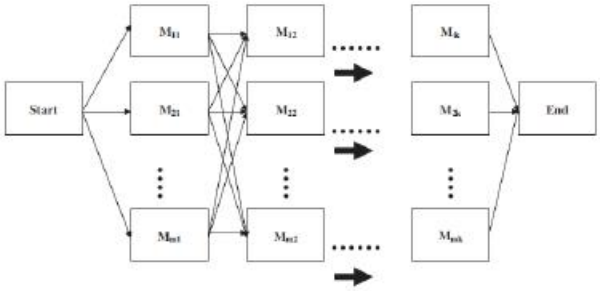

Figure 1- Layout of the hybrid flow shop environment [2].

Research efforts on production scheduling generally consider a static environment with a fixed number of jobs, deterministic processing time, and no unexpected events that would influence job processing when the schedule is executed. Real manufacturing is, however, dynamic and subject to a wide range of stochastic uncertainties, such as machine breakdown, stochastic processing time, and rush order. Therefore, production scheduling under uncertainty has indeed attracted much attention in recent years. [4]

In addition, there are various factors involved in real-world scheduling problems that are often imprecise or uncertain in nature. This is especially true when human-made factors are considered into the problems. Therefore, parameters are often encountered with uncertainties. Accordingly, production scheduling problems can be divided into two general categories: deterministic scheduling and uncertain scheduling problems [5]. There are essentially two approaches to deal with uncertainties, including the stochastic-probabilistic theory and possibility theory or fuzzy set theory [5]. In this study, fuzzy set theory is used to deal with the uncertainties in production scheduling

The initial paper in HFS scheduling problem was proposed by Arthanari and Ramamorthi [6] in 1971 and studying over this field has attracted many interests over time. Gupta [7] in 1988 considered a two-stage HFS with identical parallel machines at each stage and objective of $C_{\max}$. He proved that the problem is NP-complete and proposed a branch and bound algorithm to find an approximate solution in a specific condition in which the second stage contains one machine. Consequently, it was claimed that HFS scheduling problems with two stages are NP-hard, even if there is only one machine and two machines at stage 1 and stage 2, respectively.

Brah and Hunsucker [8] presented a branch and bound algorithm to minimize maximum completion time in flow shop scheduling with multiple processors. Riane et al. [9] considered a problem of scheduling n jobs on a three stages hybrid flowshop with the objective of minimizing the makespan and proposed two heuristic procedures to cope with it. Ruiz et al. [10] proposed a mixed integer programming model and some heuristics for the problem of scheduling n jobs at m stages where at each stage there is a specified number of unrelated machines. The optimization criterion considered was the makespan minimization. The MIP model and the heuristic proposed were tested against a comprehensive benchmark and the results were evaluated by advanced statistical tools that make use of decision trees and experimental designs.

Amin-Naseri and Beheshti-Nia [11] studied the problem of parallel batch scheduling in a hybrid flow shop environment with minimizing makespan. Since the problem was NP-hard, three heuristic algorithms were introduced to give near optimal solutions and to further enhance the solution quality, a three dimensional genetic algorithm (3DGA) was also developed. Several test problems had been solved using 3DGA and the results indicated its superiority to the other heuristics. Mahdavi et al. [12] considered a multi-stage flow shop scheduling problem with equal number of unrelated parallel machines and the objective of makespan minimization for a given set of jobs in the system. They proposed a mixed integer linear programming formulation for the considered production model. To solve the problem, they used a hybrid heuristic method and the results obtained by proposed heuristic were compared with optimal solutions reported by the Lingo 8.0 package applying the branch and bound approach. The results showed that the proposed hybrid method was more efficient when the problem sizes had been increased.

Wang et al. [13] studied a hybrid flow shop scheduling problem with multiprocessor tasks in order to minimize the makespan. They developed a simulated annealing (SA) algorithm and a new neighborhood mechanism was combined with the proposed SA for generating neighbor solutions. Ziaeifar et al. [14] presented a new mathematical model for a hybrid flow shop scheduling problem with a processor assignment that minimized makespan and cost of assigning a number of processors to stages. In this problem, it was assumed that there were a number of parallel identical processors which were assigned to all of the stages with an unlimited intermediate storage between any two successive stages. The processors could be transferred between stages and cooperate with processors at each stage in order to perform the operations. To solve the proposed problem, genetic algorithm was used.

Attar et al. [15] presented a mathematical model for a hybrid flow shop scheduling problem with unrelated machines, limited waiting time between every two successive processing operations and ready time of jobs. The aim of this paper was to model and solve the addressed problem by applying an efficient metaheuristic algorithm, entitled biogeography-based optimization (BBO). To assess the proposed BBO, two experiments were conducted and their results in terms of solutions quality, as well as

computation efficiency compared against two popular algorithms, namely imperialist competitive algorithm and population- based simulated annealing.
Mousavi et al. [16] aimed to determine a schedule for hybrid flow shop problem that minimized a convex combination of the makespan and total tardiness. A meta-heuristic procedure was proposed based on the simulated annealing/local search (SA/LS). The performance of the proposed algorithm was compared with a genetic algorithm which had had been presented in the literature for hybrid flow shop with the objective of minimizing a convex combination of the makespan and the number of tardy jobs.
Wang and Liu [17] considered a bi-objective optimization problem in a two-stage HFS scheduling problem with preventive maintenance and sequence-dependent setup time. The objectives were to minimize the unavailability of the first stage machine and to minimize the makespan simultaneously. Due to complexity of the problem, a multi-objective tabu search (MOTS) method was used and the performance of the method was compared with that of a multi-objective genetic algorithm.

Problem description and mathematical model

The HFS scheduling problem considered in this work consists of S stages in series. Each stage k $(k=1,2,...,S)$ consists of i $(i=1,2,...,M)$ machines in parallel. It is assumed that machines at each stage are unrelated, meaning that they have diverse speed for each job type. V_{kit} is the speed of machine i at stage k to process job type t. A set of N independent jobs with due date $\tilde{d}_1,\tilde{d}_2,...,\tilde{d}_n$ and different job types has to pass through S stages consecutively and be processed on anyone of the machines available at each stage. Each job j of type t, $1\le t\le l$, has a $\tilde{p}_{kt}$ fixed processing time at stage k. Processing time is the time required to process a job at a stage. The objective is to minimize a linear combination of total completion time and maximum lateness, which are briefly described as follows:

$\sum_{j=1}^{N}\tilde{C}_j$: Total job completion time at the last stage

$\tilde{L}_{\max}$: Maximum lateness which is computed as follows:

$$\tilde{L}_{\max}=\max(\tilde{L}_j)=\max(\tilde{C}_j-\tilde{d}_j),\quad j=1,...,N$$

Where $\tilde{d}_j$ is due date of job j [18].

Completion time is the moment at which the job is completed at a stage. Completion time of job j at stage k is equal to sum of its starting time and processing time at stage k. A job processing at first stage can be started if the job is ready. Job j ready time is the moment at which the job j is available for processing. Each job should be assigned to exactly one machine at each stage. Job processing can be started at a stage only when it has been completed at the previous stage.
In order to solve the HFS scheduling problem, the following assumptions are considered in this paper:

- The number of stages and machines the at each stage are known in advance
- All jobs and machines are available at the beginning of scheduling.
- Preemption is not allowed for job processing.
- There is no breakdown for machines and machines are always available.
- Job setup and transportation time are sequence independent and are included in the processing time.
- Parallel machines at each stage are unrelated and eligible.
- Infinite buffers exist between two consecutive stages.
- Each job can be processed on one machine at a time.
- A machine can not process more than one job at the same time.
- Job processing time and due date are uncertain and are considered as fuzzy numbers.
- Parallel machines at each stage have different processing speed for each job type.
- Jobs are independent and there is no precedent constraint between them.

Prior to present the proposed model, we describe indices, parameters and decision variables.

Indices

h, j : Job index, $h, j\in\{1,2,...,N\}$
q : Sequence index, $q\in\{1,2,...,N\}$
k : Stage index, $k\in\{1,2,...,S\}$
i : Machine index, $i\in\{1,2,...,M\}$
t : Job type index, $t\in\{1,2,...,L\}$

Parameters

N : Total number of jobs.
S : Number of stages.
M : Maximum number of machines at each stage.
L : Total number of job type.
e_{kit} : If machine i at stage k can process job type t, 1 otherwise 0.
w_{tj} : If job j is of type t, 1 otherwise 0.

z_{ki}: If machine i exists at stage k, 1 otherwise 0.

V_{kit}: Machine i Speed at stage k to process job type t.

$\tilde{P}_{kt}$: Fuzzy processing time of job type t at stage k.

$\tilde{d}_j$: Fuzzy due date of job j.

U: A positive large number.

Decision variables

In this part, decision variables of the model are designated. Two kinds of decision variables are used in this study: binary variables and continuous variables. In the following both of them are clarified.

In order to determine the eligible machine by which each job at each stage is processed, the binary variable X_{kiqj} is used:

X_{kiqj}: If job j at stage k is assigned to sequence q of machine i, 1 otherwise 0.

Furthermore, continuous variables are required to determine completion time of each job. These variables are described as follows:

$\tilde{C}_{kiqj}$: Fuzzy completion time of job j in sequence q of machine i at stage k,

$\tilde{L}_j$: Fuzzy lateness of job j.

Mathematical model

According to the aforementioned notations, the proposed mathematical model for this problem is as follows:

$$F_1 = \sum_{i=1}^{m}\sum_{q=1}^{N}\sum_{j=1}^{N}\tilde{C}_{siqj} \quad (1)$$

$$F_2 = \max(\tilde{C}_{siqj} - \tilde{d}_j,\ i=1,2,...,m,\ q=1,2,...,N,\ j=1,2,...,N) \quad (2)$$

$$F = \min(\alpha F_1 + (1-\alpha)F_2) \quad (3)$$

$s.t.$

$$X_{kiqj} \le \sum_{t=1}^{L} w_{tj} e_{kit},\quad k=1,2,...,S,\ i=1,2,...,m,\ j=1,2,...,N,\ q=1,2,...,N \quad (4)$$

$$\sum_{i=1}^{m}\sum_{q=1}^{N} X_{kiqj} = 1,\ k=1,2,...,S,\ j=1,2,...,N \quad (5)$$

$$\sum_{j=1}^{N} X_{kiqj} \le 1,\ k=1,2,...,S,\ i=1,2,...,m,\ q=1,2,...,N \quad (6)$$

$$\tilde{C}_{1ij} \ge \sum_{t=1}^{L} X_{1ij}\frac{\tilde{P}_{1t}}{\tilde{V}_{1it}} w_{tj} e_{1it},\ i=1,2,...,m,\ j=1,2,...,N \quad (7)$$

$$\tilde{C}_{liqj} + U(1-X_{liqj}) \ge \sum_{h\ne j}\tilde{C}_{li(q-1)h} + \sum_{t=1}^{L} X_{liqj}\frac{\tilde{P}_{1t}}{\tilde{V}_{lit}} w_{tj} e_{lit}$$
$$i=1,...,m,\ q=1,...,N,\ j=1,...,N,\ q>1$$
$$\tilde{C}_{k1j} + U(1-X_{k1j}) \ge \sum_{i'=1}^{m}\sum_{q'=1}^{N}\tilde{C}_{(k-1)i'q'j} + \sum_{t=1}^{L} X_{k1j}\frac{\tilde{P}_{kt}}{\tilde{V}_{kit}} w_{tj} e_{kit} \quad (9)$$
$$k=1,...,S,\ i=1,...,m,\ j=1,...,N,\ k>1$$

$$\tilde{C}_{kiqj} + U(1-X_{kiqj}) \ge \max(\sum_{i'=1}^{m}\sum_{q'=1}^{N}\tilde{C}_{(k-1)i'q'j}, \sum_{h\ne j}\tilde{C}_{ki(q-1)h}) + \sum_{t=1}^{L} X_{kiqj}\frac{\tilde{P}_{kt}}{\tilde{V}_{kit}} w_{tj} e_{kit},\ \forall k>1,\ q>1,\ i,j \quad (10)$$

$$\sum_{j\ne h} X_{kiqj} \ge X_{ki(q+1)h},\ \forall k,i,q,h \quad (11)$$

$$X_{kiqj} \in \{0,1\},\ i=1,...,m,\ j=1,...,N,\ k=1,...,S,\ q=1,...,N \quad (12)$$

In the above formulations, constraints (1) and (2) define total job completion time and maximum lateness, respectively. Constraint (3) gives linear combination of total job completion time and maximum lateness. Constraint (4) ensures that X_{kiqj} is 1 only if job j is of type t and machine i at stage k is eligible to process it. Constraint (5) forces each job is processed only on one machine at each stage. Constraint (6) guarantees that only one job is assigned to each sequence of a machine at a moment. Constraints (7) and (8) calculate job completion time in first and q th ($q>1$) sequence at first stage, respectively. Constraints (9) and (10) calculate job completion time in first and q th ($q>1$) sequence at stage $k(k>1)$, respectively. Constraint (11) ensures that if no job is assigned to a machine sequence, subsequent sequences have to be remained empty. Constraint (12) demonstrates binary variables of the proposed problem.

Solution methodology

Fuzzy numbers and calculations

In this study, processing time and due dates have been considered as triangular fuzzy numbers. According to these parameters, the fuzzy completion time and lateness are proposed based on the concepts of subtraction and maximum of two fuzzy numbers, which are defined by using the well-known "Extension Principle" in fuzzy sets theory. The objective function is taken as the weighted sum of fuzzy lateness and sum of fuzzy completion time through the concept of addition among fuzzy numbers. Consequently, the objective function turns into a fuzzy-valued function. Hence, in this part we introduce fuzzy numbers and then elucidate how fuzzy completion time and fuzzy lateness are calculated. Finally, the way in

which the objective function is computed and defuzzified is described.

The fuzzy subset $\tilde{a}$ of R is defined by a function $\mu_{\tilde{a}} : R \to [0,1]$, which is called a membership function. The α-level set of $\tilde{a}$, denoted by $\tilde{a}_\alpha$, is defined by $\tilde{a}_\alpha = \{x \in R : \mu_{\tilde{a}}(x) \geq \alpha\}$ for all $\alpha \in (0,1]$. The 0-level set $\tilde{a}_0$ is defined as the closure of the set $\tilde{a}_0 = \{x \in R : \mu_{\tilde{a}}(x) \geq 0\}$ [19].

Definition: The fuzzy subset $\tilde{a}$ of R is said to be a fuzzy number if the following conditions are satisfied:

(i) $\tilde{a}$ is normal, i.e., there exists an $x \in R$ such that $\xi_{\tilde{a}}(x) = 1$;

(ii) $\xi_{\tilde{a}}$ is quasi-concave, i.e., $\xi_{\tilde{a}}(tx + (1-t)y) \geq \min\{\xi_{\tilde{a}}(x), \xi_{\tilde{a}}(y)\}$ for $t \in [0,1]$;

(iii) $\xi_{\tilde{a}}$ is upper semi continuous, i.e., $\{x \in R : \xi_{\tilde{a}}(x) \geq \alpha\}$ is a closed subset of R for each $\alpha \in (0,1]$;

(iv) The 0-level set $\tilde{a}_0$ is a closed and bounded subset of R.

Keep in mind that if $\tilde{a}$ is a fuzzy number, Then the α-level of $\tilde{a}$ is denoted by $\tilde{a}_\alpha = [\tilde{a}_\alpha^L, \tilde{a}_\alpha^U]$ [19].

Proposition 3.1. Let $\tilde{a}$ and $\tilde{b}$ be two fuzzy numbers. Then $\tilde{a} \oplus \tilde{b}$, $\tilde{a} \Theta \tilde{b}$ and $\max\{\tilde{a}, \tilde{b}\}$ are also fuzzy numbers. Furthermore, we have

$$(\tilde{a} \oplus \tilde{b})_\alpha = [\tilde{a}_\alpha^l + \tilde{b}_\alpha^l, \tilde{a}_\alpha^u + \tilde{b}_\alpha^u]$$
$$(\tilde{a} \Theta \tilde{b})_\alpha = [\tilde{a}_\alpha^l - \tilde{b}_\alpha^u, \tilde{a}_\alpha^u - \tilde{b}_\alpha^l]$$
$$(\max\{\tilde{a}, \tilde{b}\})_\alpha = [\max\{\tilde{a}_\alpha^l, \tilde{b}_\alpha^l\}, \max\{\tilde{a}_\alpha^u, \tilde{b}_\alpha^u\}].$$

Triangular fuzzy number (TFN) is one of the most useful fuzzy numbers which is denoted by $\tilde{a} = (a^l, a, a^u), a^l \leq a \leq a^u$ The TFN $\tilde{a}$ can be expressed as "around a" or "being approximately equal to a". We also say that a is the core value of $\tilde{a}$ meaning $\mu_{\tilde{a}}(a) = 1$, and a^l and a^u are left and right spread values of $\tilde{a}$, respectively. A TFN $\tilde{a}$ can be denoted by its membership function:

$$\mu_{\tilde{a}}(x) = \begin{cases} (x - a^l)/(a - a^l) & \text{if } a^l \leq x \leq a, \\ (a^u - x)/(a^u - a) & \text{if } a < x \leq a^u, \\ 0 & \text{otherwise,} \end{cases}$$

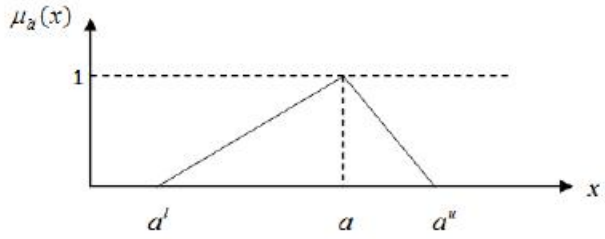

Figure 2- Triangular fuzzy number [19]

The α-level set of $\tilde{a}$ is then

$$\tilde{a}_\alpha = [(1-\alpha)a^l + \alpha a, (1-\alpha)a^u + \alpha a].$$

Suppose that $\tilde{a} = (a^l, a, a^u)$ and $\tilde{b} = (b^l, b, b^u)$ are two triangular fuzzy numbers. Some arithmetic operations on triangular fuzzy numbers can be described as follows:

Addition

$$\tilde{a} \oplus \tilde{b} = (a^l, a, a^u) \oplus (b^l, b, b^u) = (a^l + b^l, a + b, a^u + b^u)$$

Subtraction

$$\tilde{a} \Theta \tilde{b} = (a^l, a, a^u) \Theta (b^l, b, b^u) = (a^l - b^u, a - b, a^u - b^l)$$

The multiplication of a TFN and a crisp value $k, k \in R_0^+$ is defined as:

$$k.\tilde{a} = k.(a^l, a, a^u) = (k.a^l, k.a, k.a^u)$$

In this paper, processing time and due date have been considered as $\tilde{p}_{kit} = (p_{kit}^L, p_{kit}, p_{kit}^U)$ and $\tilde{d}_j = (d_j^L, d_j, d_j^U)$, respectively.

Fuzzy completion time calculation

As mentioned before, completion time has a direct relation with processing time. So, when processing time is a fuzzy number, completion time is also turned into a fuzzy number. Here, according to fuzzy processing time and by using fuzzy operations, the completion time is calculated.

If processing time is a triangular fuzzy number, i.e.

$\widetilde{p}_{kit} = (p^L_{kit}, p_{kit}, p^U_{kit})$, then completion time would be a triangular fuzzy number as $\widetilde{C}_{kiqj} = (C^L_{kiqj}, C_{kiqj}, C^U_{kiqj})$. It should be noted that in completion time calculation, it is requisite to calculate $\frac{\widetilde{P}_{kt}}{V_{kit}}$ and since $\widetilde{P}_{kt}$ is a fuzzy number, V_{kit} which is a crisp number, is transformed to a fuzzy number:

$$\widetilde{V}_{kit} = (V_{kit}, V_{kit}, V_{kit}) = (V^L_{kit}, V_{kit}, V^U_{kit}).$$

According to the proposed model, fuzzy completion time are calculated as follows:

C^L_{kiqj} ***Calculation:***

$$C^L_{1i1j} \geq \sum_{t=1}^{L} X_{1i1j} \frac{P^L_{1t}}{V^L_{1it}} w_{tj} e_{1it}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N$$

$$C^L_{1iqj} + U(1 - X_{1iqj}) \geq \sum_{h \neq j} C^L_{1i(q-1)h} + \sum_{t=1}^{L} X_{1iqj} \frac{P^L_{1t}}{V^L_{1it}} w_{tj} e_{1it},$$

$$i = 1,\ldots,m, \quad q = 1,\ldots N, \quad j = 1,\ldots,N, \quad q > 1$$

$$C^L_{ki1j} + U(1 - X_{ki1j}) \geq \sum_{i'=1}^{m} \sum_{q'=1}^{N} C^L_{(k-1)i'q'j} + \sum_{t=1}^{L} X_{ki1j} \frac{P^L_{kt}}{V^L_{kit}} w_{tj} e_{kit}$$

$$k = 1,\ldots,S, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N, \quad k > 1$$

$$C^L_{kiqj} + U(1 - X_{kiqj}) \geq \max\left(\sum_{i'=1}^{m} \sum_{q'=1}^{N} C^L_{(k-1)i'q'j}, \sum_{h \neq j} C^L_{ki(q-1)h}\right) +$$

$$\sum_{t=1}^{L} X_{kiqj} \frac{P^L_{kt}}{V^L_{kit}} w_{tj} e_{kit}, \quad \forall k > 1, \quad q > 1, \quad i, j.$$

C_{kiqj} ***Calculation:***

$$C_{1i1j} \geq \sum_{t=1}^{L} X_{1i1j} \frac{P_{1t}}{V_{1it}} w_{tj} e_{1it}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N$$

$$C_{1iqj} + U(1 - X_{1iqj}) \geq \sum_{h \neq j} C_{1i(q-1)h} + \sum_{t=1}^{L} X_{1iqj} \frac{P_{1t}}{V_{1it}} w_{tj} e_{1it},$$

$$i = 1,\ldots,m, \quad q = 1,\ldots N, \quad j = 1,\ldots,N, \quad q > 1$$

$$C_{ki1j} + U(1 - X_{ki1j}) \geq \sum_{i'=1}^{m} \sum_{q'=1}^{N} C_{(k-1)i'q'j} + \sum_{t=1}^{L} X_{ki1j} \frac{P_{kt}}{V_{kit}} w_{tj} e_{kit},$$

$$k = 1,\ldots,S, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N, \quad k > 1$$

$$C_{kiqj} + U(1 - X_{kiqj}) \geq \max\left(\sum_{i'=1}^{m} \sum_{q'=1}^{N} C_{(k-1)i'q'j}, \sum_{h \neq j} C_{ki(q-1)h}\right) +$$

$$\sum_{t=1}^{L} X_{kiqj} \frac{P_{kt}}{V_{kit}} w_{tj} e_{kit}, \quad \forall k > 1, \quad q > 1, \quad i, j$$

C^U_{kiqj} ***Calculation:***

$$C^U_{1i1j} \geq \sum_{t=1}^{L} X_{1i1j} \frac{P^U_{1t}}{V^U_{1it}} w_{tj} e_{1it}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N$$

$$C^U_{1iqj} + U(1 - X_{1iqj}) \geq \sum_{h \neq j} C^U_{1i(q-1)h} + \sum_{t=1}^{L} X_{1iqj} \frac{P^U_{1t}}{V^U_{1it}} w_{tj} e_{1it},$$

$$i = 1,\ldots,m, \quad q = 1,\ldots N, \quad j = 1,\ldots,N, \quad q > 1$$

$$C^U_{ki1j} + U(1 - X_{ki1j}) \geq \sum_{i'=1}^{m} \sum_{q'=1}^{N} C^U_{(k-1)i'q'j} + \sum_{t=1}^{L} X_{ki1j} \frac{P^U_{kt}}{V^U_{kit}} w_{tj} e_{kit},$$

$$k = 1,\ldots,S, \quad i = 1,\ldots,m, \quad j = 1,\ldots,N, \quad k > 1$$

$$C^U_{kiqj} + U(1 - X_{kiqj}) \geq \max\left(\sum_{i'=1}^{m} \sum_{q'=1}^{N} C^U_{(k-1)i'q'j}, \sum_{h \neq j} C^U_{ki(q-1)h}\right) +$$

$$\sum_{t=1}^{L} X_{kiqj} \frac{P^U_{kt}}{V^U_{kit}} w_{tj} e_{kit}, \quad \forall k > 1, \quad q > 1, \quad i, j$$

Thus, according to these relations and operations, final triangular fuzzy completion time is calculated as $\widetilde{C}_{kiqj} = (C^L_{kiqj}, C_{kiqj}, C^U_{kiqj})$ for each job.

The following method, which was used by Behnamian and Zandieh [20], is used to produce due date of jobs.

Firstly, Sum of job processing time at all S stages is calculated (it is assumed that j th job is of type t):

$$\widetilde{P}_j = \sum_{k=1}^{S} \widetilde{P}_{kt}, \quad \forall j \in N, \quad t \in L.$$

Then, due date of each job is generated as follows:

$$\widetilde{d}_j = (1 + random \times 3) \times (S / \min_{k \in S}(m_k)) \times \widetilde{p}_j \ , \quad \forall j \in n,$$

In which $random$ is a random number with a uniform distribution in $(0,1)$.

Fuzzy lateness calculation

Apart from fuzzy completion time, fuzzy lateness of jobs have to be calculated also. When completion time is greater than due date, there is lateness. In classic scheduling problems, a lateness is calculated as following:

$$L_j = C_j - d_j,$$

In which, d_j, C_j and L_j are due date, completion time and lateness of job j, respectively.

In fuzzy problems, completion time $\tilde{C}_j$ and due date $\tilde{d}_j$ are considered as fuzzy numbers.
Therefore, the lateness in fuzzy space is described as:

$$\tilde{L}_j = \tilde{C}_j \Theta \tilde{d}_j.$$

According to aforementioned relations, $\tilde{L}_j$ is a fuzzy number. Consequently, by using α -level it can be considered as following:

$$\tilde{L}_{j\alpha}^{\,L} = \tilde{C}_{j\alpha}^{\,L} - \tilde{d}_{j\alpha}^{\,U},$$
$$\tilde{L}_{j\alpha}^{\,U} = \tilde{C}_{j\alpha}^{\,U} - \tilde{d}_{j\alpha}^{\,L}.$$

Defuzzification method

It is crystal clear that objective functions are fuzzy numbers. So, to find optimum amount of final objective function, these fuzzy numbers should be changed into crisp numbers. This operation is called defuzzification and is conducted in diverse manners.
Defuzzification is a reverse process of fuzzification and it converts a fuzzy number into a crisp value. The "Center of Area"(COA) defuzzification method is one of the most prevalent and physically appealing defuzzification method. It returns the center of area under the curve. Let A be a fuzzy number and μ_A its membership function. Defuzzification of A by the COA method is denoted by $\delta(A)$ and returns the crisp value of x^*:

$$x^* = \delta(A) = \frac{\int \mu_A(x).x dx}{\int \mu_A(x) dx}.$$

For a TFN $A=(p,q,r)$, $x^* = \delta(A) = (p+q+r)/3$ and it represents the centroid of the triangle $((p,0);(q,1);(r,0))$ it can be observed in in Figure 3.

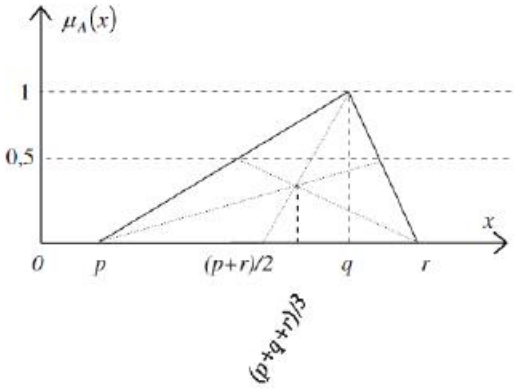

Figure 3- Center of area defuzzification method for a triangular fuzzy number [۲۱]

Ranking method

Ranking two numbers is quite important in a scheduling algorithm. Ranking is quite easily performed since the operands are all crisp numbers. However, in this paper, since fuzzy processing time and due date are considered, fuzzy ranking methods have to be used. In literature, several fuzzy comparison methods have been proposed, such as the Hamming distance method, the probability distribution method, pseudo-order fuzzy preference model, new fuzzy-weighted average, signed distance method, and so on. In this study, "mean-variance" method is used. To rank two TFN $A=(a_1,a_2,a_3)$ and $B=(b_1,b_2,b_3)$, mean and variance should be calculated for both of them:

$$\bar{x}(A) = \frac{a_1+a_2+a_3}{3},$$
$$\delta(A) = \frac{(a_1)^2+(a_2)^2+(a_3)^2-(a_1a_2+a_1a_3+a_2a_3)}{18}$$
$$\bar{x}(B) = \frac{b_1+b_2+b_3}{3},$$
$$\delta(B) = \frac{(b_1)^2+(b_2)^2+(b_3)^2-(b_1b_2+b_1b_3+b_2b_3)}{18}.$$

TFNs can be ranked by mean-variance method as follows:

$A > B$ if $\bar{x}(A) > \bar{x}(B)$,
$A > B$ If $\bar{x}(A) = \bar{x}(B)$ and $\delta(A) < \delta(B)$.

Numerical example

In order to evaluate and validate the effectiveness of proposed mathematical model, solving procedure is explained in details for a small size test problem.
Assume that there are 3 stages, 5 jobs and 2 job types. The number of parallel machines at each stage is 2, 2 and 1,

respectively and a similar speed of 1 has been considered for all machines to process all jobs. Table 1 shows fuzzy processing time and fuzzy due date and Table 2 denotes eligible machines at each stage to process jobs. It is noteworthy to be mentioned that, in order to compare the solution obtained from the mathematical model and a random solution, due dates are generated randomly from uniform distribution [1,300] and then considered as fuzzy numbers in the model. The presented mathematical model for this small size test problem is performed in GAMS optimization software and the results were compared with results obtained by a random sequence. The results obtained from both methods have been represented in Table 3 to 6. Tables 3 and 5 demonstrate objective functions and the amount of x_{kiqj} variables are presented in Table ٤ and ٦. For instance, $x_{2123}=1$ means that job 3 in stage2 is processed in sequence 2 of the first machine.

Table 1- Test problem parameters

job	Job type	$\tilde{P}_{1j}$	$\tilde{P}_{2j}$	$\tilde{P}_{3j}$	$\tilde{d}_j$
١	١	(١٥,١٦,١٧)	(٤٦,٤٧,٤٨)	(٩,١٠,١١)	(٤٨,٤٨,٤٨)
٢	٢	(٢٨,٢٩,٣٠)	(٣٣,٣٤,٣٥)	(٢٠,٢١,٢٢)	(١٣٣,١٣٣,١٣٣)
٣	٢	(٢٨,٢٩,٣٠)	(٣٣,٣٤,٣٥)	(٢٠,٢١,٢٢)	(١٤٩,١٤٩,١٤٩)
٤	١	(١٥,١٦,١٧)	(٤٦,٤٧,٤٨)	(٩,١٠,١١)	(١٢٤,١٢٤,١٢٤)
٥	١	(١٥,١٦,١٧)	(٤٦,٤٧,٤٨)	(٩,١٠,١١)	(٢٧٤,٢٧٤,٢٧٤)

Table 2- Eligible machines at each stage to process each job

Stage		١	٢	٣
Job	١	{١,٢}	{١}	{١}
	٢	{٢}	{١,٢}	{١}
	٣	{٢}	{١,٢}	{١}
	٤	{١,٢}	{١}	{١}
	٥	{١, ٢}	{١}	{١}

Table 3- Objective functions resulted from random solution

Job	$\tilde{C}_{kiqj}$	$\tilde{L}$	$\tilde{L}_{max}$	$\sum \tilde{C}_{kiqj}$	$0.5\times\sum \tilde{C}_{kiqj}+0.5\times \tilde{L}_{max}$
1	(70,73,76)	(22,25,28)			
2	(90,94,98)	(-43,-39,-35)			
3	(114,118,122)	(-35,-31,-27)	(25,30,35)	(618,640,662)	(321.5,335,348.5)
4	(149,154,159)	(25,30,35)			
5	(195,201,207)	(-79,-73,-67)			

Table 4- x_{kiqj} variables resulted from random solution

x_{kiqj}		Stage 1	Stage 2	Stage 3
Job	1	$x_{1111}=1$	$x_{2111}=1$	$x_{3111}=1$
	2	$x_{1212}=1$	$x_{2212}=1$	$x_{3122}=1$
	3	$x_{1223}=1$	$x_{2123}=1$	$x_{3133}=1$
	4	$x_{1124}=1$	$x_{2134}=1$	$x_{3144}=1$
	5	$x_{1135}=1$	$x_{2145}=1$	$x_{3155}=1$

Table 5- Objective functions resulted from GAMS

job	C_{kiqj}	L	L_{max}	$\sum C_{kiqj}$	$0.5\times\sum C_{kiqj}+0.5\times L_{max}$
1	73	25			
2	94	-39			
3	118	-31	25	580	302.5
4	128	4			
5	167	-107			

Table 6- x_{kiqj} variables resulted from GAMS

x_{kiqj}		Stage 1	Stage 2	Stage 3
Job	1	$x_{1111}=1$	$x_{2111}=1$	$x_{3111}=1$
	2	$x_{1212}=1$	$x_{2212}=1$	$x_{3122}=1$
	3	$x_{1223}=1$	$x_{2223}=1$	$x_{3133}=1$
	4	$x_{1134}=1$	$x_{2124}=1$	$x_{3144}=1$
	5	$x_{1125}=1$	$x_{2135}=1$	$x_{3155}=1$

According to Table 5, the optimal objective function obtained by GAMS is 302.5 that is better than defuzzified objective function obtained by random solution (335).
Here, we solve some small size test problems with GAMS software and evaluate its computation time. In order to generate test problems, six parameters are considered. These parameters and their considered levels are demonstrated in Table-7.

Table 7- Levels for input parameters

Parameters	Levels
Total number of jobs (n)	3,4,5,10,15,20
Total number f stages (S)	2,3,4
Maximum number of machines (m)	Uniform [1,3]
Total number of job type (L)	Uniform [1,3]
Machine speed (V)	Uniform [0.5,1.5]
Processing time (P)	Uniform [10,50]

Because processing time is a fuzzy number, it is generated as follows:

$$\tilde{P}_{kt}=(P,P+1,P+2).$$

Previously, we explained how to calculate fuzzy due date.

Results of these numerical examples are demonstrated in Table 8. Computation time is the average computation time

in 5 run.

Table 8- computational results of GAMS for small size problems

Sample	Problem data		Average run time[a]	status
	Stage number	Job number		
1	2	3	00:00:0.2412	Optimal
2	3	3	00:00:0.6158	Optimal
3	4	3	00:00:0.4918	Optimal
4	2	4	00:00:0.6024	Optimal
5	3	4	00:00:1.278	Optimal
6	4	4	00:00:2.6852	Optimal
7	2	5	00:00:0.4108	Optimal
8	3	5	00:00:2.2886	Optimal
9	4	5	8:00:00	Local
10	2	6	00:00:1.0952	Optimal
11	3	6	00:00:16.6176	Optimal
12	4	6	12:00:00	Local
13	2	8	00:04:26.7766	Optimal
14	3	8	01:03:1.651	Optimal
15	4	8	12:00:00	Local
16	2	10	12:00:00	Local
17	3	10	12:00:00	Local
18	4	10	12:00:00	local

a: computation time (hour:minute:second)

Conclusions

In this paper, a mixed integer mathematical model was presented for bi-objective HFS scheduling problem.
The proposed mathematical model included unrelated and eligible machines along with fuzzy processing time and fuzzy due date. The objective is to minimize a linear combination of total completion time and maximum lateness.
The obtained results showed that the proposed mathematical model is able to solve a small size problem in a reasonable computation time. But when problem size is increased it takes more time to solve the problem in a way that the optimal solution for a HFS problem with 10 jobs was not achieved after 12 hours. So, it reveals that to solve a large size and complex bi-objective HFS scheduling problem with unrelated and eligible machines in fuzzy space, some other solving methods like meta-heuristic algorithms have to be used.

References

[1] Johnson S.M., "Optimal two- and three-stage production schedules with setup time included", *Naval Research Logistics Quarterly,*Vol. 1, No. 6, 1954, pp. 1-8.

[2] Marichelvam, M.K., Prabaharan, T. and Yang, X.X., "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", *J. of Applied Soft Computing*, Vol. 19, 2014, pp. 93-101.

[3] Portmann, M.C., Vignier, A., Dardilhac, D. and Dezalay, D., "Branch and bound crossed with GA to solve hybridfowshops", *European Journal of Operational Research,*Vol. 107, No. 2, 1998, pp. 389-400.

[4] Choi, S.H. and Wang, K., "Flexible flow shop scheduling with stochastic processing time: A decomposition-based approach", *J. of Computers & Industrial Engineering*,Vol. 63, No. 2, 2012, pp. 362-373.

[5] Balin, S., "Parallel machine scheduling with fuzzy processing time using a robust genetic algorithm and simulation", *J. of Information Sciences*, Vol. 181, No. 17, 2011, pp. 3551-3569.

[6] Arthanary, T.S. and Ramamurthy, K.G., "An extension of two machine sequencing problems", *J. of Operations Research,* Vol. 8, 1971, pp. 10-22.

[7] Gupta, J., "Two-stage hybrid flowshop scheduling problem", *J. of Operational Research Society,* Vol. 39, No. 4, 1988, pp.359-364.

[8] A. Brah, S. and L. Hunsucker, J., "Branch and bound algorithm for the flow shop with multiple processors", *European Journal Of Operational Research*, Vol. 51, No. 1, 1991, pp. 88-99.

[9] Riane, F., Artiba, A. and E Elmaghraby, S., "A hybrid three-stage fowshop problem: Efficient heuristics to minimize makespan", *European Journal of Operational Research*, Vol. 109, 1998, pp. 321-329.

[10] Ruiz, R., Serifoglu, F.S. and Urlings, T., "Modeling realistic hybrid flexible flowshop scheduling problems", *J. of Computers and Operations Research*, Vol. 35, No. 4, 2008, pp. 1151-1175.

[11] Amin-Naseri, M.R. and Beheshti-Nia, M.A., "Hybrid flow shop scheduling with parallel batching", *International Journal of Production Economics*, Vol. 117, No. 1, 2009, pp. 185-196.

[12] Mahdavi, I., Zarezadeh, V. and Shahnazari-Shahrezaei, P., "Flexible flowshop scheduling with equal number of unrelated parallel machines", *J. Of Industrial Engineering International*, Vol. 7, No. 13, 2011, pp. 74-83.

[13] Wang, H.M., Chou, F.D. and Wu, F.C., "A simulated annealing for hybrid flow shop scheduling with

multiprocessor tasks to minimize makespan", *International Journal of Advanced Manufacturing Technology*, Vol. 53, No. 5-8, 2011, pp. 761-776.

[14] Ziaeifar, A., Tavakkoli-Moghaddam, R. and Pichka, K., "Solving a new mathematical model for a hybrid flow shop scheduling problem with a processor assignment by a genetic algorithm", *International Journal of Advanced Manufacturing Technology*, Vol. 61, No. 1-4, 2012, pp. 339-349.

[15] Attar, S.F., Mohammadi, M. and Tavakkoli-Moghaddam, R., "Hybrid flexible flowshop scheduling problem with unrelated parallel machines and limited waiting time", *International Journal of Advanced Manufacturing Technology*, Vol. 68, No. 5-8, 2013, pp. 1583-1599.

[16] Mousavai, S.M., Zandieh, M. and Yazdani, M., "A simulated annealing/local search to minimize the makespan and total tardiness on a hybrid flowshop", *The International Journal Of Advanced Manufacturing Technology*, Vol. 64, No. 1, 2013, pp. 369-388.

[17] Wang, S. and Liu, M., "Two-stage hybrid flow shop scheduling with preventive maintenance using multi-objective tabu search method", *International Journal of Production Research*, Vol. 52, No. 5, 2014, pp. 1495-1508.

[18] Fadaei, M.and Zandieh, M., "Scheduling a bi-objective hybrid flow shop with sequence-dependent family setup time using metaheuristics", *Arabian Journal for Science and Engineering*, Vol. 38, No. 8, 2013, pp. 2233-2244.

[19] Wu, H.C., "Solving the fuzzy earliness and tardiness in scheduling problems by using genetic algorithms", *Expert Systems With Applications*, Vol. 37, No. 7, 2010, pp. 4860-4866.

[20] Behnamian, J. and Zandieh, M., "Earliness and Tardiness Minimizing on a Realistic Hybrid Flowshop Scheduling with Learning Effect by Advanced Metaheuristic", *Arabian Journal for Science & Engineering*, Vol. 38, No. 5, 2013, pp. 1229-1242.

[21] Balin, S., "Non-identical parallel machine scheduling with fuzzy processing time using genetic algorithm and simulation", *International Journal of Advanced Manufacturing Technology*, Vol. 61, No. 9-12, 2012, pp. 1115-1127.