European J. of Industrial Engineering

ISSN online: 1751-5262 - ISSN print: 1751-5254
https://www.inderscience.com/ejie

Two-machine flowshop scheduling with fuzzy processing times and flexible operations

Zeynep Uruk, Ayten Yılmaz Yalçıner

DOI: 10.1504/EJIE.2024.10052125

Article History:

Received:	24 May 2022
Accepted:	18 October 2022
Published online:	08 December 2023

Two-machine flowshop scheduling with fuzzy processing times and flexible operations

Zeynep Uruk* and Ayten Yılmaz Yalçıner

Department of Industrial Engineering,
Sakarya University,
Sakarya, Turkey
Email: uruk.zeynep@gmail.com
Email: ayteny@sakarya.edu.tr
*Corresponding author

Abstract: This paper considers a two-machine flowshop scheduling problem with fuzzy processing times and flexible operations to minimise makespan. The jobs have three operations, one of which is a flexible operation that can be processed on one of the machines. The flexible operation should be assigned to one of the machines. Moreover, the processing times on the machines are not fixed but fuzzy processing times which are assumed to have triangular possibility distributions. Firstly, a possibilistic mathematical model is proposed with fuzzy processing times and then it is converted to an auxiliary crisp model using the weighted average defuzzification method. For the large cases where the exact algorithm may not be efficient, a heuristic algorithm is proposed. [Received: 24 May 2022; Accepted: 18 October 2022]

Keywords: scheduling; flowshop; makespan; fuzzy processing times; flexible operations.

Reference to this paper should be made as follows: Uruk, Z. and Yalçıner, A.Y. (2024) 'Two-machine flowshop scheduling with fuzzy processing times and flexible operations', *European J. Industrial Engineering*, Vol. 18, No. 1, pp.100–119.

Biographical notes: Zeynep Uruk is a PhD student at Industrial Engineering, Sakarya University since September 2019. She has received her MSc in Industrial Engineering from Bilkent University, Turkey, in 2011 and BSc in Chemical Engineering from the Middle East Technical University, Turkey, in 2008.

Ayten Yılmaz Yalçıner is an Associate Professor in the Industrial Engineering Department, at Sakarya University. Her expertise area is manufacturing and service information systems. Her studies also include digitalisation, industry 4.0, artificial intelligence, supply chain and logistics. Technology and innovation management field is also her interested fields. She completed her PhD at Sakarya University Institute of Science, Industrial Engineering Department, with his thesis titled "An Enterprise Manufacturing Information Model Proposal" in 2008.

1 Introduction

A two-machine flowshop scheduling problem is studied under the objective of minimising makespan with fuzzy processing times and flexible operations. The jobs have three operations: the first operation must be performed on the first machine and the second one on the second machine. The third operation is a flexible operation, and it can be performed on one of the two machines, but preemption is not allowed. Moreover, in most of the deterministic scheduling problems in the literature, job processing times are represented as constant parameters, but this is rare in real-life systems. In general, the processing times can be estimated as an interval which can be represented by a fuzzy number. Therefore, the processing times on the machines are represented as triangular fuzzy numbers instead of fixed processing times in this study. There exist many studies considering flexible manufacturing system and fuzzy processing times separately in the scheduling literature. However, there is no study that considers both simultaneously. Therefore, this is thought to be the first study which combines flexible operations and fuzzy processing times.

Resources are intrinsically flexible or can be made flexible in some scheduling contexts, which implies they may be dynamically redistributed in a production process. This flexibility can improve system efficiency when the quantity of resource devoted to an activity determines task processing durations. A frequent example of flexibility is labour. Cross-training workers can increase labour flexibility. Cross-trained employees get the abilities needed to execute various activities connected with several processing facilities. Another example of flexibility is the automated CNC machines which are extremely adaptable production systems. When the necessary cutting tools are placed into the machine's tool magazine, CNC machines can execute a variety of activities. However, it may be impossible to store all the tools necessary to complete a single project due to the restricted capacity of tool magazines. Furthermore, buying several copies of them is not cost-effective because the tools are so expensive. As a result, certain tools may be placed on one machine and this machine can only perform the related operations. Dobson and Karmarkar (1989) investigated a simultaneous sequencing and resource allocation issue in which each operation had different processing durations and resource needs. Daniels and Mazzola (1994) examined a flowshop scenario, where resources have unlimited flexibility, which implies they may be assigned to any step of the manufacturing process. They took labour flexibility into account and cross-trained personnel to do all the essential procedures. Daniels et al. (2004) discussed partial labour flexibility, in which each worker can only do a fraction of the needed processes. Workers were cross-trained to do a subset of the assembly line jobs. Sodhi et al. (1994) demonstrated that the selection of tool loading to optimise routing flexibility is a minimum cost network flow problem where routing flexibility is a function of the number of alternative routes or the average workload per tool aggregated across tool types. Stecke (1983) proposed a set of five production planning problems that must be solved for efficient utilisation of a flexible manufacturing system. He studied the tool loading and machine grouping subjects.

Processing times are assumed to be stochastic in a stochastic flowshop problem and are treated as a random variable with a given probability distribution. Talwar (1967) proposed that the makespan was minimised in the stochastic flowshop problem if it was a two-machine version with exponential processing durations. The rule is known as Talwar's rule which includes sorting the jobs by the difference in mean processing rates in non-increasing order. Cunningham and Dutta (1973) demonstrated the optimality of Talwar's rule. Ku and Niu (1986) investigated a stochastic generalisation of this problem and demonstrated that Talwar's rule resulted in a minimum for a sufficient condition on the processing time distributions. Kalczynski and Kamburowski (2004) offered a generalised version of Johnson's and Talwar's rules for independent and Gompertz distributed processing times. Kalczynski and Kamburowski (2006) applied Talwar's rule for the Weibull distributed processing times. Adiri and Frostig (1984) examined a stochastic permutation-flowshop to minimise the schedule length in distribution. Sethi et al. (1993) proposed an asymptotic study of hierarchical production planning in a manufacturing system with two tandem machines in case of breakdown and repair. The two-machine stochastic flowshop issue with arbitrary distribution of processing time was investigated by Elmaghraby and Thoney (1999). Soroush and Allahverdi (2005) investigated a stochastic two-machine flowshop with distinctively, generally, and normally distributed processing times to minimise the total completion time. Baker and Trietsch (2011) examined Johnson's and Talwar's rules, using mean processing time, for the stochastic two-machine flowshop scheduling. They discovered that none of the two rules outperformed the other. Johnson's heuristic was found to be optimal when the processing time distributions did not overlap, but it became less effective as the overlap rises. On the contrary, Talwar's heuristic decreased average deviations as the overlap rises. Kenneth and Dominik (2012) employed three heuristic techniques for the *m*-machine flowshop issue with general processing time distributions. Johnson's rule was used by Portougal and Trietsch (2006) to solve stochastic scheduling problems. They used the mean processing time of each job in Johnson's rule as the deterministic job processing time to create an asymptotically optimal makespan.

McCahon and Lee (1990) used triangular and trapezoidal fuzzy numbers to represent job processing times in job shop production systems. They modified the job sequencing algorithms of Johnson and Ignall and Schrage in terms of fuzzy job processing times. Tsujimura et al. (1993) studied three-machine flowshop scheduling in which triangular fuzzy numbers are used to represent the processing times. They used the comparison method based on the dominance property to determine the ranking of the fuzzy numbers. Branch and bound algorithm is used to minimise makespan to illustrate the proposed methodology. Tsujimura et al. (1995) also discussed an assembly-line balancing problem by considering processing times to be triangular fuzzy numbers to represent the data of real world. They used genetic algorithms to solve the problem. Sakawa and Kubota (2000) introduced job shop scheduling problems with fuzzy due date and fuzzy completion time as three-objective functions which maximise the minimum agreement index, maximise the average agreement index, and minimise the maximum fuzzy completion time. Chanas and Kasperski (2003) studied single machine scheduling problems with fuzzy processing times and fuzzy due dates separately. They minimised the maximum value among the mean values of fuzzy tardiness of jobs in a sequence for the first problem. They minimised the expected value of a maximal fuzzy tardiness for the second problem. Temiz and Erol (2004) modified branch and bound algorithm of Ignall and Schrage to accept fuzzy processing times in three-machine flowshop scheduling. They determined minimum completion time with fuzzy arithmetic on fuzzy numbers. Petrovic and Song (2006) developed a new optimisation algorithm based on Johnson's algorithm and on an improvement of McCahon and Lee's (1990) algorithm to deal with two-machine flowshop problem in the presence of triangular fuzzy processing times. Razmia et al. (2009) presented a mathematical model for *m* machine and *n* jobs flowshop scheduling problem with fuzzy processing times in the objective of expectable makespan and the probability of minimising the makespan. They proposed a solution for the fuzzy model based on developing the one presented by McCahon and Lee (1990). Al-Faruk et al. (2011) investigated an approach which incorporates statistics with triangular fuzzy processing times in the flowshop problem under the objective of minimising the makespan. Gupta et al. (2012) developed a new heuristic algorithm for *n*-jobs two-machine flowshop scheduling problem in which processing times are triangular fuzzy numbers with the objective of minimising the utilisation time of machines and their rental cost under specified rental policy. Noori-Darvisha et al. (2012) presented a novel bi-objective possibilistic mixed-integer linear programming model for open shop scheduling problem having sequence-dependent setup times, fuzzy processing times and fuzzy due dates with triangular possibility distributions. The objective functions were minimising total weighted tardiness and total weighted completion times. For medium to large size examples, they proposed a multi-objective particle swarm optimisation algorithm. Ambika and Uthra (2014) studied a branch and bound technique in flowshop scheduling problem with triangular fuzzy processing times to minimise the total elapsed time. Rostami et al. (2015) investigated to minimise both total earliness tardiness and makespan in a non-identical parallel machine scheduling environment by considering deterioration and learning effects. They represented processing times and due dates as triangular fuzzy numbers. They formulated a nonlinear mathematical model based on fuzzy chance-constrained programming and offered a multi-objective branch and bound algorithm. Najari et al. (2018) addressed a permutation flowshop scheduling problem with aging and learning effects considering maintenance process to minimise the makespan, tardiness of jobs, tardiness cost while maximising net present value, simultaneously. They proposed two Pareto-based multi-objective evolutionary algorithms that are non-dominated ranked genetic algorithm and non-dominated sorting genetic algorithm. Behmanesha et al. (2019) discussed the surgical case scheduling problem in multioperating theatre environment with fuzzy duration times of all stages to minimise makespan. Fuzzy surgical case scheduling problem is like no-wait multi-resource fuzzy flexible job shop problem. Toksari and Arik (2017) studied single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times with the objectives of minimising total completion time, makespan and total weighted completion time. They used fuzzy mixed integer nonlinear programming and algorithms that are polynomially solvable. Li et al. (2019) addressed a scheduling problem of *n* single-operation jobs on *m* uniform parallel machines which minimises the makespan, with the total resource consumption constraint. They implemented a fuzzy simplified swarm optimisation algorithm and compared the performance with a particle swarm optimisation with genetic local search and a genetic algorithm from the literature. Geyik and Elibal (2017) considered an application of non-identical parallel processor scheduling with fuzzy processing times. Jia et al. (2019) considered scheduling of parallel batch processing machines with different capacities and fuzzy processing times under the objective of minimising the makespan. They implemented a fuzzy ant colony optimisation algorithm. Li et al. (2018) studied a single machine due window assignment scheduling problem with fuzzy processing times, window size and precedence constraints to minimise the mean value of the total earliness-tardiness penalties. Kumar et al. (2018) studied a flowshop scheduling problem with fuzzy processing times using a new hybrid optimisation algorithm combining branch and bound technique with genetic algorithm.

Gupta et al. (2004) studied scheduling three-operation jobs in a two-machine flowshop environment in which each job had a flexible operation and two fixed operations under the objective of minimising makespan. They analysed two approximation algorithms. The first algorithm applied arbitrary processing order and arbitrary assignment of the flexible operations which had a worst-case performance ratio of 2. The second algorithm, improved one, created four schedules and decided on the best which had a worst-case performance ratio of 3/2. Lastly, they offered a polynomial time approximation scheme. Crama and Gultekin (2010) considered three-operation jobs, one of which is flexible, in a two-machine flowshop environment to maximise the throughput rate for identical jobs. They studied different number of jobs to be processed and different capacity of buffers in between the machines. They proposed solution methods for each case of the problem. Gultekin (2012) also discussed the same problem with infinite and zero capacity buffers in between the machines, but under a different assumption, that the flexible operation had different processing times on each machine. He developed constant time solution procedures. Uruk et al. (2013) investigated two-machine flowshop problem with flexible operations and controllable processing times for identical jobs. They proposed two mathematical models to determine the assignment of flexible operations to the machines and processing times for each job simultaneously by considering a bicriteria objective of minimising makespan and the manufacturing cost. They also developed an efficient approximation algorithm. Khorasanian and Moslehi (2017) studied the two-machine flowshop scheduling problem with blocking, multi-task flexibility of the first machine, and pre-emption with the objective of minimising makespan. They formulated two mathematical models. They also offered a variable neighbourhood search algorithm and a dynamic variable neighbourhood search algorithm for large-sized problems. Wei et al. (2019) studied two-machine flowshop scheduling with identical jobs which consists of a flexible operation and a fixed operation under the makespan minimisation criteria. They considered that flexible task had different processing times on the two machines and no buffer and infinite buffer capacity cases are evaluated with constant-time solution algorithms.

The remainder of this paper is organised as follows: in Section 2, the problem is defined, formulated as a possibilistic mixed integer problem using triangular fuzzy processing times and defuzzified based on weighted average method. The objective of the model is minimising makespan value by scheduling the jobs and assigning the flexible operations. The model is formulated and solved in GAMS. In Section 3, some properties for the problem are demonstrated which will be used in the development of the heuristic algorithm. For the large instances, a heuristic approach is proposed for efficiency in term of computational time. The heuristic algorithm is coded and compiled in MATLAB. A computational study is performed in Section 4 to test the performance of the heuristic algorithm and to compare it with the mathematical formulation. Section 5 is devoted to concluding remarks.

2 Problem formulation

In this section, a mixed integer possibilistic mathematical model will be formulated under the objective of minimising makespan in flowshop scheduling with fuzzy processing times in the presence of flexible operations. Afterwards, it will be converted to a crisp model using the weighted average defuzzification method.

There exist n jobs which have three operations to be performed by the two machines. The first operation can only be performed by the first machine, the second operation can only be performed by the second machine and the third operation is flexible, which can be performed by both machines. Since assignment of the flexible operations to different machines yields a different scheduling performance, assignment of these flexible operations to the machines should be done for each job. Infinite capacity of buffer is assumed in between the machines and pre-emption is not allowed. In contrast to most of the studies in scheduling literature, processing times are selected within an interval instead of fixed processing times. Therefore, the processing times can exactly be represented by a fuzzy number which are assumed to be triangular fuzzy numbers. The objective is to determine the schedule for the jobs with the assignment of flexible operations of each job to one of the machines under the objective of minimising the makespan.

The notation used throughout the paper is as follows.

Decision variables

$T_{j,m}$ starting time of j^{th} job on machine m

x_j decision variable which controls if flexible operation of job j is assigned to machine 1.

Parameters

n number of jobs to be processed

$\tilde{f}_j^1$ fuzzy processing time for 1st operation of job j on machine 1

$\tilde{f}_j^2$ fuzzy processing time for 2nd operation of job j on machine 2

$\tilde{s}_j$ fuzzy processing time for flexible operation of job j on one of the machines.

The processing times are considered as fuzzy parameters with triangular possibility distributions as follows:

$$\tilde{f}_i^j = \left(f_j^{ip}, f_j^{im}, f_j^{io}\right) \qquad \text{for } i = 1, 2 \text{ and } j = 1 \text{ to } n$$

$$\tilde{s}_j = \left(s_j^p, s_j^m, s_j^o\right) \qquad \text{for } j = 1 \text{ to } n$$

where $(f_j^{ip}, s_j^p), (f_j^{im}, s_j^m), (f_j^{io}, s_j^o)$ are most pessimistic values, moderate (most possible) values and most optimistic values, respectively, as illustrated in Figure 1. Decision maker determines these values.

The problem can be formulated as a possibilistic mixed integer model as follows:

$$\text{Min} \quad Z_2 = T_{n,2} + \tilde{f}_n^{52} + \tilde{s}_n \cdot (1 - x_n) \tag{1}$$

s.t.

$$T_{j,1} \geq T_{j-1,1} + \tilde{f}^1_{j-1} + \tilde{s}_{j-1} \cdot x_{j-1} \qquad j \geq 2 \tag{2}$$

$$T_{j,2} \geq T_{j-1,2} + \tilde{f}^2_{j-1} + \tilde{s}_{j-1} \cdot \left(1 - x_{j-1}\right) \qquad j \geq 2 \tag{3}$$

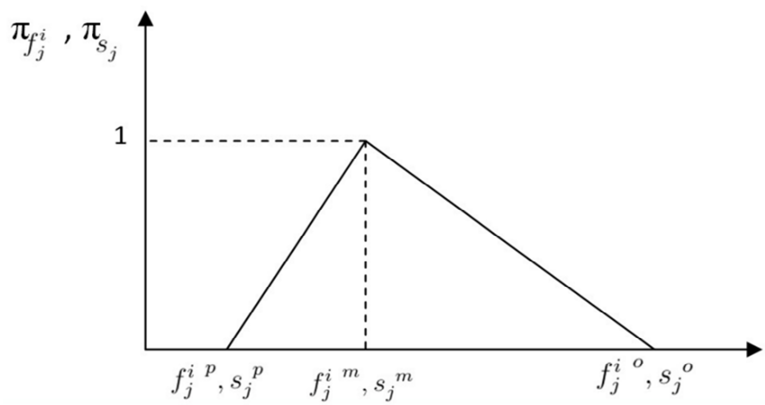
$$T_{j,2} \geq T_{j,1} + \tilde{f}^1_j + \tilde{s}_j \cdot x_j \qquad \forall j \tag{4}$$

$$T_{1,1} \geq 0 \tag{5}$$

$$x_j \in \{0, 1\} \qquad \forall j \tag{6}$$

Equation (1) is the objective function which minimises makespan. Constraint (2) states that the j^{th} job can start on the first machine only after the previous job is completed on the first machine. Similarly, constraint (3) represents the same condition for the second machine. Constraint (4) also explains the condition that the processing of a job on the second machine can be started only after the processing of this job is completed on the first machine. Constraint (5) is the non-negativity constraint of the variable $T_{1,1}$. Finally, constraint (6) expresses the assignment of flexible operation to only one of the two machines for each job.

Figure 1 Triangular possibility distribution of fuzzy processing times



Since the proposed mathematical model is possibilistic, it should first be converted to a crisp model. The weighted average method with the most likely solution parameters is used for the defuzzification of the fuzzy processing times into crisp ones. Lai and Hwang (1992) indicated that β is the minimal acceptable possibility and the events with the possibility more than or equal to β are acceptable events. They also defined w_1, w_2 and w_3 to be the weights of the most pessimistic, the moderate (most possible) and the most optimistic value of fuzzy parameters, respectively, and $w_1 + w_2 + w_3 = 1$. They provided a most likely solution by setting the parameters as $\beta = 0$, $w_1 = 1/6$, $w_2 = 4/6$ and $w_3 = 1/6$.

After defuzzification, the mixed integer model becomes:

$$\begin{aligned} \text{Min} \quad Z_2 = T_{n,2} &+ \left(w_1 \cdot f^2_{n,\beta}{}^p + w_2 \cdot f^2_{n,\beta}{}^m + w_3 \cdot f^2_{n,\beta}{}^o\right) \\ &+ \left(w_1 \cdot s^p_{n,\beta} + w_2 \cdot s^m_{n,\beta} + w_3 \cdot s^o_{n,\beta}\right) \cdot \left(1 - x_n\right) \end{aligned} \tag{7}$$

s.t.

$$\begin{aligned} T_{j,1} \geq T_{j-1,1} &+ \left(w_1 \cdot f^1_{j-1,\beta}{}^p + w_2 \cdot f^1_{j-1,\beta}{}^m + w_3 \cdot f^1_{j-1,\beta}{}^o \right) \\ &+ \left(w_1 \cdot s^p_{j-1,\beta} + w_2 \cdot s^m_{j-1,\beta} + w_3 \cdot s^o_{j-1,\beta} \right) \cdot x_{j-1} \qquad j \geq 2 \end{aligned} \tag{8}$$

$$\begin{aligned} T_{j,2} \geq T_{j-1,2} &+ \left(w_1 \cdot f^2_{j-1,\beta}{}^p + w_2 \cdot f^2_{j-1,\beta}{}^m + w_3 \cdot f^2_{j-1,\beta}{}^o \right) \\ &+ \left(w_1 \cdot s^p_{j-1,\beta} + w_2 \cdot s^m_{j-1,\beta} + w_3 \cdot s^o_{j-1,\beta} \right) \cdot \left(1 - x_{j-1}\right) \qquad j \geq 2 \end{aligned} \tag{9}$$

$$\begin{aligned} T_{j,2} \geq T_{j,1} &+ \left(w_1 \cdot f^1_{j,\beta}{}^p + w_2 \cdot f^1_{j,\beta}{}^m + w_3 \cdot f^1_{j,\beta}{}^o \right) \\ &+ \left(w_1 \cdot s^p_{j,\beta} + w_2 \cdot s^m_{j,\beta} + w_3 \cdot s^o_{j,\beta} \right) \cdot x_j \qquad \forall j \end{aligned} \tag{10}$$

$$T_{1,1} \geq 0 \tag{11}$$

$$x_j \in \{0, 1\} \qquad \forall j \tag{12}$$

Above crisp mixed integer mathematical model is formulated in GAMS and solved with CPLEX solver. Detailed results will be presented in Section 4.

3 Heuristic algorithm

In this paper, a two machine flowshop environment is studied with n non-identical jobs having fuzzy processing times to minimise makespan. There exist three operations for each job, one of which is a flexible operation that can be processed on both machines. The other two operations are fixed operations such that the first operation can only be performed by the first machine and the second operation can only be performed by the second machine.

This problem is NP-hard, so development of heuristic algorithms is necessary for large-scale problems. Suppose that the processing times of the fixed operations are set to zero. Then, there remains just one operation that should be assigned to one of the two machines for each job. In this case, the problem becomes scheduling n jobs on two identical parallel machines to minimise the makespan. This problem is proved to be NP-hard, which means the problem discussed in this paper is also NP-hard.

Johnson's (1954) rule minimises the completion time for n-jobs in two machines flowshop environment. However, processing times must be known and constant for each job on each machine. For the case of this paper, the processing times are fuzzy numbers and there exists flexible operations that should be assigned to the machines, so Johnson's rule cannot solve the problem as itself optimally. However, it will be used as an approximation of schedule after defuzzification of processing times for the scheduling of jobs without considering the assignment of flexible operations.

Crama and Gultekin (2010) formulated the optimal assignment of the flexible operation for each identical job. f^1, f^2, and s are the processing times of first, second and flexible operations, respectively. They defined r to be the total number of parts for which the flexible operation is assigned to the first machine. Following Johnson's rule, the first $n - r$ flexible operations should be assigned to the second machine and the remaining r jobs to the first machine. In this paper, the jobs are non-identical which means they all have different processing times on machines. Therefore, the formulation of Crama and Gultekin (2010) cannot be used to obtain the optimal assignment of flexible operations directly, but it will be modified to obtain an approximation of assignments.

Input: $\tilde{f}_j^1, \tilde{f}_j^2, \tilde{s}_j, n, \beta, w_1, w_2, w_3$

Output: X_j, $T_{j,m}$, *Makespan*

1 Use weighted average method for defuzzification of $\tilde{f}_j^1, \tilde{f}_j^2, \tilde{s}_j$.

Let $p_j^1 = w_1 \cdot f_{j,\beta}^{1}{}^{p} + w_2 \cdot f_{j,\beta}^{1}{}^{m} + w_3 \cdot f_{j,\beta}^{1}{}^{o}$

Let $p_j^2 = w_1 \cdot f_{j,\beta}^{2}{}^{p} + w_2 \cdot f_{j,\beta}^{2}{}^{m} + w_3 \cdot f_{j,\beta}^{2}{}^{o}$

Let $p_j^s = w_1 \cdot s_{j,\beta}^{p} + w_2 \cdot s_{j,\beta}^{m} + w_3 \cdot s_{j,\beta}^{o}$

2 Follow below steps to carry out Johnson's algorithm (Johnson, 1954)

a Create an array $M_1 = [j \mid p_j^1 < p_j^2]$

b Create another array $M_2 = [j \mid p_j^1 \geq p_j^2]$

c Sort the elements in M_1 in the non-decreasing order of p_j^1

d Sort the elements in M_2 in the non-increasing order of p_j^2

e Create the schedule of jobs by an array $S = [M_1 M_2]$

3 Assign $p_{sn}^{1}{}^{s}$, $p_{sn}^{2}{}^{s}$ and p_{sn}^{s} for $sn = 1$ to n, where sn represents sequence number, according to the schedule S.

4 Find an approximation for the assignment of flexible operations

$$s_{avr} = \sum_{j=1}^{n} \frac{p_j^s}{n}$$

$$r = \frac{\left(\sum_{sn=1}^{n-1} p_{sn}^{2}{}^{s} - \sum_{sn=2}^{n} p_{sn}^{1}{}^{s} + n \cdot s_{avr}\right)}{2.s_{avr}}$$

5 Round down r, label as r_1 and assign the flexible operations to the machines according to r_1.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute $Makespan_1$.

6 Let $r_2 = r_1 - 1$ and assign the flexible operations to the machines according to r_2.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute $Makespan_2$.

7 Let $r_3 = r_2 - 1$ and assign the flexible operations to the machines according to r_3.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute $Makespan_3$.

8 Round up r, label as r_4 and assign the flexible operations to the machines according to r_4.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute $Makespan_4$.

9 Let $r_5 = r_4 + 1$ and assign the flexible operations to the machines according to r_5.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute *Makespan*$_5$.

10 Let $r_6 = r_5 + 1$ and assign the flexible operations to the machines according to r_6.

Compute starting times of jobs on each machine $T_{j,m}$.

Compute *Makespan*$_6$.

11 *Makespan* = min{*Makespan*$_1$, *Makespan*$_2$, *Makespan*$_3$, *Makespan*$_4$, *Makespan*$_5$, *Makespan*$_6$}.

12 Report starting times of jobs on each machine $T_{j,m}$ accordingly.

13 Report assignment of flexible operations X_j accordingly.

Heuristic algorithm is given as a pseudocode in this section. It starts with defuzzification of fuzzy processing times using weighted average method with the parameters $\beta = 0$, $w_1 = 1/6$, $w_2 = 4/6$ and $w_3 = 1/6$. Then, assigns the crisp processing times into new variables as p_j^1, p_j^2 and p_j^s for first, second, and flexible operations, respectively. Afterwards, in Steps 2a–2e, Johnson's (1954) algorithm is used without considering the flexible operations, that means the processing times of flexible operations are assumed to be zero and the schedule is arranged using just p_j^1 and p_j^2. The indices of jobs are assigned to array M_1 if $p_j^1 < p_j^2$ and the indices of jobs are assigned to array M_2 if $p_j^1 \geq p_j^2$. The indices in M_1 is sorted in the non-decreasing order of p_j^1 and the indices in M_2 is sorted in the non-increasing order of p_j^2. An approximation of the schedule can be obtained as $S = [M_1M_2]$. Then, the algorithm assigns the flexible operations, in Steps 3 and 4, by modifying the formulation of Crama and Gultekin (2010). The modification is as follows.

Since the proposed algorithm firstly finds an approximation schedule without considering the flexible operation, the schedule of the flexible operation without machine assignment is also known. Let the processing times of the scheduled jobs to be $p_{sn}^{1\,s}$, $p_{sn}^{2\,s}$ and $p_{sn}^{s\,s}$, for $sn = 1$ to n, where sn represents sequence number. The idle time on the machines while processing of the first operation of the first job and the processing of the last operation of the last job cannot be avoided, which are p_1^{1s} and p_n^{2s}, respectively. There remains a total processing time $\sum_{sn=2}^{n} p_{sn}^{1\,s} + \sum_{sn=1}^{n-1} p_{sn}^{2\,s} + \sum_{sn=1}^{n} p_{sn}^{s\,s}$ to be processed. The total processing time assigned to the first and second machines become $\left(\sum_{sn=2}^{n} p_{sn}^{1\,s} + \sum_{sn=n-r+1}^{n} p_{sn}^{s\,s}\right), \left(\sum_{sn=1}^{n-1} p_{sn}^{2\,s} + \sum_{sn=1}^{n-r} p_{sn}^{s\,s}\right)$, respectively. Hence, makespan is at least $p_1^{1s} + \max\left(\left(\sum_{sn=2}^{n} p_j^{1s} + \sum_{sn=n-r+1}^{n} p_{sn}^{s\,s}\right), \left(\sum_{sn=2}^{n} p_{sn}^{1\,s} + \sum_{sn=1}^{n-r} p_{sn}^{s\,s}\right)\right) + p_n^{2s}$. Makespan takes the smallest value when the workload is balanced as evenly as possible between the two machines. This means, to find the assignment of flexible operations r should be selected such that it minimises $\left(\sum_{sn=2}^{n} p_{sn}^{1\,s} + \sum_{sn=n-r+1}^{n} p_{sn}^{s\,s} - \sum_{sn=1}^{n-1} p_{sn}^{2\,s} - \sum_{sn=1}^{n-r} p_{sn}^{s\,s}\right)$. However, to find an approximation of assignments, the algorithm computes average of flexible operation processing times of jobs and assigns it as s_{avr}.

Then, $\sum_{sn=2}^{n} p_{sn}^{1\ s} + \sum_{sn=n-r+1}^{n} p_{sn}^{s\ s} - \sum_{sn=1}^{n-1} p_{sn}^{2\ s} - \sum_{sn=1}^{n-r} p_{sn}^{s\ s}$ reduces to $\sum_{sn=2}^{n} p_{sn}^{1\ s}$ $+r \cdot s_{avr} - \sum_{sn=1}^{n-1} p_{j}^{2s} - (n-r) \cdot s_{avr}$ and r can be calculated as

$$r = \left(\sum_{sn=1}^{n-1} p_{sn}^{2\ s} - \sum_{j=2}^{n} p_{sn}^{1\ s} + n \cdot s_{avr} \right) \Big/ (2 + s_{avr}).$$

Afterwards, the heuristic algorithm checks different assignments of flexible operations by changing r value, which is the number of flexible operations assigned to first machine, to find a better makespan value. Firstly, in Step 5, the algorithm rounds down r, and labels it as r_1, and assigns the flexible operations to the machines according to r_1. Then, starting times of jobs on each machine, $T_{j,m}$, is computed. Makespan is calculated accordingly and labelled as $Makespan_1$. To check whether a smaller r values give a smaller makespan, r_1 is decreased by 1, in Step 6, and starting times and makespan ($Makespan_2$) are calculated again. The same procedure is applied to find $Makespan_3$ in Step 7. In the next step, r value is rounded up and labelled as r_4, and flexible operations are assigned to the machines according to r_4, in Step 8. Then, starting times of jobs on each machine, $T_{j,m}$, is computed. Makespan is calculated accordingly and labelled as $Makespan_4$. This time, r_4 is increased by 1 and starting times and makespan ($Makespan_5$) are calculated again, in Step 9. Lastly, $Makespan_6$ is computed by increasing r_5 by 1 in Step 10. Makespan is reported to be minimum of {$Makespan_1$, $Makespan_2$, $Makespan_3$, $Makespan_4$, $Makespan_5$, $Makespan_6$} in Step 11. Starting times of jobs on each machine $T_{j,m}$ and assignment of flexible operations X_j are reported accordingly.

4 Computational results

In this section, a computational study is performed to test the performance of the proposed heuristic algorithm by comparing it with the mathematical formulation. Mixed integer program is formulated in GAMS 2.25 and solved with CPLEX solver using a computer with 2 GB memory and Intel Pentium processor with 2.13 GHz CPU. Heuristic algorithm is coded in and complied with MATLAB R2015b using the same computer.

The experimental factor n determines the size of the problem. When the problem size is large, more CPU time is needed to solve the problem and the objective function value (makespan) is expected to be high. Since exact solutions need huge CPU times for large problem sizes, heuristic algorithms are proposed which presents qualified solutions in small CPU times. The other parameters are selected randomly from the intervals as listed in Table 1 where $U[a, b]$ is uniform distribution in interval $[a, b]$.

Table 1 Fuzzy processing times for each operation

Parameters	*Most pessimistic*	*Most possible*	*Most optimistic*
$\tilde{f}_j^1$	$U[1.2, 1.7]$	$U[2.0, 2.5]$	$U[2.8, 3.3]$
$\tilde{f}_j^2$	$U[1.4, 1.9]$	$U[2.2, 2.7]$	$U[3.0, 3.5]$
$\tilde{s}_j$	$U[1.6, 2.1]$	$U[2.4, 2.9]$	$U[3.2, 3.7]$

4.1 An example for heuristic algorithm

In this section, an example is presented for a two-mahine flowshop problem with five jobs to illustrate the heuristic algorithm proposed. The parameters are selected to be $\beta = 0$, $w_1 = 1/6$, $w_2 = 4/6$, $w_3 = 1/6$ as mentioned before in Section 2. Fuzzy processing times for each operation are selected from the intervals presented in Table 1. Table 2 presents pessimistic, moderate, and optimistic values (in other words, fuzzy processing times) of 1st, 2nd, and flexible operation of each job. Moreover, defuzzified processing times are calculated in these tables according to weighted average method and summarised in Table 3.

Table 2 Fuzzy processing times of first operations of each job

j	${f^1_{j,\beta}}^p$	${f^1_{j,\beta}}^m$	${f^1_{j,\beta}}^o$	$w_1 \cdot {f^1_{j,\beta}}^p + w_2 \cdot {f^1_{j,\beta}}^m + w_3 \cdot {f^1_{j,\beta}}^o = p^1_j$
1	1.70	2.42	3.25	1/6 ∗ 1.70 + 4/6 ∗ 2.42 + 1/6 ∗ 3.25 = 2.44
2	1.66	2.39	2.88	1/6 ∗ 1.66 + 4/6 ∗ 2.39 + 1/6 ∗ 2.88 = 2.35
3	1.44	2.27	2.86	1/6 ∗ 1.44 + 4/6 ∗ 2.27 + 1/6 ∗ 2.86 = 2.23
4	1.63	2.11	2.87	1/6 ∗ 1.63 + 4/6 ∗ 2.11 + 1/6 ∗ 2.87 = 2.16
5	1.55	2.45	3.10	1/6 ∗ 1.55 + 4/6 ∗ 2.45 + 1/6 ∗ 3.10 = 2.41
j	${f^2_{j,\beta}}^p$	${f^2_{j,\beta}}^m$	${f^2_{j,\beta}}^o$	$w_1 \cdot {f^2_{j,\beta}}^p + w_2 \cdot {f^2_{j,\beta}}^m + w_3 \cdot {f^2_{j,\beta}}^o = p^2_j$
1	1.53	2.29	3.33	1/6 ∗ 1.53 + 4/6 ∗ 2.29 + 1/6 ∗ 3.33 = 2.34
2	1.53	2.64	3.47	1/6 ∗ 1.53 + 4/6 ∗ 2.64 + 1/6 ∗ 3.47 = 2.59
3	1.47	2.56	3.09	1/6 ∗ 1.47 + 4/6 ∗ 2.56 + 1/6 ∗ 3.45 = 2.47
4	1.79	2.46	3.45	1/6 ∗ 1.79 + 4/6 ∗ 2.46 + 1/6 ∗ 3.45 = 2.51
5	1.56	2.27	3.07	1/6 ∗ 1.56 + 4/6 ∗ 2.27 + 1/6 ∗ 3.07 = 2.29
j	$s^p_{j,\beta}$	$s^m_{j,\beta}$	$s^o_{j,\beta}$	$w_1 \cdot s^p_{j,\beta} + w_2 \cdot s^m_{j,\beta} + w_3 \cdot s^o_{j,\beta} = p^s_j$
1	1.95	2.51	3.47	1/6 ∗ 1.95 + 4/6 ∗ 2.51 + 1/6 ∗ 3.47 = 2.58
2	1.71	2.68	3.29	1/6 ∗ 1.71 + 4/6 ∗ 2.68 + 1/6 ∗ 3.29 = 2.62
3	2.06	2.72	3.31	1/6 ∗ 2.06 + 4/6 ∗ 2.72 + 1/6 ∗ 3.31 = 2.71
4	1.98	2.84	3.61	1/6 ∗ 1.98 + 4/6 ∗ 2.84 + 1/6 ∗ 3.61 = 2.83
5	1.80	2.63	3.23	1/6 ∗ 1.80 + 4/6 ∗ 2.63 + 1/6 ∗ 3.23 = 2.59

Table 3 Defuzzified processing times

j	p^1_j	p^2_j	p^s_j
1	2.44	2.34	2.58
2	2.35	2.59	2.62
3	2.23	2.47	2.71
4	2.16	2.51	2.83
5	2.41	2.29	2.59

After calculating defuzzified processing times, the algorithm in Steps 2a–2e, uses Johnson's (1954) algorithm without considering the flexible operations as below:

$$M_1 = \left[j \mid p_j^1 < p_j^2 \right] = [2, 3, 4]$$

$$M_2 = \left[j \mid p_j^1 \geq p_j^2 \right] = [1, 5]$$

$$M_1^{sorted} = [4, 3, 2]$$

$$M_2^{sorted} = [1, 5]$$

Then, the approximated schedule is found to be $S = [4, 3, 2, 1, 5]$. Defuzzified processing times of each job is updated according to sequence number, using schedule S, in Step 3 of heuristic algorithm. They are presented in Table 4 where *sn* represents sequence number.

Table 4 Defuzzified processing times of each job according to sequence number

sn	p_{sn}^1	p_{sn}^2	p_{sn}^s
1	2.16	2.51	2.83
2	2.23	2.47	2.71
3	2.35	2.59	2.62
4	2.44	2.34	2.58
5	2.41	2.29	2.59

Then, the algorithm assigns the flexible operations in Step 4 as below:

$$S_{avr} = \sum_{j=1}^{n} \frac{p_j^s}{n} = \frac{2.58 + 2.62 + 2.71 + 2.83 + 2.59}{n} = 2.67$$

Average flexible operation processing time s_{avr} is calculated to use in the assignment of flexible operations to the machines. r is the total number of parts for which the flexible operation is assigned to the first machine as previously explained.

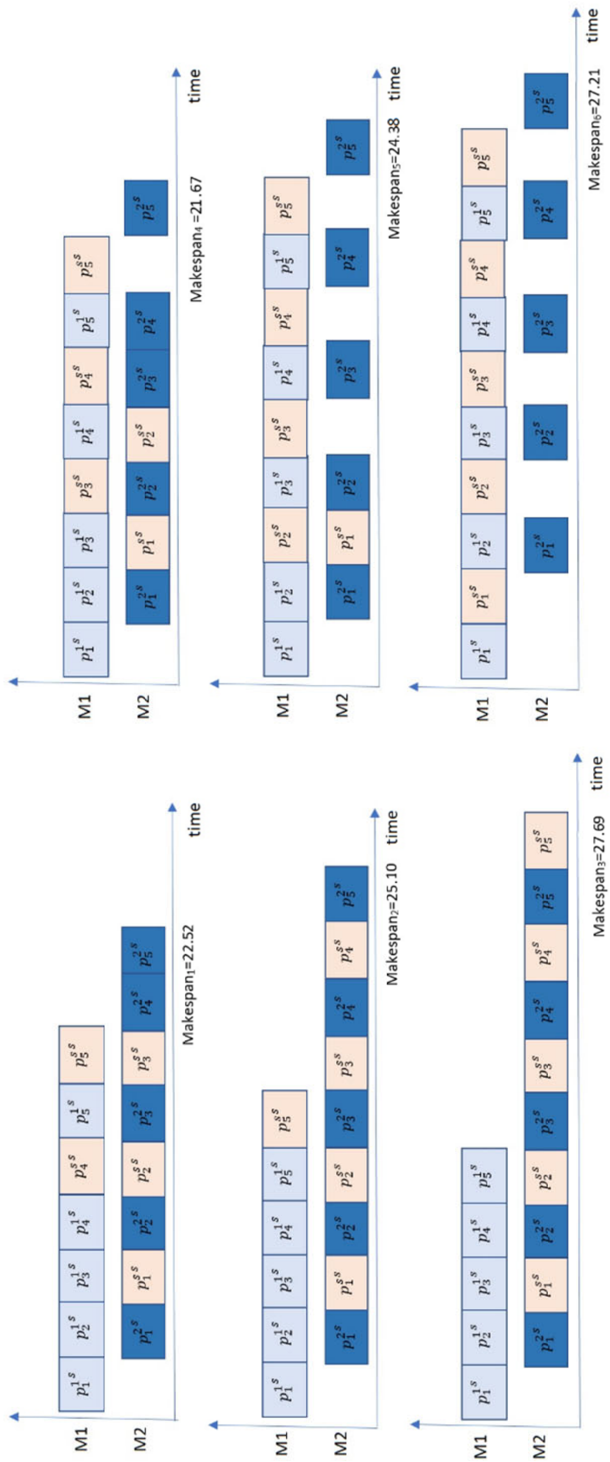
$$r = \frac{\left(\sum_{sn=1}^{n-1} p_{sn}^{2\ s} - \sum_{sn=2}^{n} p_{sn}^{1\ s} + n \cdot s_{avr} \right)}{2 s_{avr}}$$

$$= \frac{(2.51 + 2.47 + 2.59 + 2.34) - (2.23 + 2.35 + 2.44 + 2.41)(5 \times 2.67)}{2 \times 2.67} = 2.6$$

Table 5 Calculation of r_i values in Steps 5–10 of the heuristic algorithm

i	r_i	$Makespan_i$
1	$r_1 = \lfloor 2.6 \rfloor = 2$	22.52
2	$r_2 = r_1 - 1 = 1$	25.10
3	$r_3 = r_2 - 1 = 0$	27.69
4	$r_4 = \lceil 2.6 \rceil = 3$	21.67
5	$r_5 = r_4 + 1 = 4$	24.38
6	$r_6 = r_5 + 1 = 5$	27.21

Figure 2 Schedule of cases in Steps 5–10 of heuristic algorithm (see online version for colours)



Afterwards, the heuristic algorithm checks different assignments of flexible operations by changing *r* value to find a better makespan value, which is summarised in Table 5. In Figure 2, schedules of the trials are depicted.

The makespan is reported to be minimum of the calculated makespan values as below:

$$\begin{aligned} &Makespan \\ &= \min\left\{Makespan_1, Makespan_2, Makespan_3, Makespan_4, Makespan_5, Makespan_6\right\} \\ &= \min\{22.52, 25.10, 27.69, 21.67, 24.38, 27.21\} = 21.67 \end{aligned}$$

Best solution is found to be $Makespan_4$ which is 21.67. Assignment of flexible operations can be seen in detail in Figure 2.

4.2 Sample replication analysis

Table 6 shows five sample replication analysis for $n = 30$ which includes makespan values of mathematical model and heuristic algorithm, and % deviations. While some of the deviations are positive, the others are negative in Table 6. A positive deviation means that mathematical model solved by GAMS finds a better solution than proposed heuristic algorithm. On the other hand, a negative deviation means proposed heuristic algorithm can find a better solution than mathematical model. This is possible because GAMS does not always obtain the global optimum. As can be seen in Table 6, % deviations are very small which demonstrates heuristic algorithm performs sufficiently good. Moreover, in some cases, heuristic algorithm gives better results.

Table 6 Makespan and percent deviations of five replications for $n = 30$

Makespan		*Deviation*
Model	*Algorithm*	
111.53	111.62	0.081%
112.26	112.67	0.365%
112.20	112.09	–0.098%
111.46	111.60	0.126%
112.66	112.74	0.071%

4.3 Percent deviations

Table 7 shows minimum, average, and maximum deviations of heuristic algorithm from mathematical model for $n = 20, 30, 40, 50, 60$. Average deviations indicate that heuristic algorithm generates good quality solutions. The maximum deviation in Table 7 is 1.028%, which is also acceptable, it appears when number of jobs is 20. The minimum deviation in Table 7, –0.156%, appears again at 20 job, which heuristic algorithm improves mathematical model. The variance (Ω^2), standard deviation (Ω), coefficient of variation (CV) and 95% confidence interval (95% CI) values are also presented in Table 7.

Table 7 Statistical analysis of percent deviations

Number of jobs	*Deviations of replications*						
	Minimum	*Average*	*Maximum*	Ω^2	Ω	*CV*	*95% CI*
20	–0.156%	0.304%	1.028%	0.225	0.474	1.560	–0.112, 0.712
30	–0.098%	0.109%	0.365%	0.028	0.167	1.528	–0.039, 0.259
40	0.441%	0.523%	0.612%	0.005	0.068	0.131	0.459, 0.581
50	0.043%	0.381%	0.577%	0.047	0.216	0.567	0.187, 0.573
60	–0.081%	0.095%	0.310%	0.030	0.173	1.819	–0.049, 0.249

4.4 CPU times

Table 8 shows makespan values, deviations, and CPU times (in seconds) of five replications of heuristic algorithm and mathematical model for n = 40. Due to CPU restrictions, even for 40 jobs, it is not possible to run the mathematical model till the end by CPLEX. Therefore, CPLEX was run with a time limit of 1,000 seconds. CPLEX solver did not stop before the time limit exceeded for some replications, and reported a value when time limit is reached, and these CPU times is written in Table 8 as 1,000*. As Table 4 visualises, CPU times of heuristic algorithm are much smaller than GAMS. At two replications, CPLEX solver of GAMS did not stop before the limit of 1,000 second. When deviations of makespan values are considered, algorithm presents good quality solutions in less then one second.

Table 8 Makespan values, percent deviations and CPU times of five replications for n = 40

Makespan		*Deviation*	*CPU (seconds)*	
Model	*Algorithm*		*Model*	*Algorithm*
148.66	149.57	0.612%	29.25	0.624
150.46	151.31	0.565%	1,000*	0.858
149.60	150.31	0.475%	652.4	0.718
149.80	150.58	0.521%	0.124	0.905
149.56	150.22	0.441%	1,000*	0.842

Note: *Stopped due to the time limit

Table 9 Statistical analysis of CPU times of mathematical model

Number of jobs	*CPU (seconds)*					
	Minimum	*Average*	*Maximum*	Ω^2	Ω	*CV*
20	0.25	1.93	2.98	1.367	1.169	0.61
30	0.03	304.85	1,000*	202,473	449.97	1.48
40	0.12	536.35	1,000*	247,029	497.02	0.93
50	0.12	600.12	1,000*	299,811	547.55	0.91
60	0.16	800.03	1,000*	199,934	447.14	0.56

Note: *Stopped due to the time limit

Statistical analysis of CPU times of mathematical model and heuristic algorithm for n = 20, 30, 40, 50, 60 are presented in Tables 9 and 10. As can be seen in Tables 9 and 10, CPU time of mathematical model solved by CPLEX solver of GAMS increases sharply with the number job increase, but CPU time of heuristic algorithm remains lower than just one second. According to the results, the proposed algorithm can generate a relatively good solution in a very small CPU time which is much smaller than CPLEX. Even if the number of jobs increases, CPU time remains very small for the algorithm.

Table 10 Statistical analysis of CPU times of heuristic algorithm

Number of jobs	*CPU (seconds)*					
	Minimum	*Average*	*Maximum*	Ω^2	Ω	*CV*
20	0.76	0.86	0.94	0.005	0.073	0.08
30	0.82	0.89	0.96	0.003	0.054	0.06
40	0.62	0.79	0.91	0.013	0.115	0.15
50	0.66	0.73	0.81	0.004	0.060	0.08
60	0.59	0.76	0.94	0.016	0.125	0.17

5 Conclusions

In this paper, scheduling of n-jobs, each having three operations with fuzzy processing times, is studied in two-machine flowshop environment to minimise makespan. One of the operations was suitable to be processed on both machines, named as flexible operation, and it was assigned to minimise makespan. Processing times are assumed to be triangular fuzzy numbers. Firstly, a possibilistic mathematical model was formulated and defuzzified using weighted average method to get a crisp mathematical model. The model was solved using CPLEX solver of GAMS. Since the mathematical programming formulations may not be efficient in terms of CPU time for large instances, a heuristic algorithm was proposed. The algorithm was coded and solved in MATLAB and verified that heuristic algorithm generates good quality solutions in less than one second which is not possible for mathematical model for large number of jobs. As future research, the number of machines and/or flexible operations can be increased to extend this work.

References

Adiri, I. and Frostig, E. (1984) 'A stochastic permutation-flowshop scheduling problem minimizing in distribution the schedule length', *Operations Research Letters*, Vol. 3, No. 2, pp.101–103.

Al-Faruk, A., Ahmed, N., Haque, M.A. and Mahmud, S.A. (2011) 'Design of a knowledge-based manufacturing flow shop sequencing system using fuzzy logic', *International Journal of Engineering and Applied Sciences*, Vol. 3, No. 3, pp.1–14.

Ambika, G. and Uthra, G. (2014) 'Branch and bound technique in flow shop scheduling using fuzzy processing times', *Annals of Pure and Applied Mathematics*, Vol. 8, No. 2, pp.37–42.

Baker, K.R. and Trietsch, D. (2011) 'Three heuristic procedures for the stochastic two-machine flow shop problem', *Journal of Scheduling*, Vol. 14, No. 5, pp.445–454.

Behmanesha, R., Zandiehb, M. and Hadji Molanaa, S.M. (2019) 'The surgical case scheduling problem with fuzzy duration time: an ant system algorithm', *Scientia Iranica E*, Vol. 26, No. 3, pp.1824–1841.

Chanas, S. and Kasperski, A. (2003) 'On two single machine scheduling problems with fuzzy processing times and fuzzy due dates', *European Journal of Operational Research*, Vol. 147, No. 2, pp.281–296.

Crama, Y. and Gultekin, H. (2010) 'Throughput optimization in two-machine flowshops with flexible operations', *Journal of Scheduling*, Vol. 13, No. 3, pp.227–243.

Cunningham, A.A. and Dutta, S.K. (1973) 'Scheduling jobs with exponentially distributed processing times on two machines of a flow shop', *Naval Research Logistics Quarterly*, Vol. 20, No. 1, pp.69–81.

Daniels, R.L. and Mazzola, J.B. (1994) 'Flow-shop scheduling with resource flexibility', *Operations Research*, Vol. 42, No. 3, pp.390–575.

Daniels, R.L., Mazzola, J.B. and Shi, D. (2004) 'Flow shop scheduling with partial resource flexibility', *Management Science*, Vol. 50, No. 5, pp.658–669.

Dobson, G. and Karmarkar, U.S. (1989) 'Simultaneous resource scheduling to minimize weighted flow times', *Operations Research*, Vol. 37, No. 4, pp.592–600.

Elmaghraby, S.E. and Thoney, K.A. (1999) 'The two-machine stochastic flowshop problem with arbitrary processing time distributions', *IIE Transactions*, Vol. 31, No. 5, pp.467–477, Institute of Industrial Engineers.

Geyik, F. and Elibal, K. (2017) 'A linguistic approach to non-identical parallel processor scheduling with fuzzy processing times', *Applied Soft Computing*, Vol. 55, No. C, pp.63–71.

Gultekin, H. (2012) 'Scheduling in flowshops with flexible operations: throughput optimization and benefits of flexibility', *International Journal of Production Economics*, Vol. 13, No. 140, pp.900–911.

Gupta, D., Sharma, S. and Aggarwal, S. (2012) 'Specially structured flow shop scheduling with fuzzy processing time to minimize the rental cost', *Advances in Applied Science Research*, Vol. 3, No. 4, pp.1986–1991.

Gupta, J.N.D., Koulamas, C.P., Kyparisis, G.J., Potts, C.N. and Strusevich, V.A. (2004) 'Scheduling three-operation jobs in a two-machine flow shop to minimize makespan', *Annals of Operations Research*, Vol. 129, No. 1, pp.171–185.

Jia, Z., Yan, J., Leung, J.Y.T., Li, K. and Chen, H. (2019) 'Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities', *Computer Science, Business*, Vol. 75, pp.548–561.

Johnson, S.M. (1954) 'Optimal two- and three-stage production schedules with setup times included', *Naval Research Logistics Quarterly*, Vol. 1, No. 1, pp.61–68, John Wiley & Sons.

Kalczynski, P.J. and Kamburowski, J. (2004) 'Generalization of Johnson's and Talwar's scheduling rules in two-machine stochastic flow shops', *Journal of the Operational Research Society*, Vol. 55, No. 12, pp.1358–1362.

Kalczynski, P.J. and Kamburowski, J. (2006) 'A heuristic for minimizing the expected makespan in two-machine flow shops with consistent coefficients of variation', *European Journal of Operational Research*, Vol. 169, No. 3, pp.742–750.

Kenneth, R.B. and Dominik, A. (2012) 'Heuristic solution methods for the stochastic flow shop problem', *European Journal of Operational Research*, Vol. 216, No. 1, pp.172–177.

Khorasanian, D. and Moslehi, G. (2017) 'Two-machine flow shop scheduling problem with blocking, multi-task flexibility of the first machine, and pre-emption', *Computers and Operations Research*, Vol. 79, No. C, pp.94–108.

Ku, P.S. and Niu, S.C. (1986) 'On Johnson's two-machine flow shop with random processing times', *Operations Research*, Vol. 34, No. 1, pp.130–136.

Kumar, H., Kumar, P. and Sharma, M. (2018) 'An effective new hybrid optimization algorithm for solving flow shop scheduling problems', *Malaya Journal of Matematik*, Vol. S, No. 1, pp.107–113.

Lai, Y.J. and Hwang, C.L. (1992) *Fuzzy Mathematical Programming: Methods and Applications*, Springer-Verlag, Berlin.

Li, J., Xu, D. and Li, H. (2018) 'Single machine due window assignment scheduling problem with precedence constraints and fuzzy processing times', *Journal of Intelligent & Fuzzy Systems*, Vol. 34, No. 6, pp.4301–4314.

Li, K., Chen, J., Fu, H., Jia, Z. and Fu, W. (2019) 'Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint', *Appl. Soft Comput.*, Vol. 82, No. C, p.105585.

McCahon, C.S. and Lee, E.S. (1990) *Job Sequencing with Fuzzy Processing Times*, Vol. 19, No. 7, pp.31–41, Springer Verlag.

Najari, F., Alaghebandha, M., Mohammadi, M. and Sobhanallahi, M.A. (2018) 'Fuzzy multi-objective permutation flow shop scheduling problem with fuzzy processing times under learning and aging effects', *Journal of Quality Engineering and Production Optimization*, Vol. 3, No. 2, pp.77–98.

Noori-Darvisha, S., Mahdavib, I. and Mahdavi-Amiric, N. (2012) 'A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates', *Applied Soft Computing Journal*, Vol. 12, No. 4, pp.1399–1416.

Petrovic, S. and Song, X. (2006) 'A new approach to two-machine flow shop problem with uncertain processing times', *Optimization and Engineering*, Vol. 15, No. 7, pp.329–342.

Portougal, V. and Trietsch, D. (2006) 'Johnson's problem with stochastic processing times and optimal service level', *European Journal of Operational Research*, Vol. 169, No. 3, pp.751–760.

Razmia, J., Moghaddam, R.T. and Saffari, M. (2009) 'A mathematical model for a flow shop scheduling problem with fuzzy processing times', *Journal of Optimization in Industrial Engineering*, Vol. 3, No. 1, pp.39–44.

Rostami, M., Pilerood, A.E. and Mazdeh, M.M. (2015) 'Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment', *International Journal of Production Research*, Vol. 85, No. 1, pp.206–215.

Sakawa, M. and Kubota, R. (2000) 'Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms', *International Conference on Natural Computation, ICNC 2005: Advances in Natural Computation*, Vol. 120, pp.393–407.

Sethi, S., Yan, H., Zhang, Q. and Zhou, X.Y. (1993) 'Feedback production planning in a stochastic two-machine flowshop: asymptotic analysis and computational results', *International Journal of Production Economics*, Vols. 30–31, No. C, pp.79–93.

Sodhi, M.S., Agnetis, A. and Askin, R.G. (1994) 'Tool addition strategies for flexible manufacturing systems', *International Journal of Flexible Manufacturing Systems*, Vol. 6, No. 4, pp.287–310.

Soroush, H.M. and Allahverdi, A. (2005) 'Stochastic two-machine flowshop scheduling problem with total completion time criterion', *International Journal of Industrial Engineering: Theory Applications and Practice*, Vol. 12, No. 2, pp.159–171.

Stecke, K.E. (1983) 'Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems', *Management Science*, Vol. 29, No. 3, pp.273–288.

Talwar, P.P. (1967) 'A note on sequencing problems with uncertain job times', *Journal of the Operations Research Society of Japan*, Vol. 9, No. 1, pp.93–97.

Temiz, A. and Erol, S. (2004) 'Fuzzy branch and bound algorithm for flow shop scheduling', *Journal of Intelligent Manufacturing*, Vol. 15, No. 4, pp.449–454.

Toksari, M.D. and Arik, O.A. (2017) 'Single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times', *Journal of Manufacturing Systems*, Vol. 45, No. 1, pp.159–179.

Tsujimura, Y., Gen, M. and Kubota, E. (1995) 'Solving fuzzy assembly-line balancing problem with genetic algorithms', *Computer Science, Business*, Vol. 29, Nos. 1–4, pp.543–547.

Tsujimura, Y., Park, S.H., Chang, I.S. and Gen, M. (1993) 'An effective method for solving flow shop scheduling problems with fuzzy processing times', *Computers & Industrial Engineering*, Vol. 25, Nos. 1–4, pp.239–242.

Uruk, Z., Gultekin, H. and Akturk, M.S. (2013) 'Two-machine flowshop scheduling with flexible operations and controllable processing times', *Computers & Operations Research*, Vol. 40, No. 2, pp.639–653.

Wei, Q., Wu, Y., Jiang, Y. and Cheng, T.C.E. (2019) 'Two-machine hybrid flowshop scheduling with identical jobs: solution algorithms and analysis of hybrid benefits', *Journal of the Operational Research Society*, Vol. 70, No. 5, pp.817–826.