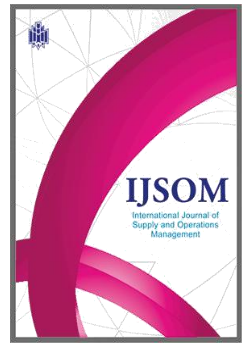
International Journal of Supply and Operations Management

IJSOM

2023, Volume 10, Issue 3, pp. 319-336
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Genetic Algorithm based on Greedy Strategy in Unrelated Parallel-Machine Scheduling Problem Using a Fuzzy Approach with Periodic Maintenance and Process Constraints

*Mohammad Yaghtin [a], Youness Javid [a] **

[a] *Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran*

Abstract

Nowadays, in production environments where the production system is parallel machines, the reliability of the machines is important and the uncertainty of scheduling parameters is common. The focus of this paper is on the unrelated parallel machine scheduling problem using a fuzzy approach with machines maintenance activities and process constraints is of concern. An important application of this problem is in the production of products that the due dates are defined as a time window and the best due date is close to the middle of the time window and the jobs processing times depend on other factors such as operator and their value is not specified and are announced as interval under uncertainty. To begin the study, a fuzzy mathematical model is introduced in which changing between a fuzzy approach and a deterministic model is described. Then, Due to the NP-hard nature of the problem, a fuzzy-based genetic algorithm has been developed to address large instances. In this algorithm, a greedy decoding approach according to fuzzy parameters is developed. Numerical experiments are used assess the efficacy of the developed algorithm. It is concluded that the proposed algorithm shows great performance in large instances and is superior to the proposed mathematical model in small instances too.

Keywords: unrelated parallel-machine scheduling, fuzzy process times, fuzzy due dates, availability constraint, genetic algorithm.

1. Introduction

Scheduling is a crucial decision-making process that is widely used in both service and production industries. Its primary objective is to allocate resources to jobs within specific time intervals, with the aim of optimizing one or more criteria. In general, smart scheduling methods are required to assign jobs to machines in case of resource and time constraints. In classical scheduling problems, the planning horizon assumes continuous availability of the machines. In some cases, this assumption may be justifiable, however, it's not an acceptable assumption in many real-world cases. A process on a machine can be interrupted in some time intervals due to preventive maintenance or inspection. Thus, using scheduling approaches according to machine unavailability in specific intervals is essential. On the other hand, process times are known in advance, while this assumption is not appropriate in real-world applications. Problem parameters like process times and due dates are not constant values in cases that human factors affect the problem. Hence, parameters are fuzzy in the proposed problem.

*Corresponding author email address: youness.javid@gmail.com
DOI: 10.22034/IJSOM.2023.109377.2359

2. Literature review

Numerous researchers have extensively studied the parallel-machine scheduling problem which contain availability constraint in a planning horizon, recently. Sun and Li [1] considered two parallel machines in their scheduling problems where each machine was unavailable in time intervals due to maintenance activities. Additionally, it is important to note that each machine has a maximum consecutive working time, which must not surpass a predetermined upper limit. They proposed two different models: in the first model, maintenance activities were performed periodically and the objective function was minimizing the makespan. In the second model, process times were known in advance and the objective function was minimizing the total completion time of jobs. A polynomial algorithm was introduced for the first model and Shortest Process Time (SPT) method was applied for the second one. Zhao et al studied a parallel-machine scheduling problem to minimize total weighted completion time. It was assumed that there is only one unavailable machine at a specific time. The researchers also demonstrated that the proposed problem is NP-hard. To address this challenge, they developed a fully polynomial-time approximation scheme for solving the problem, which they extended to cases involving more than two machines. Tan et al. [3] focused on the minimization of the total completion time for two parallel, identical machines. The problem also incorporated pre-specified periods of unavailability for each machine. They demonstrated that when each machine has a single pre-specified period of unavailability and the unavailability periods of the two machines do not overlap, the shortest process time (SPT) method has a tight worst-case ratio. Xu and Yang [4] considered two machines scheduling problem to minimize makespan where one machine is unavailable periodically. Wang and Cheng [5] studied two parallel machine scheduling problems to maximize the number of on-time jobs considering that one machine is unavailable in a specific time interval. They also proposed a heuristic for the problem. Lee and Kim [6] developed a branch and bound algorithm for a two-machine scheduling problem to minimize total tardiness subject to availability constraints due to maintenance tasks for each machine. He et al. [7] conducted a study on a scheduling problem that involves two machines with availability dependent on the machine. They employed a heuristic algorithm with the aim of reducing the makespan. Shen et al. [8] considered a non-simultaneous machine availability constraint in a parallel-machine scheduling problem where machines may not be available at time zero. The proposed problem features two objective functions. The first objective function is focused on minimizing the total completion time and the total absolute differences in completion times, while the second objective function aims to minimize the total waiting time and the total absolute differences in waiting times. They indicated a polynomial-time algorithm to find optimal solutions. Huo and Zhao [9] presented a total completion time minimization subject to a constraint that makespan mustn't extend a constant value. They also developed a polynomial algorithm assuming that at least 2 machines must be available in each time interval where preemption was allowed. Lee et.al [10] developed a branch and bound and a hybrid genetic algorithm to minimize total tardiness subject to maintenance activity. Recently, Yin et al. [11] studied a parallel-machine scheduling problem to minimize the expected total completion time of jobs given that disruption of some of the machines may occur. They represented an approximation scheme for the proposed problem. Suresh and Ghaudhuri [12] considered makespan minimization under the condition that machines are unavailable in a time interval and presented a genetic algorithm to solve the problem. Jiang and Tan [13] studied scheduling with job rejection in which some machines are not available at time zero. They provided a heuristic algorithm to minimize makespan and the total cost of rejecting and processing jobs. The worst-case ratio bound for the proposed problem was 2. Ahmadizar et al. [14] proposed an Imperialist Competitive Algorithm for an unrelated parallel-machine scheduling problem with setup times where machines are unavailable in a specific time interval. The goal is to minimize total tardiness and earliness. Each machine can process a specific set of jobs. Rosales et al. [15] investigated a novel scheduling problem on unrelated parallel machines that incorporates preventive maintenance activities and setup times that depend on the sequence and machine. The authors introduce a mathematical formulation for this problem and derive valid inequalities to enhance its performance, enabling the model to obtain optimal solutions for small-medium instances. Additionally, they develop an efficient metaheuristic algorithm based on a multi-start strategy to solve larger instances. Lu et al. investigated an unrelated parallel machine scheduling problem with deteriorating maintenance activities, parallel-batching processing, and deteriorating jobs in [16]. The authors aim to minimize the makespan while considering maintenance activities that increase with their starting time. To tackle this problem, they present a mixed-integer programming model. Since the problem is known to be NP-hard, they propose a hybrid ABC-TS algorithm that combines artificial bee colony (ABC) and tabu search (TS) to obtain a reasonable solution in a reasonable time.

In recent years, unrelated parallel-machine scheduling problem in uncertain conditions has been studied by many researchers in which some parameters were fuzzy. Balin [17] considered a parallel machine scheduling problem with fuzzy process times to minimize makespan. A robust genetic algorithm was applied to solve the proposed problem. Chyu and Chang [18] considered bi-objective unrelated parallel-machine scheduling problem to minimize total completion time and total tardiness. Process times were triangular fuzzy numbers and due dates were trapezoidal fuzzy numbers. A simulated annealing and a greedy randomized adaptive search procedure were applied to solve the problem. Senthilkumar et al. [19] combinatorial fuzzy unrelated parallel-machine scheduling problem minimizing total earliness and tardiness. In order Toe problem, they presented particle swarm optimization and ant colony optimization. Alcan and BaşLıGil [20] considered triangular fuzzy process times for parallel-machine scheduling problem. They proposed a genetic algorithm to minimize makespan. Torabi et al. [21] studied a particle swarm optimization algorithm to solve fuzzy multi-objective unrelated parallel machine scheduling problem. The goal was to minimize total machine load variation. In their paper, process times and due dates were triangular fuzzy numbers. Behnamian [22] considered Bell-shaped fuzzy numbers for process times and proposed a PSO algorithm to minimize makespan in parallel machines scheduling problem. Yeh et al. [23] also considered triangular fuzzy numbers as process times to minimize makespan. They studied parallel-machine scheduling problem with learning effects and developed SA and GA. Naderi et al. [24] studied bi-objective unrelated parallel-machine scheduling problem to minimize workload imbalance and total lateness. In their study, process times, setup times, release dates, and due dates were trapezoidal numbers. Fuzzy non dominated particle swarm optimization and fuzzy non dominated sorting genetic algorithm II (FNSGA II) was proposed for the mentioned problem. Rostami et al. [25] considered the learning effect and deterioration effect for a bi-objective parallel-machine scheduling problem with triangular fuzzy process times and due dates. The objective functions were total tardiness/earliness and makespan. In this paper, a non-linear fuzzy-based mathematical model is presented. Since the proposed model is not appropriate for large-sized instances of the problem, a branch and bound algorithm is presented. Nailwal et al. [26] studied a bi-criteria scheduling problem in a fuzzy environment in which the first criterion was the minimization of total weighted completion times and the second criterion was the minimization of maximum tardiness. Moreover, process times were triangular fuzzy numbers, and a heuristic algorithm was implemented for the proposed problem. Liao and Su [27] studied unrelated parallel machine scheduling problem with fuzzy process times, setup times, and release dates, minimizing the makespan. Also, ant colony optimization was applied to solve the proposed problem. The problem of scheduling n single-operation jobs on m uniform parallel machines is investigated by Li et al. in [28]. The authors consider a scenario in which each machine consumes a certain amount of resources per unit time when processing the jobs. The objective is to minimize the makespan while ensuring that the total resource consumption does not exceed a given limit. To tackle this problem, the authors propose a fuzzy simplified swarm optimization (SSO) algorithm. The problem of scheduling on parallel batch processing machines with different capacities, in a fuzzy environment with non-identical job sizes and fuzzy processing times, is analyzed by Jia et al. in [29]. The authors aim to minimize the makespan for this problem. To address this challenge, they propose a mathematical model and a fuzzy ant colony optimization (FACO) algorithm. Rezaeian et al. [30] investigated the unrelated parallel machines scheduling problem with sequence-dependent setup times in [30], where the objective is to minimize the total weighted fuzzy earliness and tardiness penalties. The authors consider a fully fuzzy environment, where the jobs have fuzzy processing times, fuzzy setup times, and fuzzy due dates. To address this problem, the authors propose a mathematical model and a genetic algorithm. Liu et al. [31] considered a stochastic parallel machine scheduling problem, where the job release times and processing times are uncertain. The authors aim to minimize the total cost, which includes the setup cost on machines and the expected penalty cost of jobs' earliness and tardiness. To tackle this problem, the authors propose a two-stage stochastic program and an SAA (sample average approximation) method. Li et al. [32] studied a parallel machine scheduling problem with position-dependent deteriorating jobs and DeJong's learning effects in an uncertain manufacturing system. The authors aim to minimize the fuzzy makespan for this problem. To address this challenge, they propose a fuzzy simplified swarm optimization algorithm with local search. The authors evaluate the proposed algorithm by comparing it with other state-of-the-art meta-heuristics. Ahmet Arik et al. analyzed an unrelated parallel machine scheduling problem with a restrictive common due date in [33]. The authors aim to minimize the total sum of earliness/tardiness costs for this problem. To tackle this problem, they propose four different construction algorithms that ensure a balanced number of jobs or workload per machine. Khalifa [34] examined the problem of single machine scheduling with distinct due dates in a fuzzy environment, with the goal of minimizing both total earliness and tardiness. To accomplish this, the author introduced a method for sequencing jobs on a single machine in such an environment.

The multiple objective fractional fixed-charge transportation problem (MFFTP) was investigated in a rough decision-making framework by Midya et al. [35]. The problem involves fuzzy parameters in the model design, which are handled through the use of different types of fuzzy scales. The authors applied the fuzzy chance-constrained rough approximation (FCRA) technique to extract the optimal solution with the greatest preference from the MFFTP they proposed. Midya et al. [36] investigated the multi-stage multi-objective fixed-charge solid transportation problem (MMFSTP) within the context of a green supply chain network system, under an intuitionistic fuzzy environment. To handle the fuzzy parameters, trapezoidal intuitionistic fuzzy numbers were assumed and the expected value operator was used to convert the intuitionistic fuzzy MMFSTP into a deterministic MMFSTP. The authors developed methodologies to solve the deterministic MMFSTP through weighted Tchebycheff metrics programming and min-max goal programming, yielding Pareto-optimal solutions. Roy et al. [37] the multi-objective fixed-charge solid transportation problem with product blending in an intuitionistic fuzzy environment. A novel ranking method was employed to convert the intuitionistic fuzzy multi-objective fixed-charge solid transportation problem with product blending into a deterministic form. The authors then proposed a fuzzy technique for order preference by similarity to the ideal solution, which was used to derive Pareto-optimal solutions from the proposed model. Mondal et al. [38] proposed an integrated model for incorporating sustainability in a multi-objective, multi-item, multi-choice, step fixed-charge solid transportation problem in an intuitionistic fuzzy environment, considering economic, customer satisfaction, and social aspects. To address the uncertainty of the problem, the authors developed a novel ranking concept based on total integral values. Additionally, two equivalent models were presented by utilizing the ranking concept and the possibility measure, respectively, and were transformed into fully deterministic models by converting the multi-choice parameter into a single choice using binary variables. The multi-objective multi-item fixed-charge solid transportation problem (MOMIFCSTP) is studied by Roy et al. in [39]. The authors address the challenge of dealing with vague data that cannot be fully characterized by fuzziness or roughness, by introducing fuzzy-rough variables as coefficients of the objective functions and the constraints. To solve the deterministic MOMIFCSTP, they propose a methodology based on the technique for order preference by similarity to ideal solution. Tirkolaee et al. [40] introduced a new bi-objective mixed-integer linear programming (MILP) model for flexible manufacturing systems (FMS), which incorporates an outsourcing option and just-in-time delivery. The model aims to minimize both the total cost of the production system and total energy consumption. To solve this problem, the authors propose a hybrid technique that combines an interactive fuzzy solution technique with a self-adaptive artificial fish swarm algorithm. Saffarian et al. [41] developed an integer linear programming formulation for a new fuzzy multi-period multi-depot vehicle routing problem. To solve this problem, they propose a hybrid algorithm called HGSA, which combines a genetic algorithm with simulated annealing and an auction-based approach. The authors use a modern simulated annealing cooling schedule function to enhance the performance of the algorithm.

In this paper, an unrelated parallel-machine scheduling problem is considered in which process times and due dates are fuzzy numbers. Due to maintenance activities, machines are unavailable periodically and each machine can process a specific set of jobs. Moreover, jobs are available at specific times and setup times depend on machines and jobs sequence. As far as we know, there isn't any paper with fuzzy process times and fuzzy due dates that consider periodic maintenance activities. In other words, machine availability constraint in time intervals and also fuzzy parameters leads to the high complexity of the problem. The contributions are as follows:

- Considering the due dates of jobs as a time window under fuzzy conditions and considering the processing time of jobs in a state of uncertainty and dependent on other factors in unrelated parallel machines scheduling
- A fuzzy mathematical model based on fuzzy parameters is established for the unrelated parallel-machine scheduling problem with machines maintenance activities and process constraints.
- A fuzzy-based genetic algorithm according to fuzzy parameters is developed.
- In fuzzy-based GA, a greedy decoding approach according to fuzzy parameters and machine availability which increases the efficiency of the solution method is developed.

The rest of the paper is organized as follows. In section 3, fuzzy numbers and their details are discussed. Section 4 describes the problem and a fuzzy model is presented and converted to a deterministic model. In Section 5, a fuzzy-based genetic algorithm is proposed to solve large instances of the problem. Computational results are discussed in Section 6. Finally, conclusions and future research are provided in Section 7.

3. Preliminary

A triangular fuzzy number is illustrated as $\tilde{A} = (a_1, a_2, a_3)$. If the fuzzy number is normal, its membership function is defined as follows:

$$\mu(x) = \begin{cases} 0 & x \le a_1 \\ f_a(x) = \frac{x-a_1}{a_2-a_1} & a_1 \le x \le a_2 \\ 1 & x = a_2 \\ g_a(x) = \frac{a_3-x}{a_3-a_2} & a_2 \le x \le a_3 \\ 0 & x \ge a_3 \end{cases} \tag{3.1}$$

If $\tilde{A}$ and $\tilde{B}$ are triangular fuzzy numbers which are described below:

$\tilde{A} = (a_1, a_2, a_3)$

$\tilde{B} = (b_1, b_2, b_3)$

Then, $\tilde{A} + \tilde{B}$ and $\tilde{A} - \tilde{B}$ are triangular fuzzy numbers. It should be mentioned that $\tilde{A} \times \tilde{B}$ and $\tilde{A} \div \tilde{B}$ are fuzzy numbers but, they aren't triangular fuzzy numbers.

$$\tilde{A} + \tilde{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \quad (3.2)$$

$$\tilde{A} - \tilde{B} = (a_1 - b_3, a_2 - b_2, a_3 - b_1) \quad (3.3)$$

$$\tilde{A} \times \tilde{B} = (a_1.b_1, a_2.b_2, a_3.b_3) \quad (3.4)$$

$$\tilde{A} \div \tilde{B} = \left({}^{a_1}/_{b_3}, {}^{a_2}/_{b_2}, {}^{a_3}/_{b_1}\right) \quad (3.5)$$

α-cut in fuzzy sets are subsets of its members whose membership degree is greater than or equal to α. It is denoted as A_α:

$$\alpha - cut \rightarrow A_\alpha = \left[f_a{}^{-1}(\alpha), g_a{}^{-1}(\alpha)\right] \quad (3.6)$$

α-cut creates a deterministic distance which is defined as $[f_A^{-1}(\alpha), g_A^{-1}(\alpha)]$. According to Heilpern [42], the expected interval of the fuzzy number $\tilde{A}$ is $EI(\tilde{A})$ which is defined as follows:

$$EI(\tilde{A}) = [E_1^A, E_2^A] = \left[\int_0^1 f_a^{-1}(\alpha), \int_0^1 g_a^{-1}(\alpha)\right] \quad (3.7)$$

Moreover, $EV(\tilde{A})$ denotes the expected fuzzy number of $\tilde{A}$ which is shown in the following equation:

$$EV(\tilde{A}) = \frac{E_1^A + E_2^A}{2} \quad (3.8)$$

According to Eqs. (3.7) & (3.8), the expected interval and its expected value are calculated in Eqs. (3.9) & (3.10) if there exists a triangular fuzzy number $\tilde{A}$.

$$EI(\tilde{A}) = \left[\frac{a_1 + a_2}{2} + \frac{a_2 + a_3}{2}\right] \quad (3.9)$$

$$EV(\tilde{A}) = \frac{a_1 + 2\,a_2 + a_3}{4} \quad (3.10)$$

4. Problem description and formulation

The proposed problem is as follows: there are n jobs which are available at different intervals. Each job can be processed on one of m unrelated machines in which its completion time should be less than a specific number. The penalty is assigned for the jobs whose completion time is greater than a specific value. Also, each machine can process at most one job at a specific time. Other assumptions are listed in the following:

- Setup times depend on the type of machine and job-sequence
- Machines are unavailable at special times periodically due to maintenance activity.
- Each machine cannot process all jobs.

- Each scheduling period has a planning horizon in which jobs must be completed before the planning horizon.
- Preemption is not allowed. In other words, if a job is interrupted due to an unavailable period, it must start again (not resume). Thus, the start time of jobs must be adjusted so that it is completed before a start of an unavailable period.

According to the above assumptions, assigning jobs to machines and determining the start time of jobs must be carried out so that the total tardiness of jobs is minimized.

In the following section, a mixed-integer linear programming model (MIP) is presented. Firstly, a fuzzy mathematical model is proposed. Then, this model is converted to a deterministic mathematical model, following an approach by Jimenez in 1996.

4.1. Fuzzy mathematical model

The notations for the proposed model is as follows:

Indices

i: unavailable period ($i = 0,1, \dots, N$)

j, l: job ($j, l = 0,1, \dots, n$)

k: machine ($k = 0,1, \dots, m$)

Parameters

$\tilde{P}_{jk}$: Triangular fuzzy process time of job j on machine k ($\tilde{p}_{jk} = \left(p_{jk}^1, p_{jk}^2, p_{jk}^3\right)$)

$\tilde{d}_j$: Triangular fuzzy due date of job j ($\tilde{d}_j = \left(d_j^1, d_j^2, d_j^3\right)$)

s_{jlk}: Setup time of job l which is after job j on machine k

u_{kj}: Equals 1 if machine k can process job j, otherwise 0

r_j: release date of job j

T: Time interval between two consecutive maintenance activities on a machine ($SM_{i+1} - FM_i = T$; $\forall\, i$)

t: Duration of the unavailable period due to maintenance activity ($FM_i - SM_i = t$; $\forall\, i$)

SM_i: Start time of ith maintenance activity ($SM_{N+1} = H$)

FM_i: Completion time of ith maintenance activity ($FM_0 = 0$)

H: Planning horizon

N: Number of periods for each machine

M: Large number

Decision variable

x_j : Start time of machine setup to process job j

C_j : Completion time of job j

T_j : Tardiness of job j ($T_j = \max\{0, C_j - d_j\}$)

A_{kj}: Equals 1 if job j is assigned to machine k, otherwise 0

z_{jlk}: Equals 1 if job l is processed after job j on machine k, otherwise 0

z_{0jk}: Equals 1 if job j is processed on machine k as the first job, otherwise 0

y_{ij}: Equals 1 if job j is processed between maintenance activity i and i+1, otherwise 0

F_{jlk}: Binary variable for linearization purpose instead of the nonlinear $A_{kj}A_{kl}$

The mathematical model is presented in the following:

$$Min\ Z = \sum_{j=1}^{n} T_j \tag{4.1}$$

Subject to:

$$\sum_{k=1}^{m} u_{kj} A_{kj} = 1\ ;\ \forall\, j \tag{4.2}$$

$$\sum_{\substack{j=1 \\ j\neq l}}^{n} z_{jlk} + z_{0lk} = A_{kl}\ ;\ \forall\, k, l \tag{4.3}$$

$$\sum_{\substack{j=1 \\ j\neq l}}^{n} z_{ljk} \leq A_{kl}\ ;\ \forall\, k, l \tag{4.4}$$

$$\sum_{j=1}^{n} z_{0jk} \leq 1\ ;\ \forall\, k \tag{4.5}$$

$$z_{jlk} \leq F_{jlk}\ ;\ \forall\, j, l, k\ , j \neq l \tag{4.6}$$

$$F_{jlk} \leq \frac{A_{kj} + A_{kl}}{2}\ ;\ \forall\, j, l, k\ , j \neq l \tag{4.7}$$

$$A_{kj} + A_{kl} - 2F_{jlk} \leq 1\ ;\ \forall\, j, l, k\ , j \neq l \tag{4.8}$$

$$c_j \geq x_j + \sum_{k=1}^{m} \sum_{\substack{l=1 \\ l\neq j}}^{n} z_{ljk} s_{ljk} + \sum_{k=1}^{m} z_{ojk} s_{0jk} + \sum_{k=1}^{m} A_{kj} \tilde{p}_{jk}\ ;\ \forall\, j \tag{4.9}$$

$$x_j \geq r_j\ ;\ \forall\, j \tag{4.10}$$

$$FM_i - M(1 - y_{ij}) < x_j\ ;\ \forall\, i\ , j \tag{4.11}$$

$$SM_{i+1} + M(1 - y_{ij}) > x_j\ ;\ \forall\, i\ , j \tag{4.12}$$

$$FM_i - M(1 - y_{ij}) < c_j\ ;\ \forall\, i\ , j \tag{4.13}$$

$$SM_{i+1} + M(1 - y_{ij}) > c_j\ ;\ \forall\, i\ , j \tag{4.14}$$

$$c_j \leq x_l + M\left(1 - \sum_{k=1}^{m} Z_{jlk}\right);\ \forall\, j, l\ , j \neq l \tag{4.15}$$

$$c_j \leq H\ ;\ \forall\, j \tag{4.16}$$

$$T_j \geq c_j - \widetilde{d}_j\ ;\ \forall\, j \tag{4.17}$$

$$x_j\ , c_j\ , T_j\ \geq 0\ ;\ \forall\, j \tag{4.18}$$

$$A_{kj}\ , y_{ij}\ , z_{jlk}\ , z_{0jk}, F_{jlk}\ \geq 0\ ;\ \forall\, j \tag{4.19}$$

The objective function in Eq. (4.1) is minimizing total tardiness. Constraints (4.2) guarantee that each job is processed on one available machine. Constraints (4.3 to 4.8) assure that each assigned job appears once in the job sequence of the relative machine. According to the linearization approach, constraints (4.6 to 4.8) is applied instead of nonlinear constraint $z_{jlk} \leq A_{kj}A_{kl}$. Constraints (4.9) state that the completion time of each job must be greater than the summation of the relative machine setup time and respective process time. Constraints (4.10) guarantee that the start time of a job must be greater than its release date. Constraints (4.11 to 4.14) state that machines can process a job at the available time (not during a maintenance activity). Constraints (4.15) specify the job sequence on machines. Constraints (4.16) guarantee that jobs must be completed before the planning horizon. Constraints (4.17) calculate the total tardiness and constraints (4.18), (4.19) define variables of the model.

4.2. Converting the fuzzy mathematical model to a deterministic mathematical model

In this section, an approach is described to convert fuzzy mathematical model to deterministic mathematical model. According to the expected interval and expected value of a fuzzy number, Jimenez presented the following approach to solve a fuzzy linear programming model by converting a fuzzy problem to a deterministic problem. Thus, each fuzzy parameter of the model becomes deterministic parameters according to the α-cut method.

If the fuzzy mathematical model is as follows:

$$Min\ \tilde{C}X$$

Subject to:

$$\tilde{a_\imath}X \geq (\leq)\ \tilde{b_\imath}\ ;\quad i = 1, \dots, m \tag{4.20}$$

$$X \geq 0$$

According to Jimenez approach, the proposed fuzzy model in Eq. (30) is converted to a deterministic model which is shown in the following equation.

$$Min\ EV(\tilde{C})X$$

Subject to:

$$\left[\,(1-\alpha).E_2^{a_i} + \alpha.E_1^{a_i}\right]X \geq (\leq)\alpha.E_2^{b_i} + (1-\alpha).E_1^{b_i} \tag{4.21}$$

$$X \geq 0\ , \qquad \alpha \in [0,1]$$

In the presented fuzzy mathematical model, there are only two constraints (4.9 and 4.17) which include fuzzy parameters. The rest of the equations become unchanged. Eq. (4.9) is converted to the following equation:

$$c_j \geq x_j + \sum_{k=1}^{m}\sum_{\substack{l=1\\ l\neq j}}^{n} z_{ljk}s_{ljk} + \sum_{k=1}^{m} z_{ojk}s_{0jk} + \sum_{k=1}^{m} A_{kj}\left[(1-\alpha)\left(\frac{p_{jk}^2+p_{jk}^3}{2}\right) + \alpha\left(\frac{p_{jk}^2+p_{jk}^1}{2}\right)\right]\ ;\ \forall\, j \tag{4.22}$$

And Eq. (4.17) is converted to the following equation:

$$T_j \geq c_j - \left[\alpha\left(\frac{d_j^2+d_j^3}{2}\right) + (1-\alpha)\left(\frac{d_j^2+d_j^1}{2}\right)\right]\ ;\ \forall\, j \tag{4.23}$$

The fuzzy model is converted to the deterministic model by replacing Eqs. (4.22) & (4.23) with the related fuzzy constraints and adding α variable to decision variables.

5. Genetic Algorithm

Genetic algorithm (GA) is one of the most applicable metaheuristics in order to solve optimization problems. This algorithm can reach an optimal solution in different complex problems despite its simple logic. This algorithm is based on the gradual evolution of living in nature. This method was introduced by Holland and later developed by Goldberg.

The genetic algorithm starts with some random initial populations like other evolutionary algorithms. In this algorithm, the initial population is the first generation and each population member is a chromosome. A number of the best population members (elite) are considered as parents and crossover and mutation operations are carried out with p_c and p_m rates to generate children. The children are compared with the current population and the best populations are selected for the next generation according to the objective function and population size. This process continues till the termination condition is reached. If the algorithm is terminated, the solution of the last generation is considered as the algorithm output.

Given that process times and due dates are fuzzy parameters, computations must be according to fuzzy parameters and algorithm output for each instance (objective value of each instance) must be a fuzzy number. In the following, different components of the proposed algorithm are investigated.

5.1. Solution representation

A genetic algorithm is a population-based algorithm that starts with initial solutions (chromosomes). This initial solution can be generated randomly or according to other algorithms. In this paper, initial solutions are generated randomly. Also, a special permutation is applied to represent a solution whose gene numbers are equal to the number of jobs that are not scheduled. Thus, each encoded solution is an array with n cells in which a job number is in each cell. For example, a solution of the problem with 8 jobs and 3 machines is illustrated in Figure 1.

4	2	1	3

Figure 1. Solution representation (chromosome) for an instance with 8 jobs and 3 machines

To calculate the objective value of a solution, the solution is decoded. In other words, the array must be encoded by a scheduling plan. In this research, to decode each solution and calculate the objective value, the solution process starts with the job in the first cell and the job is assigned to a machine (among machines that can process the job) that can process the job earlier so that this job has the minimum tardiness.

According to fuzzy process times and due dates, there are a few points to consider for assigning jobs to machines and determining job sequence on each machine:

1. According to triangular fuzzy process times, completion times are resultantly triangular fuzzy numbers:

$$P_l = (p_l^1, p_l^2, p_l^3) \rightarrow C_l = (c_l^1, c_l^2, c_l^3) \tag{5.1}$$

Suppose job j is processed after job l on machine k. The start time of job j must be greater than c_l^3. In other words, the following equation is established between two consecutive jobs (j and l) on each machine:

$$c_l^3 \leq x_j \tag{5.2}$$

2. If the start time of job j on a machine is within interval(FM_i, SM_{i+1}), the minimum completion time of the job must be greater than FM_i and the maximum completion time must be less than SM_{i+1}. The following equation indicates the considered relation.

$$if\ FM_i \leq x_j \leq SM_{i+1} \rightarrow FM_i < c_j^1 \quad , \quad c_j^3 < SM_{i+1} \tag{5.3}$$

3. If tardiness of job j on a machine is indicated with a triangular fuzzy number with negative component, tardiness of the considered job is 0. Eq. (5.4) is the mathematical expression for this relation:

$$if\ T_j = \left(T_j^1, T_j^2, T_j^3\right) \leq 0 \quad \rightarrow \quad T_j = 0 \tag{5.4}$$

To better illustrate the decoding approach, consider an example with 4 jobs and 2 machines. Suppose there are two machines M_1 and M_2 and jobs J_1 and J_4 can only be processed on machine M_1 and jobs J_2 and J_3 can only be processed on machine M_2. Processing times and due dates of jobs are as follow in fuzzy numbers:

$p_1 = (2, 3, 4) \quad , \quad d_1 = (1, 2, 2.5)$

$p_2 = (1, 2, 3) \quad , \quad d_2 = (3.5, 4, 4.5)$

$p_3 = (1, 2, 3) \quad , \quad d_3 = (2, 5, 5.5)$

$p_4 = (0.5, 1, 1.5) \quad , \quad d_4 = (3, 4, 4.5)$

In machine M_1, parameters T and t are equal to 4 and 1 respectively and in machine M_2, parameters T and t are equal to 5 and 0.5 respectively. Now consider the encrypted solution in Figure 1. First, job J_4, which is in the first position, is assigned to machine M_1. After that, job J_2, which is in the second position, is assigned to machine M_2. Then jobs J_1 and J_3, which are in the third and fourth positions, are assigned to machine M_1 and M_2 respectively.

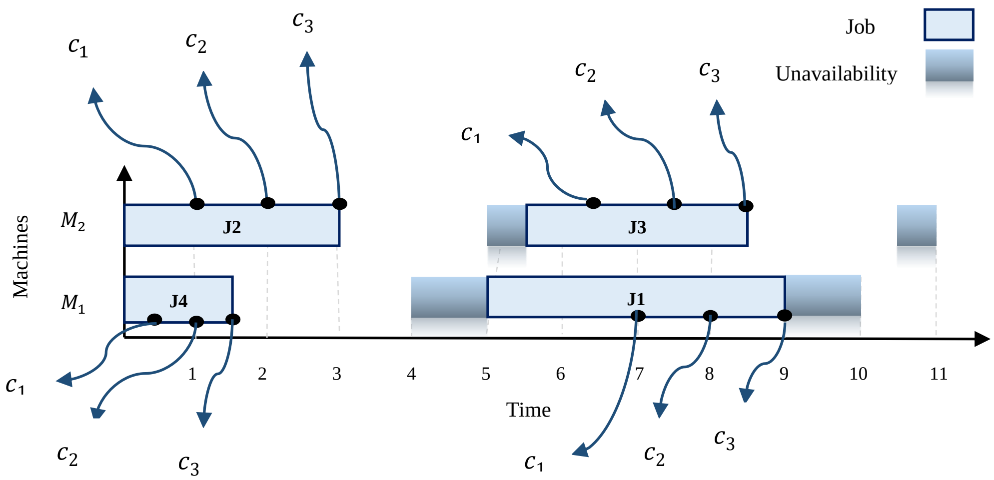


Figure 2: timetable for all jobs

According to the timetable in Figure 2, the tardiness of jobs are as follow:

$T_1 = c_1 - d_1 = (7, 8, 9) - (1, 2, 2.5) \Rightarrow T_1 = (4.5, 6, 8)$

$T_2 = c_2 - d_2 = (1, 2, 3) - (3.5, 4, 4.5) \Rightarrow T_2 = (-3.5, -2, -0.5) \Rightarrow T_2 = 0$

$T_3 = c_3 - d_3 = (6.5, 7.5, 8.5) - (2, 5, 5.5) \Rightarrow T_3 = (1, 2.5, 6.5)$

$T_4 = c_4 - d_4 = (0.5, 1, 1.5) - (3, 4, 4.5) \Rightarrow T_4 = (-4, -3, -1.5) \Rightarrow T_4 = 0$

Now the objective function value (Z) of this solution can be calculated as follow:

$Z = \sum T_i = (4.5, 6, 8) + (1, 2.5, 6.5) = (5.5, 8.5, 14.5)$

$EV(Z) = \frac{5.5+(2\times8.5)+14.5}{4} \Rightarrow EV(Z) = 9.25$

5.2. Ranking population in each generation

The ranking population of each generation must be carried out to select the population member in each generation with better objective function for crossover and mutation operation and also selecting the best populations of each generation and transmitting them to the next generation. Given that the objective value of the corresponding population member is calculated through triangular fuzzy numbers, an approach for ranking triangular fuzzy numbers should be presented. In this paper, a method of ranking triangular fuzzy numbers with three criteria is proposed. These criteria must be applied respectively. In other words, if some fuzzy numbers are not ranked according to the first criterion, second and third criteria are applied, respectively [43]. These criteria are presented in the following:

- The first criterion for ranking fuzzy numbers: triangular fuzzy number $\tilde{A}$ is less than triangular fuzzy number $\tilde{B}$ if:

$$\frac{a_1+2a_2+a_3}{4} < \frac{b_1+2b_2+b_3}{4} \quad (5.5)$$

- The second criterion for ranking fuzzy numbers (mode): After ranking fuzzy numbers by the first criterion, those that remain can be ranked by the mode criterion. According to the second criterion, triangular fuzzy number $\widetilde{A}$ is less than triangular fuzzy number $\widetilde{B}$ if:

$$a_2 < b_2 \quad (5.6)$$

- The third criterion for ranking fuzzy numbers (range): After ranking fuzzy numbers by the first and second criteria, those that remain can be ranked by this criterion. This criterion applies a range of numbers for ranking them. According to this criterion, triangular fuzzy number $\widetilde{A}$ is less than triangular fuzzy number $\widetilde{B}$ if:

$$a_3 - a_1 < b_3 - b_1 \quad (5.7)$$

5.3. Crossover operation

In this paper, the proposed crossover operation is PMX which generates children with crossover rate p_c. Hence, two chromosomes are selected as parent 1 and 2 according to the strategy of parent selection. According to this strategy, two random solutions are selected and the solution with better rank is nominated as a parent. If the ranks of the two solutions are the same, one of them is nominated randomly. Then, two genes are selected randomly according to PMX operation and the genes between the selected genes of chromosome 1 and 2 are replaced; other genes are replaced or transmitted to the child without any change. To illustrate this operator, an instance with 10 jobs and 3 machines is considered. First, two cells are selected randomly and numbers between these cells from parent 1 are transferred to the child chromosome (Figure 3-Chromosome 1). The following relationship is established for genes between cell 3 and 9: 3 ↔ 6, 4 ↔ 2, 7 ↔ 7, 10 ↔ 8. Jobs {8, 7, 2, 4, 6} are copied to child 1. As a result, jobs {1, 3, 5, 10, 9} remain. By eliminating the jobs that were in the corresponding relationship, jobs {1, 5, 9} remain which are copied to the child according to their order in parent 2 (Figure 3- chromosome 2). Finally, since the remaining jobs {10, 3} are corresponded to jobs {8, 6} respectively, the corresponding numbers are copied to child genes according to their corresponding value in parent 2.

			↓						↓	
Parent 1	1	3	5	6	4	2	7	8	10	9
Parent 2	8	6	5	3	2	4	7	10	9	1
Chromosom1				6	4	2	7	8		
Chromosom2			5	6	4	2	7	8	9	1
Child	10	3	5	6	4	2	7	8	9	1

Figure 3. Crossover operation according to PMX operator

5.4. Mutation operation

In this paper, swamp mutation is applied with mutation rate p_m. According to this method, a random chromosome is nominated and two cells are selected for the mutation purpose. Finally, the numbers in these two cells are swamped. Figure 4 shows the process of swamp mutation assuming that cell 4 and cell 7 are selected for mutation.

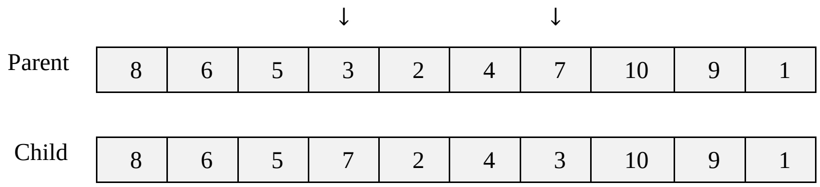


Figure 4. Swamp mutation

The flowchart of the proposed genetic algorithm is shown in Figure 5.

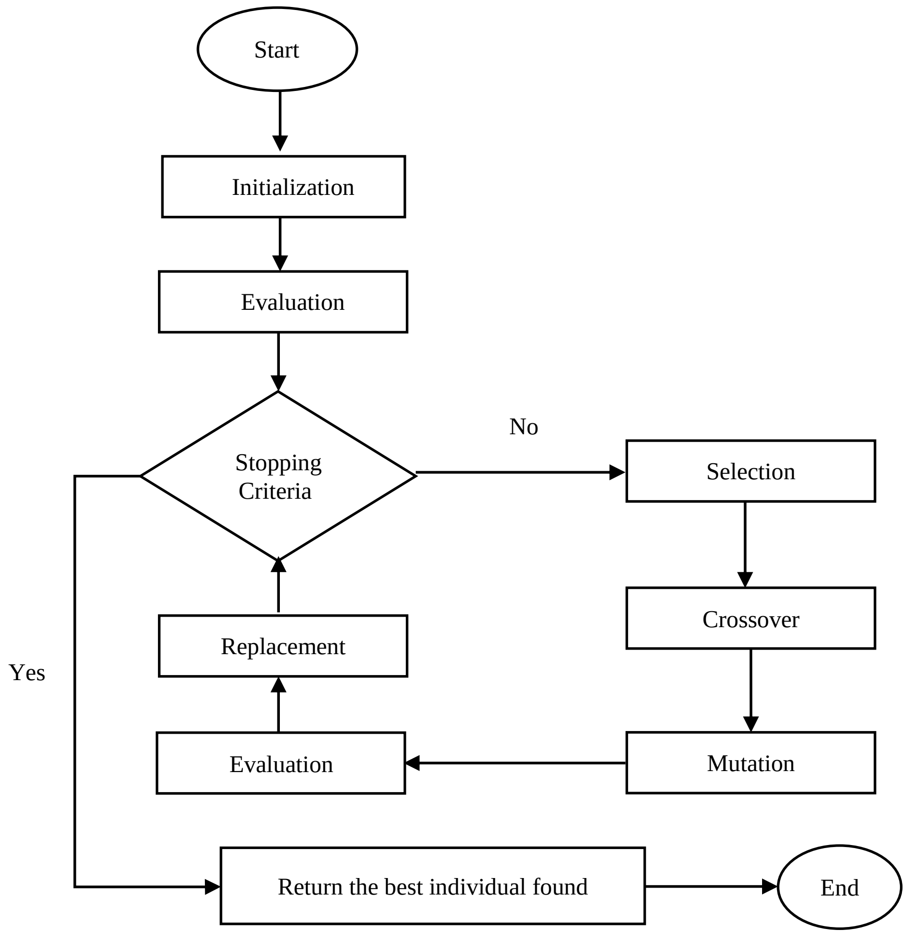


Figure 5. Genetic algorithm flowchart

6. Computational results

In this section, the performance of the mathematical model and the proposed algorithm are evaluated. Hence, some instances of different sizes are generated and are solved by deterministic mathematical model and genetic algorithm. Since GA calculates according to fuzzy parameters, the objective values are triangular fuzzy numbers. As a result, the output of the algorithm is evaluated and calculated for each instance according to Eq. (3.10).

In this study, the mathematical model is solved with GAMS software and the CPLEX solver is used. In this software, the maximum solution time for each instance is 7200 seconds and if the optimal solution is obtained during this period, it is reported. On the other hand, the Genetic algorithm is coded in C# programming language and performed all the computational experiments on a Laptop with an Intel Corei2 with 2.6 GHz CPU and 4GB RAM. To adjust the parameters of the GA, different values were considered for each parameter, and then, by solving some numerical problems in the initial experiments, the appropriate values were selected which are shown in Table 1 (n indicates the number of jobs).

Table 1. GA parameters

Value	Parameter
10 × n	Population size
80	Generation. no
0.8	Crossover rate
0.2	Mutation rate

In the proposed algorithm, if the number of algorithm repetitions reaches a specific number, GA stops and presents the best solution. In this paper, the specific number is 80 according to computational experiments.

In this research, the initial population for the genetic algorithm is generated randomly and due to the random nature of the proposed algorithm, each of the generated problems has been solved 5 times by the GA and the minimum, average and maximum values of the objective function for obtained solutions have been reported. Also, to facilitate the comparison of the results, the following relative percentage deviation has been used to measure the value of the objective function of each solution (sol) for a given problem:

$$RPD = \left(\frac{f(sol)-f(sol_{best})}{f(sol_{best})}\right) \times 100 \quad (6.1)$$

Where sol_{best} is the best solution for that problem among the obtained solutions. The RPD indicator compares each solution with the best solution obtained for that problem and declares their difference as a percentage; the lower the value of this indicator, the better the quality of the solution.

A number of instances are generated in different sizes to evaluate the efficiency of the mathematical model and the proposed algorithm. Process times are triangular fuzzy numbers $\left(p_{jk} = \left(p_{jk}^1 , p_{jk}^2 , p_{jk}^3\right)\right)$ in which p_{jk}^1, p_{jk}^2 and p_{jk}^3 are selected from a discrete uniform distribution within intervals ([15, 24], [25, 44], [45, 64]), respectively. Due dates are also triangular fuzzy numbers $\left(d_j = \left(d_j^1 , d_j^2 , d_j^3\right)\right)$ in which d_j^1, d_j^2 and d_j^3 are selected from a discrete uniform distribution within intervals ([25, 45], [46, 75], [76, 120]), respectively. u_{kj} are within the interval [0, 1] so that each job can be processed on at least half of the machines. Moreover, setup times, times between two consecutive maintenance activities (T), and duration of maintenance activity (t) are within intervals [1, 10], [50, 65], and [10, 15], respectively. The planning horizon is calculated by the following equation:

$$H = 5 \times \frac{\sum_j\left(\max_k(p_{jk}^3 \times u_{kj})\right)}{m} \quad (6.2)$$

Where p_{jk}^3 is the maximum process time of job j on machine k and u_{kj} is the probability of processing job j on machine k. It should be mentioned that 3 instances are generated for each problem size.

As shown in table 2, the results obtained from solving different sizes of the problem are reported, in which the RPD values for the problems that the mathematical model was able to find the optimal solution are bolded. Average results are also reported for instances of the same size. Table 2 indicates that for those problems that the mathematical model can find optimal solutions (11 instances out of 42 instances), GA is also able to find the optimal or near-optimal solution at least one time in five-time running. According to computational experiments, GA is superior to the proposed mathematical model. For the instances with 40 jobs and 7 machines, GAMS is not able to find any feasible solution within 7200 seconds. The remarkable point of GA is the low deviation of its presented solutions for each instance and this confirms the convergence of the algorithm. In general, according to the results, it can be claimed that the proposed GA algorithm has a great performance and can obtain good solutions in an acceptable time, especially for large instances of the problem.

Table 2. Computational results for different instances

Genetic Algorithm				Mathematical model				
Run time (sec)	Output (RPD)			Run time (sec)	Output RPD	Instance	NO. machines	NO. job
	Max	Avg	Min					
22	**0**	**0**	**0**	12	**0**	1		
19	**0**	**0**	**0**	16	**0**	2	2	5
26	0.1	0.07	**0**	18	**0**	3		
22.33	0.03	0.02	0	15.33	0		Average	
53	0.22	0.11	**0**	196	**0**	1		
38	0.19	0.06	**0**	211	**0**	2	2	7
48	0.20	0.09	**0**	268	**0**	3		
46.33	0.20	0.08	0	225	0		Average	
71	0.14	0.08	**0**	1081	**0**	1		
68	0.19	0.11	**0**	2011	**0**	2	2	8
94	0.32	0.16	**0**	1979	**0**	3		
77.66	0.21	0.11	0	1690.3	0		Average	
114	0.23	0.14	0.05	3876	**0**	1		
122	0.34	0.21	0.07	4391	**0**	2	2	9
139	0.26	0.16	0.07	7200	0	3		
125	0.27	0.17	0.06	5155.6	0		Average	
141	0.30	0.19	0.08	7200	0	1		
155	0.29	0.17	0.06	7200	0	2	3	10
137	0.31	0.18	0.05	7200	0	3		
144.33	0.30	0.18	0.06	7200	0		Average	
149	0.38	0.25	0	7200	0	1		
178	0.42	0.28	0.05	7200	0	2	3	11
165	0.33	0.24	0	7200	0	3		
164	0.37	0.25	0.01	7200	0		Average	
189	0.43	0.22	0	7200	0.26	1		
175	0.39	0.21	0	7200	0.22	2	3	12
209	0.52	0.34	0	7200	0.18	3		

191	0.44	0.25	0	7200	0.22		Average	
265	0.56	0.32	0	7200	0.84	1		
272	0.78	0.37	0	7200	0.79	2	3	15
304	0.93	0.5	0	7200	0.96	3		
280.33	0.75	0.39	0	7200	0.86		Average	
349	1.06	0.57	0	7200	1.82	1		
381	1.26	0.72	0	7200	2.19	2	4	18
396	1.38	0.78	0	7200	2.67	3		
375.33	1.23	0.69	0	7200	2.22		Average	
428	1.41	0.78	0	7200	2.86	1		
413	1.29	0.54	0	7200	2.71	2	4	20
422	1.32	0.68	0	7200	2.8	3		
421	1.34	0.66	0	7200	2.79		Average	
479	1.39	0.68	0	7200	3.78	1		
461	1.54	0.72	0	7200	3.89	2	5	25
507	1.87	0.96	0	7200	4.15	3		
482.33	1.60	0.78	0	7200	3.94		Average	
566	2.43	1.61	0	7200	4.28	1		
617	2.24	1.31	0	7200	4.51	2	6	30
642	1.93	1.19	0	7200	4.76	3		
608.3	2.20	1.37	0	7200	4.51		Average	
656	2.53	1.65	0	7200	5.28	1		
672	2.69	1.75	0	7200	5.76	2	6	35
669	2.33	1.48	0	7200	5.45	3		
665.6	2.51	1.62	0	7200	5.49		Average	
811	2.83	1.82	0	7200	-	1		
763	2.92	1.88	0	7200	-	2	7	40
783	2.76	1.71	0	7200	-	3		
785.6	2.83	1.8	0	7200	-		Average	

7. Conclusions and future research

This paper addressed an unrelated parallel-machine scheduling problem in which process times and due dates are fuzzy numbers. Due to maintenance activities, machines become unavailable periodically and each machine can process a specific set of jobs. In addition, jobs become available at specific times, and setup times depend on machine type and job sequence. One of the important applications of this issue is in the production of products that the due dates are defined as a time window and the best due date value is close to the middle of the time window and the jobs processing times depend on other factors such as operator and their value is not specified and are announced as interval under uncertainty. In this study, a fuzzy mathematical model was presented and then, the problem was changed to a deterministic programming model. Due to the inability of the exact method (GAMS) to solve large instances of the problem, a fuzzy-based genetic algorithm was developed. The proposed GA as well as the exact method presented good performance in solving small instances. In addition, it was capable to solve large instances and showing good performance according to quality and run time factors.

For future research, preemption can be applied for the same problem. Also, different objectives can be added to the problem and various multi-objective meta-heuristics may be applied to solve these problems.

References

K. Sun and H. Li, (2010). Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines. *Int. J. Prod. Econ*, vol. 124, no. 1, pp. 151–158.

C. Zhao, M. Ji, and H. Tang, (2011). Parallel-machine scheduling with an availability constraint. *Comput. Ind. Eng,* vol. 61, no. 3, pp. 778–781.

Z. Tan, Y. Chen, and A. Zhang, (2013). On the exact bounds of SPT for scheduling on parallel machines with availability constraints. *Int. J. Prod. Econ,* vol. 146, no. 1, pp. 293–299.

D. Xu and D.-L. Yang, (2013). Makespan minimization for two parallel machines scheduling with a periodic availability constraint: mathematical programming model, average-case analysis, and anomalies. *Appl. Math. Model*, vol. 37, no. 14, pp. 7561–7567.

X. Wang and T. C. E. Cheng, (2015). A heuristic for scheduling jobs on two identical parallel machines with a machine availability constraint, *Int. J. Prod. Econ*, vol. 161, pp. 74–82.

J.-Y. Lee and Y.-D. Kim, (2015). A branch and bound algorithm to minimize total tardiness of jobs in a two identical-parallel-machine scheduling problem with a machine availability constraint. *J. Oper. Res. Soc*, vol. 66, no. 9, pp. 1542–1554.

J. He, Q. Li, and D. Xu, (2016). Scheduling two parallel machines with machine-dependent availabilities. *Comput. Oper. Res,* vol. 72, pp. 31–42.

L. Shen, D. Wang, and X.-Y. Wang, (2013). Parallel-machine scheduling with non-simultaneous machine available time. *Appl. Math. Model,* vol. 37, no. 7, pp. 5227–5232.

Y. Huo and H. Zhao, (2015). Total completion time minimization on multiple machines subject to machine availability and makespan constraints. *Eur. J. Oper. Res*, vol. 243, no. 2, pp. 547–554.

W.-C. Lee, J.-Y. Wang, and L.-Y. Lee, (2015). A hybrid genetic algorithm for an identical parallel-machine problem with maintenance activity. *J. Oper. Res. Soc*, vol. 66, no. 11, pp. 1906–1918.

Y. Yin, Y. Wang, T. C. E. Cheng, (2017). W. Liu, and J. Li, Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. *Omega*, vol. 69, pp. 17–28.

V. Suresh and D. GHAUDHURI, (1996). Scheduling of unrelated parallel machines when machine availability is specified. *Prod. Plan. Control*, vol. 7, no. 4, pp. 393–400.

D. Jiang and J. Tan, (2016). Scheduling with job rejection and no simultaneous machine available time on unrelated parallel machines. *Theor. Comput. Sci*, vol. 616, pp. 94–99.

F. Ahmadizar, K. Mahdavi, and J. Arkat, (2019). Unrelated parallel machine scheduling with processing constraints and sequence dependent setup times. *Adv. Ind. Eng*, vol. 53, no. 1, pp. 495–507.

O. Avalos-Rosales, F. Angel-Bello, A. Álvarez, and Y. Cardona-Valdés, (2018). Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Comput. Ind. Eng*, vol. 123, pp. 364–377.

S. Lu, X. Liu, J. Pei, M. T. Thai, and P. M. Pardalos, (2018). A hybrid ABC-TS algorithm for the unrelated parallel-batching machines scheduling problem with deteriorating jobs and maintenance activity. *Appl. Soft Comput*, vol. 66, pp. 168–182.

S. Balin, (2011). Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. *Inf. Sci,* vol. 181, no. 17, pp. 3551–3569.

C. C. Chyu and W. S. Chang, (2011). Optimizing fuzzy makespan and tardiness for unrelated parallel machine scheduling with archived metaheuristics. *Int. J. Adv. Manuf. Technol*, vol. 57, no. 5–8, pp. 763–776.

K. M. Senthilkumar, V. Selladural, K. Raja, and V. Thirunavukkarasu, (2011). A hybrid algorithm based on pso and aco approach for solving combinatorial fuzzy unrelated parallel machine scheduling problem.

P. Alcan and H. Balişgil, (2012). A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem. *Adv. Eng. Softw*, vol. 45, no. 1, pp. 272–280.

S. A. Torabi, N. Sahebjamnia, S. A. Mansouri, and M. A. Bajestani, (2013). A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem. *Appl. Soft Comput,* vol. 13, no. 12, pp. 4750–4762.

J. Behnamian, (2014). Particle swarm optimization-based algorithm for fuzzy parallel machine scheduling. *Int. J. Adv. Manuf. Technol*, vol. 75, no. 5–8, pp. 883–895.

W. C. Yeh, P. J. Lai, W. C. Lee, and M. C. Chuang, (2014). Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects. *Inf. Sci*., vol. 269, pp. 142–158.

M. Naderi-Beni, E. Ghobadian, S. Ebrahimnejad, and R. Tavakkoli-Moghaddam, (2014). Fuzzy bi-objective formulation for a parallel machine scheduling problem with machine eligibility restrictions and sequence-dependent setup times. *Int. J. Prod. Res*, vol. 52, no. 19, pp. 5799–5822.

M. Rostami, A. E. Pilerood, and M. M. Mazdeh, (2015). Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. *Comput. Ind. Eng*, vol. 85, pp. 206–215.

K. K. Nailwal, D. Gupta, and S. Sharma, (2015). Fuzzy bi-criteria scheduling on parallel machines involving weighted flow time and maximum tardiness. *Cogent Math*, vol. 2, no. 1, p. 1019792.

T. W. Liao and P. Su, (2017). Parallel machine scheduling in fuzzy environment with hybrid ant colony optimization including a comparison of fuzzy number ranking methods in consideration of spread of fuzziness. *Appl. Soft Comput. J*., vol. 56, pp. 65–81.

K. Li, J. Chen, H. Fu, Z. Jia, and W. Fu, (2019). Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint. *Appl. Soft Comput*, p. 105585.

Z. Jia, J. Yan, J. Y. T. Leung, K. Li, and H. Chen, (2019). Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities. *Appl. Soft Comput*, vol. 75, pp. 548–561.

J. Rezaeian, S. Mohammad-Hosseini, S. Zabihzadeh, and K. Shokoufi, (2020).Fuzzy Scheduling Problem on Unrelated Parallel Machine in JIT Production System. *Artif. Intell. Evol*., pp. 17–33.

X. Liu, F. Chu, F. Zheng, C. Chu, and M. Liu, (2020). Parallel machine scheduling with stochastic release times and processing times. *Int. J. Prod. Res*., pp. 1–20.

K. Li, J. Chen, H. Fu, Z. Jia, and J. Wu, (2020). Parallel machine scheduling with position-based deterioration and learning effects in an uncertain manufacturing system. *Comput. Ind. Eng*., vol. 149, p. 106858.

O. A. Arık, M. Schutten, and E. Topan, (2021). Weighted Earliness/Tardiness Parallel Machine Scheduling Problem with a Common Due Date. *Expert Syst. Appl*., p. 115916.

H. Khalifa, (2020). On single machine scheduling problem with distinct due dates under fuzzy environment. *Int. J. Supply Oper. Manag,* vol. 7, no. 3, pp. 272–278.

S. Midya, S. Kumar Roy, and G. Wilhelm Weber, (2021). Fuzzy multiple objective fractional optimization in rough approximation and its aptness to the fixed-charge transportation problem. *RAIRO--Operations Res*, vol. 55, no. 3.

S. Midya, S. K. Roy, and F. Y. Vincent, (2021). Intuitionistic fuzzy multi-stage multi-objective fixed-charge solid transportation problem in a green supply chain. *Int. J. Mach. Learn. Cybern*, vol. 12, no. 3, pp. 699–717.

S. K. Roy and S. Midya, (2019). Multi-objective fixed-charge solid transportation problem with product blending under intuitionistic fuzzy environment. *Appl. Intell*, vol. 49, no. 10, pp. 3524–3538.

A.Mondal, S. K. Roy, and S. Midya, (2021). Intuitionistic fuzzy sustainable multi-objective multi-item multi-choice step fixed-charge solid transportation problem. *J. Ambient Intell. Humaniz. Comput*, pp. 1–25.

S. K. Roy, S. Midya, and G.-W. Weber, (2019). Multi-objective multi-item fixed-charge solid transportation problem under twofold uncertainty. *Neural Comput. Appl*, vol. 31, no. 12, pp. 8593–8613.

E. B. Tirkolaee, A. Goli, and G.-W. Weber, (2020). Fuzzy mathematical programming and self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shop scheduling problem with outsourcing option. *IEEE Trans. fuzzy Syst*, vol. 28, no. 11, pp. 2772–2783.

M. Saffarian, M. Niksirat, and S. M. Kazemi, (2021). A Hybrid Genetic-Simulated Annealing-Auction Algorithm for a Fully Fuzzy Multi-Period Multi-Depot Vehicle Routing Problem. *Int. J. Supply Oper. Manag*, vol. 8, no. 2, pp. 96–113.

S. Heilpern, (1992). The expected value of a fuzzy number. *Fuzzy Sets Syst*, vol. 47, no. 1, pp. 81–86.

M. Sakawa and R. Kubota, (2000). Fuzzy programming for multi objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *Eur. J. Oper. Res*, vol. 120, no. 2, pp. 393–407.