Journal of Uncertain Systems
Vol.3, No.2, pp.114-122, 2009
Online at: www.jus.org.uk

A Heuristic Algorithm for Minimizing the Expected Makespan in Two-Machine Flow Shops with Fuzzy Processing Time

Seyed Reza Hejazi *, Saeed Emami, Ali Arkan
Department of Industrial & Systems Engineering, Isfahan University of Technology, Isfahan, 84156-83111, IRAN

Received 18 February 2008; Accepted 9 June 2008

Abstract

Production scheduling, with the objective of minimizing the makespan, is an important task in manufacturing systems. In the past, the processing time for each job was usually assumed to be exactly known, but in many real-world applications, processing times may vary dynamically due to human factors or operating faults. Fuzzy programming theory has been developed for flow shop scheduling problems with uncertain processing times. In this article, the processing times are described by triangular fuzzy numbers. A new heuristic algorithm is also developed and presented which draws upon Johnson's algorithm, the interval numbers concept, and an improved version of McCahon & Lee's algorithm. Numerical examples will be presented to show the effectiveness of the proposed algorithm as compared with McCahon and Lee's algorithm.

Keywords: production scheduling, fuzzy flow shop problem, interval number, makespan, fuzzy processing time

1 Introduction

Flow-shop scheduling problems frequently occur in real-world application environments [1]. In a flow shop, jobs are processed by a series of machines in a predefined order. Most of the literature on flow-shop problems deals with the makespan criterion: completing the last job as soon as possible.

The flow-shop problem has been proved to be NP-hard [2], which has prompted the development of many heuristics to provide a valuable and quick solution [3, 4, 5, 6, 7].

Here, the flow-shop problem with two machines is illustrated (Recall that the problem is strongly NP-hard for $m \geq 3$) [2]:

A set of jobs $\{1,2,...,n\}$, available at time zero has to be processed in a shop with m= 2 machines, *A* and *B*. Each job is processed first on *A* and next on *B*. No machine can process more than one job at a time, no job preemption is allowed, all setup times are included in the job processing times, and there is unlimited intermediate storage between the machines. The problem is to determine a job sequence (permutation) that minimizes the completion time of the last job, also known as the makespan. Assume that a_k and b_k denote the deterministic processing times of job *k* on machines *A* and *B*, respectively. In a breakthrough paper on scheduling theory, Johnson [8] proposed to apply the rule:

$$\text{job } i \text{ precedes job } j \text{ if: } \min(a_j, b_i) > \min(a_i, b_j) \qquad (1)$$

He showed that it yields a transitive ordering among the jobs, and every job sequence satisfying (1) has the minimum makespan.

This problem has held the attention of many researchers and has been extensively studied in literatures [9, 10]. A survey paper about the flow-shop problem was presented by Dudek *et al.* [11]. Morton and Pentico published a book on this problem [1].

Most of the methods proposed in the literature assume that all of the time parameters and relevant data are already exactly known.

A large number of deterministic scheduling algorithms have been proposed over the last decades to deal with flow shop scheduling problems with various objectives and constraints [12]. However, it is often difficult to apply

* Corresponding author. Email: rehejazi@cc.iut.ac.ir (S.R. Hejazi).

those algorithms to real-life flow shop problems. For example, in practice the processing times of jobs could be uncertain due to incomplete knowledge or uncertain environment, which implies that there exist various external sources and types of uncertainty. Several theories such as fuzzy-set theory [13, 14], probability theory, D-S theory [15], and approaches based on certainty factors [16], have been developed to account for uncertainty. Among them, fuzzy-set theory is more and more frequently used in intelligent control because of its simplicity and similarity to human reasoning [17]. This theory has been applied to many fields such as manufacturing, engineering, diagnosis, economics, among others [14, 18]. Although fuzzy set concepts are used mainly in linguistic domains, they can also be used in numerical domains by assigning each number a membership value. Examples are Gazdik's fuzzy network planning [19], Klein's fuzzy shortest path [20], Nasution's fuzzy critical path [21], and Hong *et al.*'s fuzzy LPT scheduling [22].

A number of fuzzy flow-shop models with heuristics have been proposed in the literature over the past few years [23, 24, 25]. Ishii *et al.* [26] investigate scheduling problems with fuzzy due-dates, that is, a generalized two machine open shop scheduling problem with fuzzy due-dates and an identical machine scheduling problem with fuzzy due-dates. Ishibuchi *et al.* [27] examined two fuzzy flow shop problems with fuzzy due dates. One fuzzy flowshop scheduling problem is to maximize the minimum grade of satisfaction over given jobs, and the other is to maximize the total grade of satisfaction. McCahon and Lee [28] proposed an algorithm to obtain the job sequence with minimum makespan for a two-machine flow shop problem with triangular fuzzy processing times. Yao and Lin [29] have proposed a fuzzy flow shop sequencing model based on statistical data.

Temiz and Erol [30] presented the branch-and-bound algorithm for a three-machine flow shop problem with fuzzy processing time.

In this paper, we consider a two-machine flow shop with triangular fuzzy processing time. The objective is to find a job sequence with minimum makespan. McCahon and Lee [28] proposed an algorithm for this problem, but their algorithm failed to find the optimal solution in some situations. Therefore, we intend to improve McCahon & Lee's work by applying the nearest interval approximation of the fuzzy processing times. This will expectedly lead to improved performance of the scheduling procedure.

The paper is organized as follows. In Section 2, we outline the fuzzy flow shop problem and the related fuzzy set operations. McCahon and Lee's algorithm is described in Section 3. In Section 4, we give our proposed algorithm. In Section 5, an extension numerical study will be carried out to illustrate the effectiveness of the proposed algorithm. More examples are discussed in Section 6. Finally, in Section 7, we conclude and discuss future research.

2 Problem Definition

2.1 Review of Related Fuzzy Set Operations

In this section, fuzzy set operations used in this paper are reviewed. Triangular membership functions are used here to represent the fuzzy processing time of tasks. A triangular fuzzy membership function can be denoted by $\tilde{P}=(a,b,c)$ (Fig. 1). The membership value of x represents how likely x is to occur given $\tilde{P}$. It is denoted by $\mu_{\tilde{P}}(x)$, and is calculated according to (2).

$$\mu_{\tilde{P}}(x)=\begin{cases}0, & x\le a \quad or\, x\ge c\\ \dfrac{x-a}{b-a}, & a<x\le b\\ \dfrac{c-x}{c-b}, & b<x<c.\end{cases} \tag{2}$$

Let us introduce the notion of all λ-cuts of a fuzzy set $\tilde{P}$, because it will be used in our algorithm. The λ-cuts of $\tilde{P}$, denoted by $\tilde{P}_\lambda$, is a crisp set consisting of those elements in $\tilde{P}$ whose membership values exceed level λ:

$$\tilde{P}_\lambda=\{x\,|\,\mu_{\tilde{P}}(x)\ge\lambda\}. \tag{3}$$

It means that $\tilde{P}_\lambda$ consists of elements in $\tilde{P}$ with a degree of at least λ.

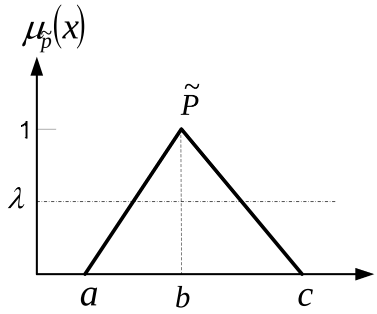

Fig. 1: Membership function of the fuzzy set $\tilde{P}$ describing the processing time of a job on a machine.

2.2 Fuzzy Flow Shop and Computation of Fuzzy Makespan

One kind of scheduling problem that frequently occurs in real world applications is the flow shop problem. Consider *n* different jobs that must pass through *m* processing machines. A machine can process one job at a time and we assume that the order of the jobs cannot be changed once the processing has begun. Given the time required for processing of each job on each machine, the question is in what order the jobs should be fed into the first machine in order to minimize the makespan (the total time required to process all the jobs).

Suppose we have a simple flow shop problem with jobs and only two machines. The objective is to find a job sequence which will achieve the minimum makespan. A solution sequence of jobs is a "permutation" of $1,2,...,n$, and we use an *n*-dimensional vector $(x_1, x_2,..., x_n)$ to represent the job sequence, where x_i represents the *i*th job, $i = 1,...,n$.

Let $\tilde{P}_{x_i,j}$ and $\tilde{C}_{x_i,j}$ be the fuzzy processing time and fuzzy completion time of job x_i on machine *j*, *j*=1,2, respectively, and that all fuzzy times have the triangular membership function. To calculate the completion time for each job, two basic fuzzy arithmetic operations are involved: fuzzy addition, denoted by $\tilde{+}$, and fuzzy max, denoted by $\tilde{\max}$. Let $\tilde{C}$ be the minimum makespan. If $\tilde{P}_1 = (a_1, b_1, c_1)$ and $\tilde{P}_2 = (a_2, b_2, c_2)$, then:

$$\tilde{P}_1 \tilde{+} \tilde{P}_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2) , \tag{4}$$

$$\tilde{\max}(\tilde{P}_1, \tilde{P}_2) = (\max(a_1, a_2), \max(b_1, b_2), \max(c_1, c_2)). \tag{5}$$

The mathematical formulation of the problem is as follows:

Find a sequence of jobs to achieve minimum makespan $\tilde{C}$, such that:

$$\tilde{C}_{x_1,1} = \tilde{P}_{x_1,1}, \tag{6}$$

$$\tilde{C}_{x_i,1} = \tilde{C}_{x_{i-1},1} \tilde{+} \tilde{P}_{x_i,1} , \quad for \quad i = 2,3,...,n , \tag{7}$$

$$\tilde{C}_{x_1,j} = \tilde{C}_{x_1,j-1} \tilde{+} \tilde{P}_{x_1,j} \quad for \quad j = 2,3,...,m , \tag{8}$$

$$C_{x_i,j} = \tilde{\max}(\tilde{C}_{x_{i-1},j}, \tilde{C}_{x_i,j-1}) \tilde{+} \tilde{P}_{x_i,j} \quad for \quad i = 2,3,...,n , \ j = 2,3,...,m . \tag{9}$$

The representation theorem in the fuzzy set theory states that any fuzzy set $\tilde{P}$ can be decomposed into a series of its λ-cuts, i.e., $\tilde{P} = \bigcup_{\lambda \in [0,1]} \lambda P^{\lambda}$. Conversely, any fuzzy set can be derived from a family of nested sets, providing that if $\lambda_1 \succ \lambda_2$, then $P^{\lambda_1} \subset p^{\lambda_2}$. This theorem implies that problems formulated in the framework of fuzzy sets can be solved by transforming these fuzzy sets into their corresponding families of λ-cuts. The partial results derived from all λ-cuts can be merged to obtain a solution to the original problem formulated with fuzzy sets.

Fuzzy makespan $\tilde{C}$ can be calculated by considering all of its λ-cuts, i.e., C^{λ}, $0 \le \lambda \le 1$. To calculate a makespan at λ-cuts, for simplicity, we use a triplet (l, m, u) to represent it, where *l* and *u*, respectively, represent the lower and upper bounds of the membership function of $\tilde{C}$, and *m* represent its modal point.

Yager index is used to evaluate fuzzy makespan $\tilde{C}$ obtained in McCahon and Lee's algorithm as presented here:

$$Y(\tilde{C}) = \frac{1}{2} \int_0^1 (C_l + C_u) \, d\lambda \cdot \tag{10}$$

3 McCahon and Lee's Algorithm

For a two-machine flow shop problem, Johnson's algorithm (see equation 1) gives an optimal job sequence with minimum makespan. In Johnson's algorithm, each processing time was considered to be a deterministic value.

To deal with the same problem with fuzzy job processing times, McCahon and Lee proposed to rank the fuzzy processing times using their corresponding Generalized Mean Values (GMVs), and then applied the modified Johnson's algorithm to account for the fuzzy processing time to get the optimal job sequence [13]. Given a triangular fuzzy processing time, i.e., $\tilde{P} = (a,b,c)$, its GMV, denoted by $G_{\tilde{P}}$, is equal to:

$$G_{\tilde{P}} = \frac{(a+b+c)}{3}. \tag{11}$$

However, the single GMV is not able to fully represent the uncertainties of fuzzy processing times and it might entail negative effects for obtaining an optimal sequence. We shall illustrate this point more clearly by presenting a simple example. Suppose we consider a simple 2-machine flow shop problem with only two jobs. Table 1 lists processing times of two jobs on two machines. *A* and *B* represent machines 1 and 2, respectively. The GMV of fuzzy processing times are calculated and shown in Table 2. According to McCahon and Lee's algorithm, the job sequence should be x_2, x_1 as shown in Table 3, and the achieved makespan is (35, 45, 87), which is obtained through the formulae (7)-(10) with 0.1 as the incremental step of λ. The calculated Yager's value of the makespan is 53. However, if we change the job sequence to x_1, x_2 as shown in Table (4), the achieved makespan is (31, 39, 80) with the Yager's value being 47.25, meaning that this job sequence is optimal.

This example shows that McCahon and Lee's algorithm fails to find the optimal solution because a single GMV is not able to fully reflect the uncertainties of fuzzy processing times. Especially, GMV has a limitation in fully describing the relationship between fuzzy processing times.

Table 1: Problem (2 jobs, 2 machines with processing times described by triangular fuzzy sets)

Machine \ Job	x_1	x_2
A	(1,2,36)	(5,8,23)
B	(21,25,28)	(9,12,16)

Table 2: GMVs of jobs

Machine \ Job	x_1	x_2
A	13	12
B	24.7	12.3

Table 3: McCahon and Lee's solution: sequence x_2 , x_1

Machine \ Job	x_2	x_1
A	(5,8,23)	(6,10,59)
B	(14,20,39)	(35,45,87)

Table 4: Optimal solution: Sequence x_1, x_2

Machine \ Job	x_1	x_2
A	(1,2,36)	(6,10,59)
B	(22,27,64)	(31,39,80)

4 Our Proposed Algorithm

For an illustration of our algorithm, we first describe the basic interval arithmetics in Section 4.1. Also, we need to define the nearest interval approximation of fuzzy numbers to be described in Section 4.2. The description of our proposed algorithm will then follow.

4.1 The Basic Interval Arithmetic

Throughout this paper, closed intervals are denoted by upper case letters *A*, *B*, while real numbers are denoted by lower case letters. The set of all real numbers is denoted by *R*. An interval number is defined by an ordered pair of brackets as:

$$A = [a_1, a_2] = \{a : a_1 \le a \le a_2, a \in R\}, \tag{12}$$

where a_1, a_2 are, respectively, the left and right limits of *A*.

Definition 1: Consider two interval numbers $A = [a_1, a_2]$ and $B = [b_1, b_2]$, then:

$$A + B = [a_1 + b_1,\ a_2 + b_2], \tag{13}$$

$$\max(A, B) = [\max(a_1, b_1), \max(a_2, b_2)], \tag{14}$$

$$\min(A, B) = [\min(a_1, b_1), \min(a_2, b_2)]. \tag{15}$$

Definition 2: The interval number $A = [a_1, a_2]$ is preferred to $B = [b_1, b_2]$ if and only if

$$a_1 \ge b_1, a_2 \ge b_2. \tag{16}$$

It should be noted that there are four types of interval preference relations of which Definition 2 is the most popular. Definition 2 is depicted in Fig. 2 as an example.

Since the final preference is obtained by interval arithmetic, we might have many partial order relations between alternatives in some cases. The following definition must be considered in addition to Definition 2 if a decision maker wants to have as many linear order relations as possible.

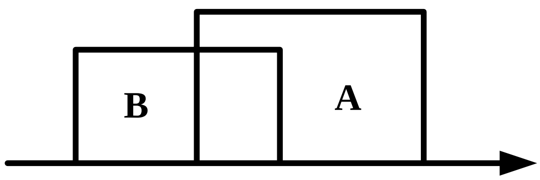

Fig. 2. An example of $A \ge B$

Definition 3: In the case where $A \supseteq B$ or $A \subseteq B$, the interval number $A = [a_1, a_2]$ is preferred to $B = [b_1, b_2]$ if and only if

$$a_1 + a_2 \ge b_1 + b_2, a_2 - a_1 \le b_2 - b_1, \tag{17}$$

in which the center of *A* is larger than that of *B* and the width of *A* is narrower than that of *B* [31].

4.2 The Nearest Interval Approximation

Here, we want to approximate a fuzzy number by a crisp number. Suppose $\tilde{A}$ and $\tilde{B}$ are two fuzzy numbers with λ-cuts $\tilde{A}_\lambda = [A_l(\lambda), A_u(\lambda)]$ and $\tilde{B}_\lambda = [B_l(\lambda), B_u(\lambda)]$, respectively. Then, the distance between $\tilde{A}$ and $\tilde{B}$ is:

$$d(\tilde{A}, \tilde{B}) = \sqrt{\int_0^1 (A_l(\lambda) - B_l(\lambda))^2 d\lambda + \int_0^1 (A_u(\lambda) - B_u(\lambda))^2 d\lambda} \tag{18}$$

We have to find a closed interval $C_d(\tilde{A})$ that is nearest to $\tilde{A}$ with respect to metric d. This is possible because each interval is also a fuzzy number with constant λ-cuts for all $\lambda \in [0,1]$. Hence, $(C_d(\tilde{A}))_\lambda = [C_l, C_u]$.

Now we have to minimize

$$d\left(\tilde{A}, C_d(A)\right) = \sqrt{\int_0^1 (A_l(\lambda) - C_l)^2 d\lambda + \int_0^1 (A_u(\lambda) - C_u)^2 d\lambda} \tag{19}$$

with respect to C_l and C_u.

In order to minimize $d\left(\tilde{A}, C_d(A)\right)$, it is sufficient to minimize the function $D(C_l, C_u) = \left(d^2\left(\tilde{A}, C_d(A)\right)\right)$. The first partial derivatives are:

$$\frac{\partial D(C_l, C_u)}{\partial C_l} = -2\int_0^1 (A_l(\lambda) - C_l) d\lambda = -2\int_0^1 A_l(\lambda) d\lambda + 2C_l$$

and

$$\frac{\partial D(C_l, C_u)}{\partial C_u} = -2\int_0^1 (A_u(\lambda) - C_u) d\lambda = -2\int_0^1 A_u(\lambda) d\lambda + 2C_u \cdot$$

Solving $\frac{\partial D(C_l, C_u)}{\partial C_l} = 0$ and $\frac{\partial D(C_l, C_u)}{\partial C_u} = 0$, we get:

$$C_L^* = \int_0^1 A_l(\lambda) d\lambda,\ C_u^* = \int_0^1 A_u(\lambda) d\lambda \,. \tag{20}$$

Since

$$H\left(C_L^*, C_U^*\right) = \frac{\partial^2}{\partial C_L^2} D\left(C_L^*, C_U^*\right) . \frac{\partial^2}{\partial C_U^2} D\left(C_L^*, C_U^*\right) - \left(\frac{\partial^2}{\partial C_L . \partial C_U} D\left(C_L^*, C_U^*\right)\right)^2 = 4 > 0,$$

so $D(C_l, C_u)$, i.e., $d\left(\tilde{A}, C_d(A)\right)$ is the global minimum.

Therefore, the interval

$$C_d\left(\tilde{A}\right) = \left[C_l = \int_0^1 A_l(\lambda) d\lambda \ ,\ C_u = \int_0^1 A_u(\lambda) d\lambda \right] \tag{21}$$

is the nearest interval approximation of the fuzzy number $\tilde{A}$ with respect to the metric d. Let $\tilde{A} = (a_1, a_2, a_3)$ be a fuzzy number. The λ -level interval of $\tilde{A}$ is defined as $\left(\tilde{A}\right)_\lambda = [A_l(\lambda), A_u(\lambda)]$. When $\tilde{A}$ is a TFN then: $A_l(\lambda) = a_1 + \lambda(a_2 - a_1)$ and $A_u(\lambda) = a_3 - \lambda(a_3 - a_2)$.

By the nearest interval approximation method, the lower limit of the interval is:

$$C_l = \int_0^1 A_l(\lambda) d\lambda = \int_0^1 [a_1 + \lambda(a_2 - a_1)] d\lambda = \frac{1}{2}(a_2 + a_1) \tag{22}$$

and the upper limit of the interval is:

$$C_u = \int_0^1 A_u(\lambda) d\lambda = \int_0^1 [a_3 + \lambda(a_3 - a_2)] d\lambda = \frac{1}{2}(a_2 + a_3) \,. \tag{23}$$

Therefore, the interval number considering $\tilde{A}$ as a TFN (Triangular Fuzzy Numbers) is: $[(a_1 + a_2)/2, (a_2 + a_3)/2]$.

Similarly, when $\tilde{A}$ is a PFN (Parabolic fuzzy numbers), then

$$A_l(\alpha) = a_2 - (a_2 - a_1)\sqrt{(1 - \lambda)},\quad A_u(\alpha) = a_2 - (a_3 - a_2)\sqrt{(1 - \lambda)}.$$

Following the procedure stated above, the interval number is $[(2a_1 + a_2)/3, (a_2 + 2a_3)/3]$.

4.3 The Heuristic Algorithm Based on the Nearest Interval Approximation of Fuzzy Numbers

The algorithm used for solving the problem is summarized as follows:
Input: processing times of two jobs on two machines.
Output: a solution of flow shop scheduling problems that minimizes the maximum flow time (makespan).
Our proposed algorithm:

Step 1: Convert all fuzzy processing times to internal numbers by using the method described in Section 4.2. (e.g. for triangular fuzzy numbers, apply relations 22 and 23).
Step 2: Calculate $\min(A_1, B_2)$ and $\min(A_2, B_1)$. Then rank the two interval numbers ($\min(A_1, B_2)$ and $\min(A_2, B_1)$) using Definitions 2 or 3, and then use Johnson's algorithm to determine an optimal job sequence.
Step 3: In this step, based on the optimal job sequence obtained in Step 2, calculate the makespan. It should be noted that all fuzzy processing times are converted to nearest interval approximation; thus, the makespan is also calculated as an interval number.

Let $P_{x_i,j} = [p_1, p_2], i = 1, 2, ..., n, j = 1, 2$ and $C_{x_i,j} = [c_1, c_2], i = 1, 2, ..., n, j = 1, 2$ be the fuzzy processing time and fuzzy completion time of job x_i on machine j, $j = 1,2$, respectively. The mathematical formulation of the problem is as follows: find a sequence of jobs to achieve the minimum makespan $(C_{x_n,2})$ such that:

$$C_{x_1,1} = P_{x_1,1}, \tag{24}$$

$$C_{x_i,1} = C_{x_{i-1},1} + P_{x_i,1} \quad for \;\; i = 1.2, ..., n, \tag{25}$$

$$C_{x_1,2} = C_{x_1,1} + P_{x_1,2}, \tag{26}$$

$$C_{x_i,2} = \max\left(C_{x_{i-2},2}, C_{x_i,1}\right) + P_{x_i,2} \quad for \;\; i = 1.2, ..., n. \tag{27}$$

The makespan is an interval number; thus, in order to evaluate the makespan obtained, we use the mean value of the left and right limits of the interval number as follows (number approximation of the interval number):

$$C_{x_n,2} = [c_1, c_2] = \frac{c_1 + c_2}{2} \tag{28}$$

5 Illustration of the Proposed Algorithm

We use the problem presented in Section 3 to illustrate our algorithm.
Step 1: All fuzzy processing times are converted to interval numbers as shown in Table (5).

Table 5: Interval number of jobs

Machine \ J	1	2
A	[1.5,19]	[6.5,15.5]
B	[23,26]	[10.5,14]

Step 2: The optimal sequence is obtained by Johnson's algorithm as follows:

$$\min(A_1, B_2) < \min(A_2, B_1) \Rightarrow$$
$$\min([1.5,19],[10.5,14]) = [1.5,14] < \min([23,26],[6.5,15.5]) = [6.5,15.5].$$

Thus, the job sequence should be x_1, x_2 (Table 6).
Step 3: Calculate the makespan. The makespan achieved is [35, 59.5] obtained through the formulae 24-27. The calculated approximation number is 47.5 obtained through (28).

Table 6: Solution (sequence x_1, x_2)

Machine \ J	1	2
A	[1.5,19]	[8,34.5]
B	[24.5,45.5]	[35,59.5]

We will show below that the results thus obtained are more satisfactory than those obtained from by McCahon and Lee's algorithm and, further, that ours is the optimal solution.

6 Computational Results

The algorithm described in Section 4.3. was tested and compared with both full search and McCahon & Lee's methods. The method was coded in MATLAB 7.0 (The Math Works, Inc. 2004) and experiments were executed on a Pentium IV 3.2 GHz processor with 1GB RAM. We conducted simulation experiments in which 50 instances were randomly generated for n=3, 4, 5, 6, 7 jobs. Because the full search method is computationally intractable for high numbers of jobs, we had to use small size problems to evaluate the efficiency of our algorithm. In other words, as the problem size increases, finding an optimal solution in reasonable time becomes extremely difficult.

While all the methods under consideration yielded identical results in 49 example problems, in 1 example, however, the results obtained from the McCahon & Lee's method was worse than those obtained from other methods. This example is shown in Table (7). Six jobs are to be scheduled in a two-step operation. Each job has two tasks that must be executed consecutively on two machines.

Table 7: Triangular fuzzy processing time of the problem

Machine \ Job	1	2	3	4	5	6
A	(1,14,40)	(20,33,35)	(3,7,19)	(18,24,37)	(17,21,32)	(13,16,36)
B	(12,16,18)	(50,60,65)	(24,30,40)	(7,10,15)	(8,11,14)	(3,8,20)

Table 8 shows the job sequences obtained by the McCahon & Lee’s algorithm, our algorithm, and the full search method. It shows that our algorithm gave a new sequence with improved makespan compared with McCahon and Lee’s solution; the solution was equal to that of the full search method.

Table 8: Results

Solving method	**Job sequence**	**makespan**
full search’s solution	3-2-1- 6-5-4	147
our proposed algorithm	3-2-1- 6-5-4	147
McCahon and Lee’s algorithm	3-2-1-5-4-6	154

7 Conclusion

Flow shop scheduling problems under uncertainty are discussed in this paper. We use fuzzy sets to describe uncertain processing times in two machine flow shop problems. The goal is to determine a job permutation that minimizes the makespan. We have proposed a new scheduling algorithm using the interval numbers concept that improves McCahon & Lee’s work by applying the nearest interval approximation of the fuzzy processing times. Unlike the McCahon & Lee’s algorithm that converts all fuzzy processing times to exact numbers using the GMV concept, in our heuristic algorithm, all fuzzy processing times are converted to interval numbers. This operation avoids missing information on fuzzy processing times. Computational results show that our heuristic algorithm is better than McCahon & Lee’s.

In this paper, we assume that a fuzzy set is continuous and we use a triangular membership function. However, the algorithm can be easily extended to handle other types of membership functions, while calculating GMV for other types of fuzzy numbers is difficult. It may be possible to consider other task constraints, such as set-up times, due dates, and priorities in future research.

References

[1] Morton, T.E., and D.W. Pentico, *Heuristic Scheduling Systems with Applications to Production Systems and Project Management*, John Wiley & Sons Inc, New York,1993.

[2] Garey, M.R.D., D. Johnson, and R. Sethi, The complexity of flowshop and jobshop scheduling, *Mathematics of Operations Research*, vol.1, pp.117–129, 1976.

[3] Dannenbring, D., An evaluation of flow shop sequencing heuristics, *Management Science*, vol.23, pp.1174–1182, 1977.

[4] Ho, J., and Y. Chang, A new heuristic for the *n*-job *m*-machine flow shop problem, *European Journal of Operational Research*, vol.52, pp.194–202, 1991.

[5] Hundal, T.S., and J. Rajgopal, An extension of Palmer's heuristic for the flow-shop scheduling problem, *International Journal of Production Research*, vol.26, pp.1119–1124, 1998.

[6] Nawaz, M., E. Enscore, and I. Ham, A heuristic algorithm for the *m*-machine *n*-job flow shop sequencing problem, *Omega*, vol.11, pp.11–95, 1983.

[7] Taillard, E., Some efficient heuristic methods for the flowshop sequencing problem, *European Journal of Operational Research*, vol.47, pp.65–74, 1990.

[8] Johnson, S.M., Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, vol.1, pp.61-68, 1954.

[9] Baker, K.R., *Introduction to Sequencing and Scheduling*, Wiley, New York, 1974.

[10] Coffman, E.G. , *Computer and Job-Shop Scheduling Theory*, Wiley, New York, 1976.

[11] Dudek, R.A., S.S. Panwalkar, and M.L. Smith, The lessons of flowshop scheduling research, *Operations Research*, vol.40, pp.7–13, 1992.

[12] Pinedo, M., *Scheduling Theory, Algorithms, and Systems*, Second Edition, Prentice Hall, 2002.

[13] Zimmermann, H.J., *Fuzzy sets, decision making and expert systems*, Kluwer Academic Publishers, Boston, 1987.

[14] Zimmermann, H.J., *Fuzzy set theory and its applications*, Kluwer Academic Publishers, Boston, 1991.

[15] Klir, G.J., and T.A. Folger, *Fuzzy sets uncertainty and information*, Prentice-Hall, 1992.

[16] Buchanan, B.G., and E.H. Shortli, *Rule-based Expert System: The MYCIN Experiments of the Stanford Heuristic Programming Projects*, Addison-Wesley, 1984.

[17] Graham, I., and P.L. Jones, *Expert Systems: Knowledge, Uncertainty and Decision*, Chapman and Computing, Boston, 1988.

[18] Zadeh, L.A., Fuzzy logic, *IEEE Computer*, vol.21, pp.83-93, 1988.

[19] Gazdik, I., Fuzzy-network planning, *IEEE Transactions on Reliability*, vol.32, no.3, pp.304-313, 1983.

[20] Klein, C.M., Fuzzy shortest paths. *Fuzzy Sets and Systems*, vol.39, pp.27-41, 1991.

[21] Nasution, S.H., Fuzzy critical path method, *IEEE Transactions on Systems, Man and Cybernetics*, vol.24, no.1, pp.48-57, 1994.

[22] Hong, T.P., C.M. Huang, and K.M. Yu., LPT scheduling for fuzzy tasks, *Fuzzy Sets and Systems*, vol.97, pp.277-286, 1998.

[23] Nezhad, S.S., and R.G. Assadi, Preference ratio-based maximum operator approximation and its application in fuzzy flow shop scheduling, *Applied Soft Computing*, vol.8, pp.759–766, 2008.

[24] Cheng, J., G. Steiner, and P. Stephenson, A computational study with a new algorithm for the tree-machine permutation flow-shop problem with release times, *European Journal of Operational Research*, vol.130, pp.559-575, 2001.

[25] Hong, T.P., and T.T. Wang, Fuzzy flexible flow shops at two machine centers for continuous fuzzy domains, *Information Sciences*, vol.129, pp.227-237, 2000.

[26] Ishii, H., M. Tada, and T. Masuda, Two scheduling problems with fuzzy due dates, *Fuzzy Sets and Systems*, vol.46, pp.339–347, 1992.

[27] Ishibuchi, H., N. Yamamoto, *et al.*, Genetic algorithms and neighborhood search algorithms for fuzzy flow shop scheduling problems, *Fuzzy Sets and Systems*, vol.67, pp.81-100, 1994.

[28] McCahon, C.S., and E.S. Lee, Fuzzy job sequencing for a flow shop, *European Journal of Operational Research*, vol.62, pp.294-301, 1992.

[29] Yao, J.S., and F.T. Lin, Constructing a fuzzy flow-shop sequencing model based on statistical data, *International Journal of Approximate Reasoning*, vol.29, pp.215–234, 2002.

[30] Temiz, I. and S. Erol, Fuzzy branch-and-bound algorithm for flow shop scheduling, *Journal of Intelligent Manufacturing*, vol. 15, pp. 449-454, 2004.

[31] Sugihara, K., and H. Ishii, Interval priorities in AHP by interval regression analysis, *European Journal of Operational Research*, vol.158, pp.745-754, 2004.