Original Research

Improved TLBO and JAYA algorithms to solve new fuzzy flexible job-shop scheduling problems

Raviteja Buddala[1*]. Siba Sankar Mahapatra[2]. Manas Ranjan Singh[3]. Bhanu Chandar Balusa[4]. Venkata Phanikrishna Balam[5]

Received: 08 July 2021 / Accepted: 06 March 2023 / Published online: 12 March 2023

* Corresponding Author, raviteja317@gmail.com

1- School of Mechanical Engineering, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India

2- Department of Mechanical Engineering, National Institute of Technology Rourkela, Rourkela, 769008, Odisha, India

3- Department of Basic Sciences and Humanities, Silicon Institute of Technology, Bhubaneswar, Odisha, India

4- School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamilnadu, India

5- School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India

Abstract

Flexible job-shop scheduling problem (FJSP) finds significant interest in the field of scheduling in dealing with complexity, solution methodology and, industrial applications. However, most of the studies on FJSP, consider the processing time of operations to be deterministic and known at priori while solving the problem. Since uncertainty is bound to occur in industries, deterministic approaches for solving FJSP may not yield good solutions. Schedules generated considering uncertainties may help the manufacturing firms to handle the uncertainties efficiently. The present work aims at solving FJSP in a realistic manner, considering uncertainty in the processing times. A modified version of optimization algorithms without tuning parameters such as teaching-learning-based optimization (TLBO) and JAYA is proposed to solve fuzzy FJSP (FFJSP) with less computational burden. Although there are enough challenging benchmark problems for deterministic FJSP problems, only limited benchmarks are available for a fuzzy variant of FJSP. The currently available FFJSP problems in the literature are small in size as compared to Brandimate data instances which are widely accepted for a deterministic variant of FJSP. Therefore, an attempt has been made in this paper to solve the instances of Kacem's and Brandimarte's by converting them into fuzzy FJSP. The present work also provides new challenging problems compared to the existing benchmark problems to study FFJSP.

Keywords - Flexible job-shop scheduling problem; JAYA; processing time's uncertainty; TLBO; triangular fuzzy numbers.

INTRODUCTION

Flexible job-shop scheduling problem (FJSP) attracts the attention of the researchers because it is one of the difficult NP-hard (non-deterministic polynomial-time hard) problems and poses difficulty in solving due to its complexity (Garey, Johnson, &

Sethi, 1976). A large number of methods have been proposed to solve FJSP and flexible flow-shop with the objective of minimization of makespan under deterministic conditions (Apornak, Raissi, & Pourhassan, 2021; Buddala & Mahapatra, 2016; Buddala, Mahapatra, & Singh, 2021; Dunke & Nickel, 2022; Gao et al., 2016; Raissi, Rooeinfar, & Ghezavati, 2019; Rooeinfar, Raissi, & Ghezavati, 2019; Singh & Mahapatra, 2016; L. Wang, Zhou, Xu, Wang, & Liu, 2012; Yuan, Xu, & Yang, 2013). In deterministic FJSP, it is assumed that processing times are known at priori and deterministic. In practice, uncertainties like uncertain processing times and machine failures are frequently encountered on a shop floor (Li, Gong, & Lu, 2022a, 2022b; Ourari, Berrandjia, Boulakhras, Boukciat, & Hentous, 2015; Subramaniam & Raheja, 2003).

In a job-shop environment, the uncertainties that arise can be categorized into two categories. They are (i) resource related and (ii) job related (Vieira, Herrmann, & Lin, 2003). Unavailability of machines or failure of tools, operator illness, loading limits and delay or shortage in the arrival of materials are known as resource related uncertainties. Unexpected cancellation of existing jobs or arrival of new jobs, changes in due dates, early arrival of jobs or delay in arrival of jobs and uncertainties in processing times are treated as job related uncertainties. If all these uncertainties are taken into account while solving a scheduling problem, the problem becomes too difficult to solve (Chaari, Chaabane, Aissani, & Trentesaux, 2014; Jin, Zhang, Wen, Sun, & Fei, 2021; K. Wang & Choi, 2012). It is to be noted that solution methodologies designed to solve deterministic scheduling problems cannot be applied to solve scheduling problems with uncertainty (He, Sun, & Liao, 2013). Probability distribution may be used to deal with the shop floor uncertainties (Bruni, Beraldi, Guerriero, & Pinto, 2011). It is found that estimating makespan distribution is intractable even when durations of operations are independent random variables (Ludwig, Möhring, & Stork, 2001). Therefore, an alternative approach to model the variations in processing times emphasizes the use of fuzzy intervals or fuzzy numbers (Dubois, Fargier, & Fortemps, 2003; Sowinski & Hapke, 2000). Adding flexibility to a job shop scheduling problem (JSP) makes the scheduling problem complex. Further, if fuzzy processing times are considered, the problem becomes too difficult to solve (Deming Lei, 2010a).

Uncertainty in processing times is one of the important uncertainties encountered on the shop floor. If vagueness in the operation times is considered, the problem is known as a fuzzy flexible job shop scheduling problem (FFJSP). Since large-scale FFJSP is difficult to solve using traditional approaches, meta-heuristics are usually used to find a near-optimal solution with less computational burden. In this work, improved versions of teaching-learning-based optimization (TLBO) and JAYA algorithms are used to solve FFJSP because these algorithms do not require any tuning parameter. Also, finding the correct tuning parameters specific to the algorithm is a cumbersome process. The proposed solution methodologies lose their diversity as the iterations proceed. Therefore, a new local search method proposed by (Buddala & Mahapatra, 2018) is embedded into the basic TLBO and JAYA to improve the efficiency of the algorithm.

LITERATURE REVIEW

Fuzzy numbers are considered for job processing times to consider uncertainty in processing times for single machine scheduling problem (Chanas & Kasperski, 2003; Kasperski, 2007). Similarly, fuzzy processing times are considered for parallel machine scheduling problem by Peng (Peng & Liu, 2004) and Balin (Balin, 2011), Ishibushi (Ishibuchi & Murata, 1998) and Celano (Celano, Costa, & Fichera, 2003) have considered fuzzy processing times for flow shop scheduling problem whereas (Palacios, González-Rodríguez, Vela, & Puente, 2014) have considered for open shop scheduling problem. In job shop scheduling, fuzzy processing times are considered by many authors (Fortemps, 1997; Gonzalez-Rodriguez, Puente, Vela, & Varela, 2008; González Rodrıguez, Vela, Puente, & Varela, 2008; Deming Lei, 2008, 2010b; Niu, Jiao, & Gu, 2008; Petrovic, Fayad, Petrovic, Burke, & Kendall, 2008; Puente Peinador, Rodríguez Vela, & González Rodríguez, 2010; Sakawa & Kubota, 2000; Tavakkoli-Moghaddam, Safaei, & Kah, 2008; Zheng & Li, 2012).

Lei (Deming Lei, 2010a) made an attempt to solve FJSP with an aim to tackle the uncertain processing times using fuzzy numbers (fuzzy processing times). Even though there were sufficient number of benchmark problems for FJSP to test the performance of an algorithm, sufficient number of benchmark problems for FFJSP are not readily available in the literature. Therefore, four test instances were proposed to conduct the experimental studies. Lei (Deming Lei, 2012) proposed additional new instance with size 15 jobs by 10 machines of 80 operations and applied co-evolutionary genetic algorithm to solve FFJSP. Wang (L. Wang, Zhou, Xu, & Liu, 2013) have proposed a hybrid artificial bee colony algorithm (HABC) to solve four instances proposed by Lei (Deming Lei, 2010a). Wang et. al. (S. Wang, Wang, Xu, & Liu, 2013) have proposed an estimation of distribution algorithm (EDA) to solve all the five instances proposed by Lei (Deming Lei, 2010a, 2012). Xu et al. (Xu, Wang, Wang, & Liu, 2015) have proposed a teaching learning-based optimization algorithm to solve all the five instances proposed by Lei (Deming Lei, 2010a, 2012). Wang et al. (X. Wang, Gao, Zhang, & Li, 2012) have proposed a new set of four FFJSP instances to test the efficiency of multi-objective genetic algorithm (MOGA). Wang et al. (X. Wang, Li, & Zhang, 2013) have proposed an improved multi-objective genetic algorithm (IMOGA) and compared its efficiency with NSGA-II. It is observed that IMOGA is found to be superior to NSGA-II in solving FFJSP. Wang et al. (C. Wang, Tian, Ji, & Wang, 2017) have proposed a memetic algorithm (MA) to solve multi-objective FFJSP. It is indicated that the proposed MA outperforms MOGA and IMOGA. Since the test instances for FFJSP are limited in the literature, Palacios et al. (Palacios, González, Vela, González-Rodríguez, & Puente, 2015) and (Palacios, Puente, Vela, & González-Rodríguez, 2016) emphasized on necessity of challenging benchmark problems for fuzzy scheduling problems. In case of deterministic FJSP, Kacem's instances (Kacem, Hammadi, & Borne, 2002a, 2002b) and Brandimarte's instances (Brandimarte, 1993) are most widely solved challenging benchmark problems. Brandimarte's instances have a large number of operations per job (minimum 5 operations per job to maximum 12 operations per job for different instances) as compared to Lei's instances and Wang's (X. Wang et al., 2012) instances (only three operations per job). In FJSP, Brandimarte's instances are challenging problems but these are deterministic problems. If these Brandimarte's instances can be converted into fuzzy instances, these problems serve as better challenging problems in FFJSP.

In order to generate fuzzy processing times (fP_{ijk}) from the crisp processing times (P_{ijk}) of a deterministic scheduling problem, different researchers have used different techniques in different scheduling problems. Here P_{ijk} indicates processing time of j^{th} operation for job i on machine k. In general, a triangular fuzzy processing time can be represented as $fP_{ijk} = (P_{ijk1}, P_{ijk2}, P_{ijk3})$. The most likely time value P_{ijk2} of the fuzzy processing time is same as crisp processing time P_{ijk}, P_{ijk1} is the minimum processing time and P_{ijk3} is the maximum processing time for an operation. The values of P_{ijk1} and P_{ijk3} are obtained randomly from $[\delta_1 P_{ijk}, P_{ijk}]$ and $[P_{ijk}, \delta_2 P_{ijk}]$ respectively. Lin (Lin, 2002) has considered δ_1=0.5 and δ_2=1.5. Rodriguez et al. (Gonzalez-Rodriguez, Vela, & Puente, 2007) have considered δ_1=0.7 and δ_2=1.3. The fuzzy conversion method was first proposed by Fortemps (Fortemps, 1997). The same technique is used recently by Palacios et al. (Palacios et al., 2016) to propose some new standard benchmark problems for fuzzy JSP. Petrovic et al. (Petrovic et al., 2008) have considered δ_1=0.9 and δ_2=1.1. Ghrayeb (Ghrayeb, 2003) and Puente et al. (Puente Peinador et al., 2010) have considered δ_1=0.85 and δ_2=1.3. Lei (Demion Lei, 2011) has considered δ_1 in the range $[0.85P_{ijk}, 0.95P_{ijk}]$ and δ_2 in the range $[1.1P_{ijk}, 1.9P_{ijk}]$. Rodriguez et al. (González-Rodríguez, Vela, & Puente, 2006) have considered δ_1 in the range $[(1/3Pijk), P_{ijk}]$ and δ_2 in the range $[P_{ijk}, (4/3Pijk)]$.

Out of many types of fuzzy conversion methods, it has been concluded by Palacios et al. (Palacios et al., 2016) that the fuzzy conversion method proposed by Rodriguez et al. (Gonzalez-Rodriguez et al., 2007) is superior in converting the deterministic processing times into fuzzy processing times. Palacios et al. (Palacios et al., 2016) have not only applied it for converting deterministic job shop scheduling (JSP) problems into fuzzy JSP but also proposed new standard fuzzy JSP benchmark problems to evaluate the performance of different algorithms. As FJSP and FFJSP are the extensions of JSP, the fuzzy conversion proposed by Rodriguez et al. (Gonzalez-Rodriguez et al., 2007) can be conveniently applied with δ_1=0.7 and δ_2=1.3 to convert the deterministic FJSP into fuzzy FJSP for proposing new fuzzy FJSP benchmark problems using the deterministic FJSP problems.

From the literature of scheduling problems, it is observed that performance of a method is not solely judged based on makespan criterion but also in terms of relative error (RE) from the lower bound (LB). Finding the lower bounds for unsymmetrical fuzzy numbered processing times is very difficult. However, it is pointed out that symmetrical fuzzy instances have same lower bounds as that of deterministic crisp scheduling problem (Fortemps, 1997). Based on this principle, Palacios et al. (Palacios et al., 2016) have generated symmetric triangular fuzzy processing times to convert the deterministic job shop scheduling problems into fuzzy job shop scheduling problems so that the lower bounds of deterministic job shop scheduling problems (JSP) can be readily used for the fuzzy job-shop scheduling problems for comparing the efficacy of solution strategies. As FJSP and FFJSP are the extensions of JSP, the above principle is incorporated in the present work to convert the deterministic FJSP into fuzzy FJSP (FFJSP) so that the lower bounds of deterministic FJSP can be readily used for FFJSP. The formula to find relative error (RE) with respect to a LB is given in the equation 1.

$$RE = \frac{E(C_{max}) - LB}{LB} \quad (1)$$

where $E(C_{max})$ is the expected most likely makespan of a FFJSP.

FUZZY FJSP (FFJSP)

I. FJSP with fuzzy durations

A FFJSP is explained as follows: There a set of m machines on which n jobs need to be processed. All jobs are available at time zero to process them on mahcines. Each job i has Oi sequence of operations (j=1, 2, 3, …., Oi). For any operation Oij, there exists a machine *Mijk* such that *Mijk* ϵ *m*. The fuzzy processing time fP_{ijk} of any operation *Oij* on a given machine *k* is represented as a triangular fuzzy number (TFN) $fP_{ijk} = (P_{ijk1}, P_{ijk2}, P_{ijk3})$ where, P_{ijk1} is the minimum processing time and P_{ijk3} is the maximum processing time for an operation. P_{ijk2} is the most likely time value. Similarly, the completion time of any operation *Oij* on a given machine *k* is also represented as a TFN $fC_{ijk} = (C_{ijk1}, C_{ijk2}, C_{ijk3})$ where, C_{ijk1} is the minimum completion time, P_{ijk2} is the most likely completion time and P_{ijk3} is the maximum completion time for an operation. The basic assumptions of a deterministic FJSP problem are also applicable to FFJSP. All machines are independent from each other. There are no machine breakdowns during the process and all jobs are independent from each other. The objective is to find a schedule with minimum makespan value (equation 2) with the uncertain fuzzy durations.

$$C_{max} = \max(\min C_i)\ \forall\ i \epsilon\ n \tag{2}$$

where C_i is the completion time of job i and C_{max} is the maximum completion time (makespan).

II. Operations on fuzzy durations

In a fuzzy shop floor environment, few operations (BORTOLAN & DEGANI, 1993) on fuzzy numbers are necessary to generate a schedule. These fuzzy operations consist of addition operation to determine the fuzzy completion time, maximum operation to calculate the fuzzy beginning time of an operation and rank method to find the makespan or maximum completion time (Xu et al., 2015). For example, let us assume that there are two triangular fuzzy numbers (TFN) $B = (b_1, b_2, b_3)$ and $C = (c_1, c_2, c_3)$. Then, the addition operation between the two TFN's *B* and *C* is given by $B+C = (b_1 + c_1, b_2 + c_2, b_3 + c_3)$. In the present work, the criterion proposed in Lei (Deming Lei, 2012) is used to find the maximum of a TFN. That is if $B>C$, $B \vee C = B$; else $B \vee C = C$. From literature review (Deming Lei, 2010a, 2010b; Xu et al., 2015), it is found that the rank method proposed by Sakawa and Kubota (Sakawa & Kubota, 2000) is most widely used for its simplicity and efficiency when compared to Sakawa and Mori (Sakawa & Mori, 1999). It is explained as follows.

Criterion 1: The first criterion to rank the greatest number among two TFN's is given by $U_1 (B) = (b_1 + b_2 + b_3)/4$.
Criterion 2: If $U1$ value of the two TFN's is same, then $U_2 (B) = b_2$ is used as the second criterion to decide the greatest number.
Criterion 3: If U_1 and U_2 values of the two TFN's is same, then $U_3 (B) = b_3 - b_1$ is chosen as the third criterion to decide the greatest number.

PROBLEM MAPPING MECHANISM

TABLE 1
EXAMPLE PROBLEM

Job	Operation	Machine 1	Machine 2	Machine 3
1	O_{11}	1.4,2,2.6	0.7,1,1.3	2.1,3,3.9
	O_{12}	2.8,4,5.2	4.2,6,7.8	2.1,3,3.9
2	O_{21}	0.7,1,1.3	0.7,1,1.3	1.4,2,2.6
	O_{22}	1.4,2,2.6	0.7,1,1.3	1.4,2,2.6
	O_{23}	3.5,5,6.5	2.8,4,5.2	4.2,6,7.8
3	O_{31}	0.7,1,1.3	2.1,3,3.9	0.7,1,1.3
	O_{32}	1.4,2,2.6	2.1,3,3.9	0.7,1,1.3

In this work, real number encoding system proposed by Niu et al.(Niu et al., 2008) is used to solve fuzzy flexible job-shop scheduling problem. A real number consists of two parts. They are the integer part and the fractional part. In this encoding system, integer part is used to assign an operation to a machine whereas to sequence the operations allotted to a machine the fractional part is used. A brief explanation of the same is done using the example problem given in the table 1. In table 1, it is clear that the processing times are symmetric triangular fuzzy numbers. Here, the most likely time is the deterministic processing time of a general 'deterministic FJSP' problem. Based on this deterministic processing times, a priority order matrix is obtained with the increase of processing times for an operation as shown in the table 2.

TABLE 2
PRIORITY MATRIX

Job	Operation	Priority 1	Priority 2	Priority 3
1	O_{11}	2	1	3
	O_{12}	3	1	2
2	O_{21}	1	2	3
	O_{22}	2	1	3
	O_{23}	2	1	3
3	O_{31}	1	3	2
	O_{32}	3	1	2

Using this priority order matrix, the integer part is used to allot the machine. If there is a tie in processing times, the machine with lower index number is given priority. For example, the operation O_{21} has same processing times on machine 1 and machine 2. In this case, machine 1 is given priority. A brief explanation of values representation (student's in TLBO / population in JAYA) is given in the table 3. The values in table 3 also give the initial best solution of the population. Initially, values of the population are generated at random. The values are positive real numbers where the maximum value ranges up to total number of available machines (tam) while the minimum value is 1.

TABLE 3
STOCHASTIC VALUE REPRESENTATION

Operation	O_{11}	O_{12}	O_{21}	O_{22}	O_{23}	O_{31}	O_{32}
Value	1.4216	1.4256	1.5990	1.7758	1.8219	1.5584	1.5634
Priority	1	1	1	1	1	1	1
Machine allotment	2	3	1	2	2	1	3

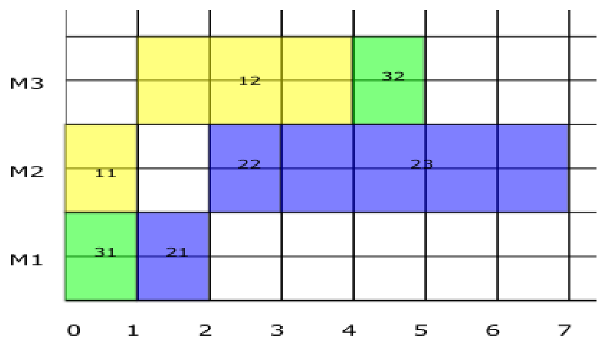

FIGURE 1A
INITIAL GANTT CHART SCHEDULE FROM DATA IN TABLE 3 WITHOUT FUZZY DURATIONS

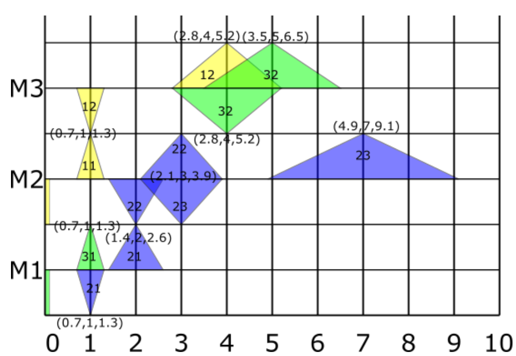

FIGURE 1B
INITIAL FUZZY GANTT CHART SCHEDULE FROM DATA IN TABLE 3

Figures 1a and 1b show the initial Gantt chart of the schedule obtained from the data in table 3. Figure 1a is without the consideration of fuzzy durations while the figure 1b is under the consideration of fuzzy durations. The double digit number inside the colored boxes (rectangular box in figure 1a and triangular box in figure 1b) represents the operation numbers of jobs. Example ‘21’ means first operation of job 2 (O_{21}).

In the recent years, several meta-heuristic techniques have been proposed to tackle several real world complex challenging problems. But all the meta-heuristics except TLBO and JAYA algorithms have algorithm specific tuning parameters. The special feature of TLBO and JAYA algorithms alleviate the drawback of time consuming process of tuning algorithmic parameters. Moreover, these algorithms are simple and easy to apply. Therefore, in the present research work, TLBO and JAYA algorithms are chosen for the study.

TEACHING-LEARNING-BASED OPTIMIZATION ALGORITHM

Teaching-learning-based optimization (TLBO) is a nature inspired meta-heuristic technique. It is proposed on the basis of the general teaching-learning process that occurs in day to day life. A teacher strives to bring his/her students’ knowledge to his/her level. The increased knowledge of the students can be measured in terms of the grades obtained by them in the exams. The meta-heuristic technique is proposed by Rao et al. (R. V. Rao, Savsani, & Vakharia, 2011) explain the two ways of learning process. They are (i) teacher phase (i.e. via teacher) and (ii) student phase (i.e.; discussing with the fellow co-students). At any iteration in the proposed algorithm, number of students (or learners) present in the class represent the population and the student with best knowledge is considered as the teacher. Implementation of TLBO is explained as follows

I. Teacher phase

Even though, a teacher wishes to bring his/her students’ knowledge to his/her level, the increased knowledge of the students depends on the capability of the students to learn. Therefore, all students cannot gain full knowledge in one go. But, during this teaching learning process, all students learn something from their teacher. Therefore, the average knowledge of the class increases. Let S_{mean} indicate the average knowledge of the students in a class and $S_{teacher}$ indicate the teacher of the class. Then the improved student’s knowledge at any iteration is given by the equation 3 as follows

$$S_{new\ i} = S_{old\ i} + r \times (S_{teacher} - (T_f \times S_{mean})) \qquad (3)$$

where ‘T_f’ is the teaching factor. Its value is randomly set to 1 or 2 and ‘r’ is a uniformly distributed random number between (0,1). Here, ‘T_f’ is a parameter of TLBO. But tuning of this ‘T_f’ is not required. This is the main advantage in this algorithm. $S_{new\ i}$ and $S_{old\ i}$ are the new and old knowledge values of the student i. $S_{new\ i}$ is accepted if it provides an improved value.

II. Student phase

As all students cannot gain full knowledge in one go. They discuss among themselves after a class is taught by the teacher. During this process again students’ knowledge increases. Let S_a and S_b denote two students who discuss among themselves, a≠b. Then the new improved knowledge of the learner student from the teaching student is given by the equations 4 and 5 as follows

$$S_{new\ a} = S_{old\ a} + r \times (S_a - S_b) \text{ if } F(S_a) <= F(S_b) \qquad (4)$$

$$S_{new\ a} = S_{old\ a} + r \times (S_b - S_a) \text{ if } F(S_b) < F(S_a) \qquad (5)$$

where ‘r’ is a uniformly distributed random number between (0,1), $S_{old\ a}$ is the old knowledge of the student ‘a’ and $S_{new\ a}$ is the new knowledge of the student ‘a’. $S_{new\ a}$ is accepted if it provides an improved value.

JAYA ALGORITHM

Rao (R. Rao, 2016) proposed Jaya algorithm very recently. In Sanskrit language, JAYA means victory. Therefore, the algorithm is named JAYA. The design of this algorithm is based on the strategy that any solution of a given population tends to escape from the worst solution and also tries to approach towards the best solution. In order to get optimal solution, Jaya do not possess any algorithm specific tuning parameters. Moreover, it contains only one equation. The specialty of this algorithm is its simplicity and ease to understand. This makes the algorithm very easy to apply. Therefore, it is clearly different from any other algorithms in a unique manner. The mathematical formulation of JAYA is explained as follows. Let *f(x)* denote the objective function that should be optimized. Out of the total population, let S_{worst} and S_{best} denote the worst and best solutions. Then the equation to modify a solution is as follows

$$S_{new\ i} = S_{old\ i} + r_1 \times (S_{best} - |S_i|) - r_2 \times (S_{worst} - |S_i|) \qquad (6)$$

where r_1 and r_2 are the two uniformly distributed random numbers between (0,1). In equation 6, the term "– r_2 * (S_{worst} - | S_i |)" represents the inclination of the solution to escape from the worst solution and the term "r_1 * (S_{best} - | S_i |)" represents the inclination of solution to approach towards the best solution. The new solution is accepted if it provides an improved value.

LOCAL SEARCH

Although both the algorithms appear to be elegant from computational point of view, they lose diversity after few iterations and may get trapped at the local optima. In order to improve the efficiency of algorithms, local search techniques have been used by the researchers in the past (Buddala & Mahapatra, 2016; Singh & Mahapatra, 2016; Xu et al., 2015). Therefore, to increase the efficiency of TLBO and JAYA algorithms to reach the near optimal solution, the current meta-heuristics also requires some improvement to the basic algorithm. To enhance the solution quality of FFJSP, a new local search proposed by Buddala and Mahapatra (Buddala & Mahapatra, 2018)for flexible flow-shop scheduling problem is incorporated to FFJSP in the present work. This local search is done in three steps. (i) Sequence swap, (ii) Machine swap and (iii) Mutation strategy.

I. Sequence swap

Critical operations are very crucial to apply this local search. Therefore, critical operations are to be found first. There is possibility of critical operations being assigned to a same machine. On each machine, the critical operations which are assigned adjacent to each other are swapped and makespan of the new schedules are to be found. If the swapped schedule provides a better makespan value, then old schedule is replaced by this new schedule.

II. Machine swap

In FJSP, flexibility exists to choose one machine among several machines. By using the priority order matrix, each critical operation is assigned to each of the first three priority machines from priority order matrix separately and check whether the new schedule gives better makespan value. This method is named as machine swap technique. If the swapped schedule provides an improved value of makespan, then old schedule is replaced by this new schedule. It is observed that there is a quicker rate of convergence in both TLBO and JAYA techniques. But, it should be noted that the diversity in solutions is lost. Also, solutions get struck at the local optima. Therefore, in order to preserve the diversity in the solutions, mutation phenomenon (taken from genetic algorithm) is embedded to TLBO and JAYA techniques. This mutation strategy is previously implemented in Singh and Mahapatra (Singh & Mahapatra, 2016). The next section gives a brief description of the proposed mutation strategy.

II. Mutation strategy

When TLBO and JAYA are directly applied to the FFJSP, it is observed that they show a tendency to get trapped at the local optima. As the iterations proceed, all population converge towards the local optima. This is due to loss of diversity in the population. This leads to no improvement in optimal solution in later iterations. Therefore, to maintain the diversity, mutation strategy from genetic algorithm is embedded to TLBO and JAYA algorithms to alleviate this drawback. During an evolution process the sudden variation that happens in the genes of an off spring is called mutation. In the present work, if the best solution does not improve up to a specified iteration count, then mutation technique is applied to create some diversity in the population. Thus the convergence of population towards the local optima is prevented and diversity in population is maintained. It is found that the implementation of mutation strategy enhanced the performance of TLBO and JAYA algorithms.

PROPOSED ALGORITHMS

I. TLBO

1. Initialize the problem with input data like number of machines, jobs, machines available for each operation and their respective processing times.
2. Initialize the students (population). Initial subject values of the students are generated randomly within the range S_{ij} = min S_{ij} + r × (max S_{ij} – min S_{ij}) where S_{ij} is the subject value of i^{th} job's j^{th} operation, min S_{ij} =1, max S_{ij} = 1+tam and 'r' is a random number between (0,1).
3. Generate the schedules with the real number encoding system and with the problem mapping mechanism discussed above.
4. Now calculate the makespan using the population (knowledge of the students).
5. Now evaluate the mean of students' knowledge of the class.
6. Update the students' knowledge using the equations 3,4 and 5 in teacher phase and student phase.
7. Update the population using local search technique.

8. Now best student of the class is identified and it should be replaced with the teacher of the class (only if the best student functional value is better than the teacher).
9. Repeat the cycle to step 3 until the termination criteria is met.
10. End.

II. JAYA

1. Initialize the problem with input data like number of machines, jobs, machines available for each operation and their respective processing times.
2. Initialize the population. Initial subject values of the population are generated randomly within the range S_{ij} = min S_{ij} + r × (max S_{ij} – min S_{ij}) where the value of i^{th} job's j^{th} operation, min S_{ij} =1, max S_{ij} = 1+tam and 'r' is a random number between (0,1).
3. Generate the schedules with the real number encoding system and with the problem mapping mechanism discussed above.
4. Update the population using local search technique.
5. Now calculate the makespan using the population.
6. Update the population using the equation 6.
7. Repeat the cycle to step 3 until the termination criteria is met.
8. End.

RESULTS AND DISCUSSION

For illustration purpose, optimized makespan result of the example problem is explained briefly. Figures 2a and 2b show the final Gantt chart schedule obtained for the data shown in table 4. Figure 2a is without the consideration of fuzzy durations while the figure 2b is under the consideration of fuzzy durations. In all the four figures 1a, 1b, 2a and 2b, yellow color represents first job, blue color represents second job and the green color represents third job. In figures 1b and 2b, each machine consists of two rows (i.e.; top row and bottom row). The bottom row shows the range for starting time of an operation and the top row shows the range for completion time of an operation.

TABLE 5
Stochastic value representation

Operation	O11	O12	O21	O22	O23	O31	O32
Value	1.3570	1.3997	1.1498	1.4102	1.4858	1.3922	1.43 77
Priority	1	1	1	1	1	1	1
Machine allotment	2	3	1	2	2	1	3

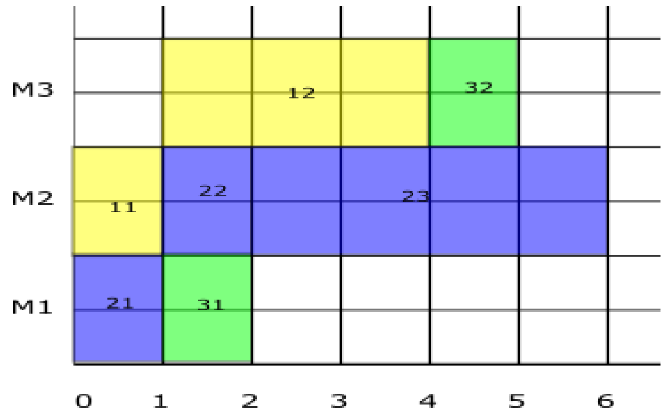

FIGURE 2A
FINAL GANTT CHART SCHEDULE FROM DATA IN TABLE 4 WITHOUT FUZZY DURATIONS

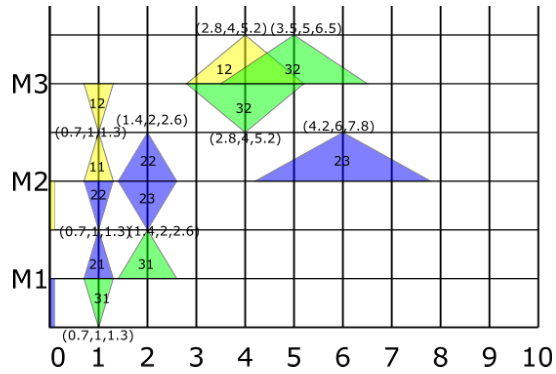

FIGURE 2B
FINAL FUZZY GANTT CHART SCHEDULE FROM DATA IN TABLE 4

To evaluate the efficiency of proposed TLBO and JAYA algorithms, experiments have been conducted on Kacem's and Brandimarte's fuzzy FJSP data instances. Each problem has been solved for 30 number of runs and the best results are tabulated in table 5 and the average of results for 30 runs in given table 6. Also, by using the most likely time values from the results, the relative error percentage (RE%) from lower bound (LB) has been calculated in columns 5 and 7 of tables 5 and 6 respectively. The formula to find the relative error percentage is given in the equation 7. Experiments have been conducted using MATALB software running at 3.40GHz on a 4 GB ram, windows 7 platform with an i7 processor.
The relative error percentage (RE%) with respect to a LB is defined as

$$RE = \frac{E(Cmax)-LB}{LB} \times 100 \qquad (7)$$

TABLE 6
BEST RESULTS OF PROPOSED TLBO AND JAYA

Sl. No.	LB	Problem Size	TLBO	RE%	JAYA	RE%
1	11	4 × 5	7.7,11,14.3	0	7.7,11,14.3	0
2	12	8 × 8	9.8,14,18.2	16.667	10.5,15,19.5	25
3	11	10 × 7	7.7,11,14.3	0	7.7,11,14.3	0
4	7	10 × 10	4.9,7,9.1	0	5.6,8,10.4	14.286
5	10	15 × 10	8.4,12,15.6	20	9.8,14,18.2	40
6 MK01	36	10 × 6	28,40,52	11.111	30.1,43,55.9	19.444
7 MK02	24	10 × 6	19.6,28,36.4	16.667	21.731,40.3	29.167
8 MK03	204	15 × 8	142.8,204,265.2	0	142.8,204,265.2	0
9 MK04	48	15 × 8	44.1,63,81.9	31.25	46.9,67,87.1	39.583
10 MK05	168	15 × 4	120.4,172,223.6	2.381	122.5,175,227.5	4.1667
11 MK06	33	10 × 15	45.5,65,84.5	96.97	48.3,69,89.7	109.09
12 MK07	133	20 × 5	100.8,144,187.2	8.2707	104.3,149,193.7	12.03
13 MK08	523	20 × 10	366.1,523,679.9	0	366.1,523,679.9	0
14 MK09	299	20 × 10	217.7,311,414.7	4.0134	220.5,315,419.9	5.3512
15 MK10	165	20 × 15	149.8,214,278.2	29.697	158.9,227,295.1	37.576
		Average RE%		**15.802**		**22.38**

TABLE 7
AVERAGE RESULTS OF PROPOSED TLBO AND JAYA

SI. No.	LB	Problem Size	TLBO	RE%	JAYA	RE%
1	11	4 × 5	7.7,11,14.3	0	7.7,11,14.3	0
2	12	8 × 8	10.08,14.4,18.72	20	10.715,15.3,19.89	27.5
3	11	10 × 7	7.98,11.4,14.82	3.6364	8.12,11.6,15.08	5.455
4	7	10 × 10	5.39,7.7,10.01	10	5.88,8.4,10.92	20
5	10	15 × 10	8.78,12.57,16.33	25.7	10.01,14.3,18.59	43
6 MK 01	36	10 × 6	28.68,40.97,53.26	13.806	30.54,43.63,56.73	21.19
7 MK 02	24	10 × 6	20.23,28.9,37.57	20.417	21.19,31.7,41.21	32.08
8 MK 03	204	15 × 8	143.22,204.6,265.98	0.2941	143.55,205.067,266.59	0.523
9 MK 04	48	15 × 8	44.99,64.27,83.55	33.896	47.55,67.93,88.31	41.52
10 MK 05	168	15 × 4	121.12,173.03,224.94	2.994	123.03,175.77,228.52	4.625
11 MK 06	33	10 × 15	46.55,66.5,86.45	101.52	48.62,69.47,89.84	110.5
12 MK 07	133	20 × 5	101.62,145.16,188.08	9.1429	104.83,149.77,194.7	12.61
13 MK 08	523	20 × 10	366.38,523.4,680.42	0.0765	366.52,523.6,680.68	0.115
14 MK 09	299	20 × 10	218.47,312.1,416.13	4.3813	221.06,315.86,421.03	5.639
15 MK 10	165	20 × 15	150.85,215.5,280.15	30.606	159.6,228,296.4	38.18
		Average RE%		**18.431**		**24.2**

Even though there is a lot of research work carried out on these benchmark problems in deterministic form, no research work is carried out on the above benchmark problems in fuzzy duration form. Therefore, to compare the obtained results of TLBO and JAYA, there are no other results available from the literature. Moreover, the present work shows a path to the future researchers to solve these complex benchmark fuzzy FJSP problems. Tables 5 and 6 consist of seven columns. First column represents the serial number. Second column shows the lower bound value of that particular problem. Third column gives the size of the problem. Fourth and sixth columns show the obtained fuzzy makespan values of the TLBO and JAYA algorithms respectively. Fifth and seventh columns show the relative error percentage of the obtained makespan from the lower bound values of TLBO and JAYA algorithms respectively. From the best results (table 5), it is clear that, TLBO attained lower bound values to five problems and the JAYA attained lower bound values to four problems. The average relative error percentage of all the problems of TLBO is 15.802 and that of JAYA is 22.38. From the average results (table 6), it is clear that, TLBO and JAYA attained lower bound values to one problem each. The average relative error percentage of all the problems of TLBO is 18.431 and that of JAYA is 24.2. By comparing the fuzzy makespan and relative error percentage values from tables 5 and 6, it is clear that the TLBO gives better results than JAYA. Therefore, we can say that TLBO clearly outperforms JAYA.

CONCLUSIONS

In this work, an effort has been made to solve fuzzy flexible job-shop scheduling problems using TLBO and JAYA algorithms. A new local search method is embedded to the basic algorithms to enhance the efficiency the algorithms. Based on the Kacem's and Brandimarte's benchmark problems of deterministic FJSP, a new complex and challenging problems for fuzzy FJSP (FFJSP) have been proposed by successfully by converting them into fuzzy FJSP. There are many methods to convert a deterministic FJSP into fuzzy FJSP in the literature. The problem to find a best method to convert deterministic FJSP to fuzzy FJSP still persists. To address this issue, an effort has been made and some new challenging benchmark problems have been proposed in this work. This new set of benchmark problems are larger in size and are more complex than the existing benchmark problems that are available in the literature of fuzzy FJSP. Therefore, the new benchmark problems, proposed in this work, are very much useful in determining the efficiency of different algorithms in solving different fuzzy FJSP problems in the future. Computational experiments have been carried out on the proposed fuzzy FJSP problems to test the performance of the proposed TLBO and JAYA algorithms. From results and discussion, it is clear that the TLBO gives better results than JAYA both in terms of best makespan and average makespan values. Even in terms of deviation from the lower bound (RE%), TLBO is superior to JAYA. Also, an example problem is discussed for the ease of understanding to the readers.

FUTURE WORK

In future, the work can be extended to study the performance of different algorithms that are available in the literature to solve FFJSP proposed in this paper. Also the work can be extended to study the multi-objective optimization of different uncertainties like machine breakdown, involved in FJSP using the TLBO and JAYA algorithms.

REFERENCES

[1] Apornak, A., Raissi, S., & Pourhassan, M. R. (2021). Solving flexible flow-shop problem using a hybrid multi criteria Taguchi based computer simulation model and DEA approach. *Journal of Industrial and Systems Engineering, 13*(2), 264-276.

[2] Balin, S. (2011). Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. *Information Sciences, 181*(17), 3551-3569. doi: https://doi.org/10.1016/j.ins.2011.04.010

[3] BORTOLAN, G., & DEGANI, R. (1993). A review of some methods for ranking fuzzy subsets *Readings in Fuzzy Sets for Intelligent Systems* (pp. 149-158): Elsevier.

[4] Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research, 41*(3), 157-183. doi: https://doi.org/10.1007/BF02023073

[5] Bruni, M. E., Beraldi, P., Guerriero, F., & Pinto, E. (2011). A heuristic approach for resource constrained project scheduling with uncertain activity durations. *Computers & Operations Research, 38*(9), 1305-1318. doi: https://doi.org/10.1016/j.cor.2010.12.004

[6] Buddala, R., & Mahapatra, S. S. (2016). *An effective teaching learning based optimization for flexible job shop scheduling.* Paper presented at the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).

[7] Buddala, R., & Mahapatra, S. S. (2018). Improved teaching–learning-based and JAYA optimization algorithms for solving flexible flow shop scheduling problems. *Journal of Industrial Engineering International, 14*(3), 555-570. doi: 10.1007/s40092-017-0244-4

[8] Buddala, R., Mahapatra, S. S., & Singh, M. R. (2021). Solving multi-objective flexible flow-shop scheduling problem using teaching-learning-based optimisation embedded with maximum deviation theory. *International Journal of Industrial and Systems Engineering*. doi: 10.1504/IJISE.2021.10037222

[9] Celano, G., Costa, A., & Fichera, S. (2003). An evolutionary algorithm for pure fuzzy flowshop scheduling problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 11*(06), 655-669. doi: https://doi.org/10.1142/S0218488503002466

[10] Chaari, T., Chaabane, S., Aissani, N., & Trentesaux, D. (2014). *Scheduling under uncertainty: Survey and research directions.* Paper presented at the Advanced Logistics and Transport (ICALT), 2014 International Conference on.

[11] Chanas, S., & Kasperski, A. (2003). On two single machine scheduling problems with fuzzy processing times and fuzzy due dates. *European Journal of Operational Research, 147*(2), 281-296. doi: https://doi.org/10.1016/S0377-2217(02)00561-1

[12] Dubois, D., Fargier, H., & Fortemps, P. (2003). Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research, 147*(2), 231-252. doi: https://doi.org/10.1016/S0377-2217(02)00558-1

[13] Dunke, F., & Nickel, S. (2022). A multi-method approach to scheduling and efficiency analysis in dual-resource constrained job shops with processing time uncertainty. *Computers & Industrial Engineering, 168*, 108067.

[14] Fortemps, P. (1997). Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions on Fuzzy Systems, 5*(4), 557-569. doi: https://doi.org/10.1109/91.649907

[15] Gao, K.-Z., Suganthan, P. N., Pan, Q.-K., Chua, T. J., Cai, T. X., & Chong, C.-S. (2016). Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *Journal of Intelligent Manufacturing, 27*(2), 363-374. doi: https://doi.org/10.1007/s10845-014-0869-8

[16] Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research, 1*(2), 117-129. doi: https://doi.org/10.1287/moor.1.2.117

[17] Ghrayeb, O. A. (2003). A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. *Applied Soft Computing, 2*(3), 197-210. doi: https://doi.org/10.1016/S1568-4946(02)00069-8

[18] Gonzalez-Rodriguez, I., Puente, J., Vela, C. R., & Varela, R. (2008). Semantics of schedules for the fuzzy job-shop problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38*(3), 655-666. doi: https://doi.org/10.1109/TSMCA.2008.918603
[19] Gonzalez-Rodriguez, I., Vela, C. R., & Puente, J. (2007). *A memetic approach to fuzzy job shop based on expectation model.* Paper presented at the Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International.
[20] González-Rodríguez, I., Vela, C. R., & Puente, J. (2006). *Study of objective functions in fuzzy job-shop problem.* Paper presented at the International Conference on Artificial Intelligence and Soft Computing.
[21] Rodrıguez, I., Vela, C. R., Puente, J., & Varela, R. (2008). *A new local search for the job shop problem with uncertain durations.* Paper presented at the Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling.
[22] He, W., Sun, D., & Liao, X. (2013). Applying novel clone immune algorithm to solve flexible job shop problem with machine breakdown. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE, 10*(9), 2783-2797. doi: https://doi.org/10.12733/jics20101851
[23] Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28*(3), 392-403. doi: https://doi.org/10.1109/5326.704576
[24] Jin, L., Zhang, C., Wen, X., Sun, C., & Fei, X. (2021). A neutrosophic set-based TLBO algorithm for the flexible job-shop scheduling problem with routing flexibility and uncertain processing times. *Complex & Intelligent Systems, 7*(6), 2833-2853.
[25] Kacem, I., Hammadi, S., & Borne, P. (2002a). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32*(1), 1-13. doi: https://doi.org/10.1109/TSMCC.2002.1009117
[26] Kacem, I., Hammadi, S., & Borne, P. (2002b). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and computers in simulation, 60*(3-5), 245-276. doi: https://doi.org/10.1016/S0378-4754(02)00019-8
[27] Kasperski, A. (2007). Some general properties of a fuzzy single machine scheduling problem. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 15*(01), 43-56. doi: https://doi.org/10.1142/S0218488507004364
[28] Lei, D. (2008). Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems. *The International Journal of Advanced Manufacturing Technology, 37*(1-2), 157-165. doi: https://doi.org/10.1007/s00170-007-0945-8
[29] Lei, D. (2010a). A genetic algorithm for flexible job shop scheduling with fuzzy processing time. *International Journal of Production Research, 48*(10), 2995-3013. doi: https://doi.org/10.1080/00207540902814348
[30] Lei, D. (2010b). Solving fuzzy job shop scheduling problems using random key genetic algorithm. *The International Journal of Advanced Manufacturing Technology, 49*(1-4), 253-262. doi: https://doi.org/10.1007/s00170-009-2379-y
[31] Lei, D. (2011). Scheduling fuzzy job shop with preventive maintenance through swarm-based neighborhood search. *The International Journal of Advanced Manufacturing Technology, 54*(9-12), 1121-1128. doi: https://doi.org/10.1007/s00170-010-2989-4
[32] Lei, D. (2012). Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied Soft Computing, 12*(8), 2237-2245. doi: https://doi.org/10.1016/j.asoc.2012.03.025
[33] Li, R., Gong, W., & Lu, C. (2022a). A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. *Expert Systems with Applications, 203*, 117380.
[34] Li, R., Gong, W., & Lu, C. (2022b). Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time. *Computers & Industrial Engineering, 168*, 108099.
[35] Lin, F.-T. (2002). Fuzzy job-shop scheduling based on ranking level (/spl lambda/, 1) interval-valued fuzzy numbers. *IEEE Transactions on Fuzzy Systems, 10*(4), 510-522. doi: https://doi.org/10.1109/TFUZZ.2002.800659
[36] Ludwig, A., Möhring, R. H., & Stork, F. (2001). A computational study on bounding the makespan distribution in stochastic project networks. *Annals of Operations Research, 102*(1-4), 49-64. doi: https://doi.org/10.1023/A:1010945830113
[37] Niu, Q., Jiao, B., & Gu, X. (2008). Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time. *Applied Mathematics and Computation, 205*(1), 148-158. doi: https://doi.org/10.1016/j.amc.2008.05.086
[38] Ourari, S., Berrandjia, L., Boulakhras, R., Boukciat, A., & Hentous, H. (2015). Robust approach for centralized job shop scheduling: Sequential flexibility. *IFAC-PapersOnLine, 48*(3), 1960-1965. doi: https://doi.org/10.1016/j.ifacol.2015.06.375
[39] Palacios, J. J., González-Rodríguez, I., Vela, C. R., & Puente, J. (2014). Robust swarm optimisation for fuzzy open shop scheduling. *Natural Computing, 13*(2), 145-156. doi: https://doi.org/10.1007/s11047-014-9413-1
[40] Palacios, J. J., González, M. A., Vela, C. R., González-Rodríguez, I., & Puente, J. (2015). Genetic tabu search for the fuzzy flexible job shop problem. *Computers & Operations Research, 54*, 74-89. doi: https://doi.org/10.1016/j.cor.2014.08.023
[41] Palacios, J. J., Puente, J., Vela, C. R., & González-Rodríguez, I. (2016). Benchmarks for fuzzy job shop problems. *Information Sciences, 329*, 736-752. doi: https://doi.org/10.1016/j.ins.2015.09.042
[42] Peng, J., & Liu, B. (2004). Parallel machine scheduling models with fuzzy processing times. *Information Sciences, 166*(1-4), 49-66. doi: https://doi.org/10.1016/j.ins.2003.05.012
[43] Petrovic, S., Fayad, C., Petrovic, D., Burke, E., & Kendall, G. (2008). Fuzzy job shop scheduling with lot-sizing. *Annals of Operations Research, 159*(1), 275-292. doi: https://doi.org/10.1007/s10479-007-0287-9
[44] Puente Peinador, J., Rodríguez Vela, M. d. C., & González Rodríguez, I. (2010). Fast local search for fuzzy job shop scheduling. *Frontiers in Artificial Intelligence and Applications*. doi: http://dx.doi.org/10.3233/978-1-60750-606-5-739
[45] Raissi, S., Rooeinfar, R., & Ghezavati, V. R. (2019). Three Hybrid Metaheuristic Algorithms for Stochastic Flexible Flow Shop Scheduling Problem with Preventive Maintenance and Budget Constraint. *Journal of Optimization in Industrial Engineering, 12*(2), 131-147.
[46] Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations, 7*(1), 19-34. doi: http://dx.doi.org/10.5267/j.ijiec.2015.8.004
[47] Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303-315. doi: https://doi.org/10.1016/j.cad.2010.12.015
[48] Rooeinfar, R., Raissi, S., & Ghezavati, V. (2019). Stochastic flexible flow shop scheduling problem with limited buffers and fixed interval preventive maintenance: a hybrid approach of simulation and metaheuristic algorithms. *Simulation, 95*(6), 509-528.
[49] Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research, 120*(2), 393-407. doi: https://doi.org/10.1016/S0377-2217(99)00094-6

[50] Sakawa, M., & Mori, T. (1999). An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy duedate. *Computers & Industrial Engineering, 36*(2), 325-341. doi: https://doi.org/10.1016/S0360-8352(99)00135-7
[51] Singh, M. R., & Mahapatra, S. S. (2016). A quantum behaved particle swarm optimization for flexible job shop scheduling. *Computers & Industrial Engineering, 93*, 36-44. doi: https://doi.org/10.1016/j.cie.2015.12.004
[52] Sowinski, R., & Hapke, M. (2000). *Scheduling under fuzziness*: Physica-Verlag.
[53] Subramaniam, V., & Raheja, A. S. (2003). mAOR: A heuristic-based reactive repair mechanism for job shop schedules. *The International Journal of Advanced Manufacturing Technology, 22*(9-10), 669-680. doi: https://doi.org/10.1007/s00170-003-1601-6
[54] Tavakkoli-Moghaddam, R., Safaei, N., & Kah, M. (2008). Accessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach. *Journal of the Operational Research Society, 59*(4), 431-442. doi: https://doi.org/10.1057/palgrave.jors.2602351
[55] Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling, 6*(1), 39-62. doi: https://doi.org/10.1023/A:1022235519958
[56] Wang, C., Tian, N., Ji, Z., & Wang, Y. (2017). Multi-objective fuzzy flexible job shop scheduling using memetic algorithm. *Journal of Statistical Computation and Simulation, 87*(14), 2828-2846. doi: https://doi.org/10.1080/00949655.2017.1344846
[57] Wang, K., & Choi, S. (2012). A decomposition-based approach to flexible flow shop scheduling under machine breakdown. *International Journal of Production Research, 50*(1), 215-234. doi: https://doi.org/10.1080/00207543.2011.571456
[58] Wang, L., Zhou, G., Xu, Y., & Liu, M. (2013). A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. *International Journal of Production Research, 51*(12), 3593-3608. doi: https://doi.org/10.1080/00207543.2012.754549
[59] Wang, L., Zhou, G., Xu, Y., Wang, S., & Liu, M. (2012). An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 60*(1-4), 303-315. doi: https://doi.org/10.1007/s00170-011-3610-1
[60] Wang, S., Wang, L., Xu, Y., & Liu, M. (2013). An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *International Journal of Production Research, 51*(12), 3778-3793. doi: https://doi.org/10.1080/00207543.2013.765077
[61] Wang, X., Gao, L., Zhang, C., & Li, X. (2012). A multi–objective genetic algorithm for fuzzy flexible job–shop scheduling problem. *International Journal of Computer Applications in Technology, 45*(2-3), 115-125. doi: https://doi.org/10.1504/IJCAT.2012.050700
[62] Wang, X., Li, W., & Zhang, Y. (2013). An improved multi–objective genetic algorithm for fuzzy flexible job–shop scheduling problem. *International Journal of Computer Applications in Technology, 47*(2-3), 280-288. doi: https://doi.org/10.1504/IJCAT.2013.054360
[63] Xu, Y., Wang, L., Wang, S.-y., & Liu, M. (2015). An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Neurocomputing, 148*, 260-268. doi: https://doi.org/10.1016/j.neucom.2013.10.042
[64] Yuan, Y., Xu, H., & Yang, J. (2013). A hybrid harmony search algorithm for the flexible job shop scheduling problem. *Applied Soft Computing, 13*(7), 3259-3272. doi: https://doi.org/10.1016/j.asoc.2013.02.013
[65] Zheng, Y. L., & Li, Y. X. (2012). Artificial bee colony algorithm for fuzzy job shop scheduling. *International Journal of Computer Applications in Technology, 44*(2), 124-129. doi: https://doi.org/10.1504/IJCAT.2012.048682