RAIRO-Oper. Res. 55 (2021) S1515–S1528
https://doi.org/10.1051/ro/2020121

RAIRO Operations Research
www.rairo-ro.org

FUZZY GILMORE AND GOMORY ALGORITHM: APPLICATION IN ROBOTIC FLOW SHOPS WITH THE EFFECTS OF JOB-DEPENDENT TRANSPORTATION AND SET-UPS

Shahabeddin Sotudian, Ali Akbar Sadat Asl* and Mohammad Hossein Fazel Zarandi

Abstract. This paper addresses the scheduling of robotic cells with job-dependent transportation and set-up effects with fuzzy methodology. Since transportation and set-up times are a large portion of the production time in a flexible manufacturing cell, ignoring these parameters may cause significant errors in determining the optimal makespan. Furthermore, determining the exact values of these time parameters is a challenging task. To overcome this problem, we represent these parameters using fuzzy L-R numbers. Using the capability of fuzzy numbers to represent approximate values, we can represent these parameters without losing valuable information. For generating the optimal part sequencing in the cells, the Gilmore and Gomory algorithm is modified, and instead, a fuzzy Gilmore and Gomory algorithm is developed. We compare the results of the proposed fuzzy method with those of crisp ones. The results indicate the superiority of the proposed algorithm in terms of robustness, flexibility, and reduction of makespan.

Mathematics Subject Classification. 65L05, 03E72, 34K28.

Received August 21, 2019. Accepted October 24, 2020.

1. Introduction

The last decade has seen numerous studies focusing on mathematical optimization approaches for different scheduling problems [1, 10]. In particular, special attention has been paid to Robotic flow shop systems in recent years. Robotic flow shop systems are comprised of a set of robots that are responsible for transferring jobs between pairs of successive machines. These systems broadly appear in automatic manufacturing systems. As manufacturers implement more intricate robotic cells, more complex models and algorithms are needed to optimize the operations of these cells [9].

The origin of cellular manufacturing can be traced back to efforts aimed at mixing the efficiency of product layouts with the flexibility of the process [20]. Nowadays, the application of robots in manufacturing systems is becoming an interesting area of research. A robotic cell consists of an input station, several machines arranged in series, an output station, and one or more robots for handling the parts between the stations and machines [6]. Since the robotic cell scheduling problem is an NP-hard problem, enumeration of all possible solutions is compu-

Keywords. Fuzzy flow shop scheduling, robotic cells, fuzzy L-R numbers, Gilmore and Gomory algorithm.

Department of Industrial and Systems Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.
*Corresponding author: a.sadatasl@aut.ac.ir

tationally expensive [2]. Thus, various optimization approaches such as exact optimization, heuristics, and metaheuristics have been developed to solve these problems. There have been several studies based on the exact optimization approaches like mixed integer programming (see [15]), but these approaches are not appropriate for large-scale problems due to their exponential time requirements.

Apart from the exact optimization methods, some researchers (see [14]) proposed heuristic algorithms for the cyclic robotic flow shop scheduling problems. Moreover, metaheuristic algorithms are robust and adaptive search optimization methods and can be used to generate optimal or near-optimal solutions for large-scale scheduling problems in reasonable computational times. Li *et al.* [18] extended the classical flow shop scheduling problem to the distributed permutation flow shop scheduling problem (DPFSP). The DPFSP extends the classic permutation flow shop scheduling problem (PFSP) by introducing α identical factories, each of which possesses a flow shop layout with β machines. Li *et al.* assume the robot must transfer from the predecessor machine to the successor machine. They proposed an improved iterated greedy algorithm where each factory has only one robot and the aim is to minimize the makespan. Specifically, their algorithm generates four types of neighborhood structures. Then, a simulated annealing algorithm is embedded to improve exploration abilities.

The objective of minimizing makespan in the two-machine flow shop scheduling problem is also known as Johnson's problem. The results obtained by Johnson are among the very first formal results in the theory of scheduling [4]. Some researchers (see [13]) have been developed various modifications of the latter problem by considering the transportation and set-up times.

Flow shop scheduling model with no work-in-process storage (no-WIP-storage) has been investigated by researchers with two basic modifications. The first one does not include input/output automated storage-and-retrieval stations (AS/RS). This modification has been studied by Levner [16] and Stern *et al.* [22] who have considered its relations with both the Johnson problem and the traveling salesman problem (TSP). In the second modification of the no-WIP-storage scheduling model, AS/R stations are considered in the model. This modification proposed by Kise *et al.* [12] in 1991. They considered the automated two-machine flow shop scheduling problems with no buffer storage for WIP. Also, they proposed $O(n^3)$ time algorithms based on the well-known Gilmore–Gomory algorithm to obtain optimal schedules that minimize the maximum completion time. Levner *et al.* studied the two-machine, no-WIP-storage robotic cell with AS/RS problem [17]. They introduced several AS/RS and demonstrated that the resultant flow shop scheduling problem with job-dependent transportation times can exactly solve in $O(n^3\log(n))$ time using a modified Gilmore–Gomory algorithm.

All of the above-mentioned studies assume certainty in all aspects of the problem. Nevertheless, in real-world problems, we encounter uncertainty of information regarding time parameters of jobs, traveling times of robot between machines, and makespans of different schedules. Fuzzy logic has the ability of modeling uncertainty, vagueness, and imprecision that exist in the majority of real-world problems. Consequently, it has found successful applications in a wide variety of fields such as healthcare [23], banking industry [25], facility location-network design [21] and so on. It also has proven its capability and robustness in the flow shop scheduling problems. Tirkolaee *et al.* [3] proposed a bi-objective mixed-Integer linear programming model with an outsourcing option and Just-in-Time delivery. They considered two objectives of total cost minimization and total energy consumption minimization. In their study, a hybrid methodology is developed based on an interactive fuzzy solution method and a self-adaptive artificial fish swarm algorithm. Their results show the effectiveness and robustness of the fuzzy methodology.

In the current study, we use fuzzy logic to make this robotic scheduling problem more realistic and robust. The real-world scheduling problems usually contain various forms of uncertainty. Fuzzy set theory as a powerful tool can model these uncertainties. This paper aims to introduce a fuzzy flow shop scheduling of robotic cells with job-dependent transportation and set-up effects and to present an efficient solution method. The most important contribution of the current paper is the fuzzy Gilmore and Gomory algorithm. Moreover, we generalize the model of Levner *et al.* into a fuzzy flow shop scheduling problem to demonstrate that better performance is achievable by incorporating fuzzy logic in the model.

Based on the above discussion, the rest of the paper is organized as follows: the related fuzzy concepts, the problem definitions, notations, and formulations are presented in Section 2. Moreover, this section describes

the steps of the proposed fuzzy Gilmore and Gomory algorithm. In Section 3, numerical examples are presented and the results of fuzzy and crisp makespans will be compared. Section 4 is devoted to discussions. Finally, conclusions and future work are presented in Section 5.

2. Materials and methods

2.1. Fuzzy concepts

An L-R fuzzy number $C = (h, k, \sigma, \beta)_{LR},\ h \leq k$, is defined as follows [19]:

$$\mu_C(x) = \begin{cases} L\left(\frac{h-x}{\sigma}\right) & -\infty < x < h, \\ 1 & h < x < k, \\ R\left(\frac{x-k}{\beta}\right) & k < x < +\infty, \end{cases}$$

where σ and β are the left-hand and right-hand spreads. $L\left(\frac{h-x}{\sigma}\right)$ and $R\left(\frac{x-k}{\beta}\right)$ are non-increasing functions with $L(0) = 1$ and $R(0) = 1$, respectively.

Here, we use L-R fuzzy numbers since their definition is very general and allows quantification of quite different types of information. In practice, some special L-R fuzzy numbers like the triangular fuzzy number, the Gaussian fuzzy number, and the trapezoidal fuzzy number are widely used in many areas to deal with various vague information. For example, a fuzzy number Ω is called a Gaussian fuzzy number whose membership function (MF) has the following characteristics:

$$\widetilde{\Omega}(x) = \begin{cases} \exp\left(-\frac{(x-\overline{x})^2}{2\sigma_L^2}\right), & \text{for} \quad x < \overline{x} \\ \exp\left(-\frac{(\overline{x}-x)^2}{2\sigma_R^2}\right), & \text{for} \quad x \geq \overline{x} \end{cases}$$

where σ_L and ó$_R$ donate the left-hand and right-hand spreads, corresponding to the standard deviation of the Gaussian distribution. A Gaussian fuzzy number can be denoted by $\widetilde{\Omega} = (\overline{x}, \sigma_L, \sigma_R)$.

Moreover, as a part of our algorithm we need to sort L-R fuzzy numbers. To that end, we use the following method proposed by Nasibov *et al.* [19]. Let Y denotes the space of L-R fuzzy numbers. Then, the decomposition representation of a fuzzy number $\lambda \in Y$ (L-R representation) has the following form:

$$\lambda = \bigcup_{\alpha \in (0,1]} (\alpha, [L_\lambda(\alpha), R_\lambda(\alpha)]),$$

where $L(\alpha) = \mu_{\uparrow}^{-1}(\alpha)$ and $R(\alpha) = \mu_{\downarrow}^{-1}(\alpha)$ denote quasi-inverse functions of the increasing and decreasing parts of the membership functions $\mu(t)$, respectively. The weighted averaging based on levels (WABL) value of the fuzzy number λ can be defined as follows:

$$I(\lambda) = \int_0^1 (c_L L_\lambda(\alpha) + c_R R_\lambda(\alpha)) p(\alpha)\, \mathrm{d}\alpha, \tag{2.1}$$

where the parameters c_L and c_R are the "pessimism/optimism" parameters while the function $p(\alpha)$ is the distribution function of the importance of the level sets. Furthermore, the latter satisfies the following conditions:

$$\begin{aligned} c_L &\geq 0 \\ c_R &\geq 0 \\ c_L + c_R &= 1 \\ \int_0^1 p(\alpha)\, \mathrm{d}\alpha &= 1. \end{aligned}$$

Here, $p(\alpha)$ is called WABL strategy parameters and can be defined as follows:

$$p(\alpha) = (d+1)\,\alpha^d,$$

where $d > 0$ is a parameter. Finally, for any two L-R fuzzy numbers λ and β, the ranking order by $I(.)$ is determined based on the following rules:

$$\begin{cases} I(\lambda) > I(\beta) & \text{iff} \quad \lambda \succ \beta \\ I(\lambda) < I(\beta) & \text{iff} \quad \lambda \prec \beta \\ I(\lambda) = I(\beta) & \text{iff} \quad \lambda \sim \beta \end{cases}.$$

Additionally, to calculate the fuzzy makespan and fuzzy waiting times, we use α-cut of fuzzy sets for addition, subtraction, minimization, and maximization operations. Moreover, for defuzzification of fuzzy results, the BADD defuzzification method is used. For further details, please refer to [7, 11, 24].

2.2. Problem definition

Levner *et al.* proposed the flow shop scheduling of robotic cells with job-dependent transportation and set-up effects. We generalize their model into fuzzy flow shop scheduling of robotic cells with job-dependent transportation and set-up effects. Moreover, we also generalize their crisp notation into fuzzy ones. Then, to solve this problem, we proposed a fuzzified version of Gilmore and Gomory algorithm. In fact, we just use the Levner's problem as an example of problems that can be improved using our algorithm. A general description of the Levner's problem can be summarized as follows:

Consider a robotic cell comprises of two machines, several inputs AS/RS, and a single output AS/RS. A given set X of n jobs is to be processed on the machines M_1 and M_2. A job is loaded at one of the input stations and unloaded at the output station after its processing. Transportations between the input/output stations and every machine, and between two machines as well as loading/unloading jobs are performed by a transporting robot. The main assumptions in the model are as follows:

- No machine has buffer storage for work-in-process.
- Robot can transport only one job at a time.
- Each machine can process at most one job at a time.
- Robot is not allowed to interrupt processing once it has started.

The objective is to determine in which order the jobs should be processed to minimize the makespan. Consider a sequence $S = (s_1, s_2, \ldots, s_n)$ of jobs from X to be processed on the machines. These jobs have to process in conformity with the following technological principles [17].

2.2.1. Technological Principle 1

The first job, s_1, is loaded on the robot at the AS/RS station, and transferred to the machine M_1. Then, it is loaded on the machine M_1. At the same time with the above transportation and loading, a tool is set on the machine M_1 for performing s_1. After that, the machine M_1 starts processing s_1, while the robot waits there until it is finished. Then, the robot unloads it from M_1, transports, and loads it on the machine M_2. Concurrently, a tool is set on the machine M_2 for performing s_1. Now, set $r := 1$ and apply Technological Principle 2.

2.2.2. Technological Principle 2

Job s_r is launched on the machine M_2 and begins its processing. At the same time, an empty robot moves to the input AS/RS station where job s_{r+1} is stored. The robot loads job s_{r+1} and transports it to the machine M_1, and eventually loads it on the machine M_1. Concurrently with these processes, a tool on the machine M_1 is replaced for performing job s_{r+1}, if needed. Next, job s_{r+1} is processed on machine M_1 and at the same time, an empty robot moves to the machine M_2, wait there until job s_r is accomplished on the machine M_2 and unloads it. Then, the job s_r is transferred to the output AS/RS and it is unloaded there. After that, the empty

robot moves to the machine M_1, wait there if a job s_{r+1} is not finished on M_1, and unloads it from M_1.Then, the robot transfers and loads job s_{r+1} on the machine M_2. Concurrently with these processes, a tool on the machine M_2 is replaced for performing s_{r+1}, if needed. Now, set $r := r + 1$, and, if $r < n$, apply Technological Principle 2. If $r = n$, apply Technological Principle 3.

2.2.3. Technological Principle 3

The machine M_2 starts processing job s_n. An empty robot waits at the machine M_2 until the last job is finished. The robot unloads s_n and transfers it to the output AS/RS, where it unloads. Stop.

2.3. Notations

The mathematical notations that will be used to formulate our model are as follows:

$X = [1, 2, \ldots, n]$	Set of n jobs to be processed.
$\widetilde{\mathcal{L}}_{\mathrm{I}}(j)$	Fuzzy time is taken by a robot to load job j at the input AS/RS.
$\widetilde{\mathcal{T}}_{I,M_1}(j)$	Fuzzy transportation time is taken by a robot to transfer job j from its input AS/RS to M_1.
$\widetilde{\mathcal{L}}_{M_1}(j)$ and $\widetilde{\mathcal{L}}_{M_2}(j)$	Fuzzy time is taken by a robot to load job j on M_1 and M_2, respectively.
$\widetilde{\mathcal{Q}}_{M_1}(j)$ and $\widetilde{\mathcal{Q}}_{M_2}(j)$	Fuzzy times needed to set a tool for processing job j on M_1 and M_2, respectively.
$\widetilde{\mathcal{P}}_{M_1}(j)$ and $\widetilde{\mathcal{P}}_{M_2}(j)$	Fuzzy processing times of job j on M_1 and M_2, respectively.
$\widetilde{\mathcal{U}}_{M_1}(j)$ and $\widetilde{\mathcal{U}}_{M_2}(j)$	Fuzzy time is taken by a robot to unload any job from M_1 and M_2, respectively.
$\widetilde{\mathcal{U}}_O$	Fuzzy time is taken by a robot to unload any job at the output AS/RS.
$\widetilde{\mathcal{T}}_{M_1,M_2}$	Fuzzy transportation time taken by a robot to deliver any job finished on M_1 for processing on M_2.
$\widetilde{\mathcal{T}}^e_{M_1,M_2}$	Fuzzy transportation time is taken by an empty robot to move from M_1 to M_2.
$\widetilde{\mathcal{T}}^e_{M_2,I}$	Fuzzy transportation time is taken by an empty robot to move from M_2 to the input AS/RS where job j is stored
$\widetilde{\mathcal{T}}_{M_2,O}$	Fuzzy transportation time is taken by a robot to deliver a finished job from M_2 to the output AS/RS.
$\widetilde{\mathcal{T}}^e_{O,M_1}$	Fuzzy transportation is taken by an empty robot to move from the output AS/RS to M_1, in order to serve a job finished at that machine.

2.4. Mathematical formulation of the model

The fuzzy time in order to complete all n jobs in sequence S can be calculated as follows:

$$\widetilde{\mathcal{M}}(S) = \tilde{z}(s_1) + \tilde{w}(s_1, \ldots, s_{n-1}) + \tilde{v}(s_n). \tag{2.2}$$

Equation (2.2) has three parts that represent the three technological principles. $\tilde{v}(s_n)$ is the fuzzy time to fulfill the final operations described in Technological Principle 3 and can be defined as follows:

$$\tilde{v}(s_n) = \widetilde{\mathcal{P}}_{M_2}(s_n) + \widetilde{\mathcal{U}}_{M_2} + \widetilde{\mathcal{T}}_{M_2,O} + \widetilde{\mathcal{U}}_O.$$

$\tilde{w}(s_1, \ldots, s_{n-1})$ is the fuzzy time to fulfill all the operations described in Technological Principle 2, for all r; it has the following form:

$$\begin{aligned}\tilde{w}(s_1, \ldots, s_{n-1}) = \sum_{r=1}^{n-1} & \max\left(\tilde{W}^1_{r+1}, \tilde{W}^2_{r+1}, \tilde{W}^3_r\right) \\ & + \widetilde{\mathcal{U}}_{M_1} + \max\left(\widetilde{\mathcal{T}}_{M_1,M_2} + \widetilde{\mathcal{L}}_{M_2}(s_{r+1}), \widetilde{\mathcal{Q}}_{M_2}(s_{r+1})\right),\end{aligned}$$

where $\tilde{W}_{r+1}^1$, $\tilde{W}_{r+1}^2$, and $\tilde{W}_r^3$ are as follows:

$$\begin{aligned}
\tilde{W}_{r+1}^1 &= \widetilde{\mathcal{T}}_{M_2,I}^e\left(s_{r+1}\right) + \widetilde{\mathcal{L}}_{\mathrm{I}}\left(s_{r+1}\right) + \widetilde{\mathcal{P}}_{M_1}\left(s_{r+1}\right) \\
&\quad + \max\left(\widetilde{\mathcal{T}}_{I,M_1}\left(s_{r+1}\right) + \widetilde{\mathcal{L}}_{M_1}\left(s_{r+1}\right), \widetilde{\mathcal{Q}}_{M_1}\left(s_{r+1}\right)\right) \\
\tilde{W}_{r+1}^2 &= \widetilde{\mathcal{T}}_{M_2,I}^e\left(s_{r+1}\right) + \widetilde{\mathcal{L}}_{\mathrm{I}}\left(s_{r+1}\right) \\
&\quad + \max\left(\widetilde{\mathcal{T}}_{I,M_1}\left(s_{r+1}\right) + \widetilde{\mathcal{L}}_{M_1}\left(s_{r+1}\right), \widetilde{\mathcal{Q}}_{M_1}\left(s_{r+1}\right)\right) \\
&\quad + \widetilde{\mathcal{T}}_{M_1,M_2}^e + \widetilde{\mathcal{U}}_{M_2} + \widetilde{\mathcal{T}}_{M_2,O} + \widetilde{\mathcal{U}}_O + \widetilde{\mathcal{T}}_{O,M_1}^e, \\
\tilde{W}_r^3 &= \widetilde{\mathcal{P}}_{M_2}\left(s_r\right) + \widetilde{\mathcal{U}}_{M_2} + \widetilde{\mathcal{T}}_{M_2,O} + \widetilde{\mathcal{U}}_O + \widetilde{\mathcal{T}}_{O,M_1}^e.
\end{aligned} \tag{2.3}$$

Finally, $\tilde{z}(s_1)$ is the fuzzy time to fulfill the final operations described in Technological Principle 1:

$$\begin{aligned}
\tilde{z}(s_1) &= \widetilde{\mathcal{L}}_{\mathrm{I}}(s_1) + \max\left(\widetilde{\mathcal{T}}_{I,M_1}(s_1) + \widetilde{\mathcal{L}}_{M_1}(s_1), \widetilde{\mathcal{Q}}_{M_1}(s_1)\right) \\
&\quad + \widetilde{\mathcal{P}}_{M_1}(s_1) + \widetilde{\mathcal{U}}_{M_1} + \max\left(\widetilde{\mathcal{T}}_{M_1,M_2} + \widetilde{\mathcal{L}}_{M_2}(s_1), \widetilde{\mathcal{Q}}_{M_2}(s_1)\right).
\end{aligned}$$

It has been proved that minimizing $\widetilde{\mathcal{M}}\left(S\right)$ is equivalent to finding the permutation minimizing the following function [17]:

$$\widetilde{\mathcal{M}}'\left(S\right) = \tilde{z}'(s_1) + \tilde{w}'\left(s_1, \ldots, s_{n-1}\right) + \tilde{v}'\left(s_n\right),$$

where

$$\begin{aligned}
\tilde{z}'(s_1) &= \widetilde{\mathcal{L}}_{\mathrm{I}}(s_1) + \widetilde{\mathcal{P}}_{M_1}(s_1) + \max\left(\widetilde{\mathcal{T}}_{I,M_1}(s_1) + \widetilde{\mathcal{L}}_{M_1}(s_1), \widetilde{\mathcal{Q}}_{M_1}(s_1)\right), \\
\tilde{w}'\left(s_1, \ldots, s_{n-1}\right) &= \sum_{r=1}^{n-1} \max\left(\tilde{W}_{r+1}^1, \tilde{W}_{r+1}^2, \tilde{W}_r^3\right), \\
\tilde{A}\left(s_r\right) &= \max\left(\tilde{W}_r^1, \tilde{W}_r^2\right), \tilde{B}\left(s_r\right) = \tilde{W}_r^3 \quad r = 1, \ldots, n \\
\tilde{v}'\left(s_n\right) &= \widetilde{\mathcal{P}}_{M_2}\left(s_n\right).
\end{aligned} \tag{2.4}$$

Eventually, this problem is reducible to a special case of the traveling salesman problem studied by Gilmore and Gomory [5,8]. The proof of this lemma is similar to the crisp problem. For more information, please refer to [17]. Thus, one evident way to minimize (2.3), is to calculate the value of following formula for each pair (s_1, s_n), and to choose the minimal among the resulting $n(n-1)$ values.

$$\tilde{z}'(s_1) + \min_s \sum_{r=1}^{n-1} \max\left(\tilde{A}\left(s_{r+1}\right), \tilde{B}\left(s_r\right)\right) + \tilde{v}'\left(s_n\right).$$

This problem can be solved in $O\left(n^3 \log\ n\right)$ time using the Gilmore and Gomory algorithm. The Gilmore and Gomory algorithm can be modified to be performed in essentially cubic time. In the next section, the modified fuzzy Gilmore and Gomory algorithm is presented for solving this problem in cubic time.

2.5. Modified fuzzy Gilmore and Gomory algorithm

To solve the above-mentioned fuzzy model, we need to fuzzify the modified Gilmore and Gomory algorithm proposed by Levner *et al.* [17]. Then, we can obtain the optimal makespan for the problem and the sequence of jobs that lead to it. For this purpose, we assume that the time parameters of all jobs on each machine are fuzzy L-R numbers.

The main reason behind this choice is the higher ability of fuzzy numbers to model uncertainties. In this specific problem, we need to define several time parameters for the model. However, we know that in a real-world problem, the values of these parameters cannot be determined exactly. Thus, if we use a crisp number for these parameters, we will lose valuable information. To overcome this issue, we define the parameters of this model using fuzzy L-R numbers. Now, the steps of the modified fuzzy Gilmore and Gomory algorithm can be defined as follows:

Step 1	Read n pairs $\left(\tilde{A}(i),\ \tilde{B}(i)\right)\ 1 \leq\ i\ \leq\ n$, defined by (2.4), and renumbered so that $\tilde{B}\,(1)\ \leq\ \tilde{B}\,(2)\ \leq\ \ldots\ \leq\ \tilde{B}(n)$, and an auxiliary pair of numbers, $(\tilde{A}(n\ +\ 1),\ \tilde{B}(n+1)) = (K,\ K)$, where $K \geq \max(\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n, \tilde{B}_1, \tilde{B}_2, \ldots, \tilde{B}_n)$.
Step 2	Sort the $\tilde{A}(i)$ values in the non-decreasing order using Equation (2.1). Find $\varphi\,(j)$ for all j. The permutation φ is defined by $\varphi\,(j) = q$, where j being such that $\tilde{A}(q)$ is the j^{th} smallest of the $\tilde{A}(i)$.
Step 3	For each pair of indices, i and j, $i\ \neq j,\ \ i,\ j = 1, \ldots, n$, we find "a partial sorting," φ_{ij},that is obtained from the sorting at Step 2 by excluding $\tilde{A}(\varphi(i))$ and $\tilde{A}(\varphi\,(j))$.
For all φ_{ij} **repeat** Steps 4–11	
Step 4	Compute the fuzzy cost of arc $(i, i\ +\ 1)$ with fuzzy operations as follows: $\tilde{C}_{i,i+1}\ =\ \max\left\{\tilde{0}, \left\{\min\left(\tilde{B}(i+1), \tilde{A}(\varphi(i+1))\right)\ -\max(\tilde{B}(i), \tilde{A}(\varphi(i)))\right\}\right\}$ (2.5) $i \in [1,\ n-1]$.
Step 5	Construct a fuzzy undirected graph with n nodes and undirected arcs $(i, \varphi(i))$, $i = 1, 2, \ldots, n$.
Step 6	If the current fuzzy graph has only one component, go to Step 8; otherwise select the smallest value $\mathrm{I}(\tilde{C}_{i,i+1})$ such that i is in one component and $i+1$ in another one. In the case of a tie for smallest, choose any.
Step 7	Adjoin the undirected arc $(i,\ i+1)$ to the fuzzy graph using the i value selected in Step 6.
Step 8	Divide the arcs added in Step 7 into two groups. Arcs $(i,\ i+1)$ for which $I\left(\tilde{A}\,(\varphi(i))\right) \geq I\left(\tilde{B}(i)\right)$ go in Group 1 while arcs with $I\left(\tilde{A}\,(\varphi(i))\right) < I\left(\tilde{B}(i)\right)$ go in Group 2.
Step 9	Find the largest index i_1 such that arc $(i_1,\ i_1+1)$ is in Group 1. Find the second largest i_2, etc., up to i_z, assuming there are z elements in Group 1.
Step 10	Find the smallest index j_1 such that arc $(j_1,\ j_1+1)$ is in Group 2. Find the second smallest j_2, etc., up to j_t, assuming there are t elements in Group 2.
Step 11	The optimal permutation for this φ_{ij} is obtained by $\Psi^*(i)$, where $\Psi^*(i) = \varphi_{\alpha_{i_1,\ i_1+1}\ \alpha_{i_2,\ i_2+1}\cdots\ \alpha_{i_z,\ i_z+1}\ \alpha_{j_1,\ j_1+1}\ \alpha_{j_2,\ j_2+1}\cdots\ \alpha_{j_t,\ j_t+1}}\ (i)$. In the above expression the permutation $\alpha_{p,q}$ is defined to be $\alpha_{p,q}\,(p) = q$, $\alpha_{p,q}\,(q) = p$ and $\alpha_{p,q}(i) = i$, if $i \neq p, q$. Denote the optimal permutations obtained at this step by $S^*_{i,j}$.
Step 12	For each optimal permutation, $S^*_{i,j},\ \ i,\ j = 1, \ldots, n,\ i\ \neq j$, compute makespan $\widetilde{\mathcal{M}}'\left(S^*_{ij}\right) = \tilde{z}'(i) + \tilde{v}'(j) + \tilde{w}'\left(S^*_{ij}\right)$.
Step 13	Find permutation S^* that minimizes $\widetilde{\mathcal{M}}\ \left(S^*_{ij}\right) : \widetilde{\mathcal{M}}\ (S^*) = \min_{i,j}\{\widetilde{\mathcal{M}}\ \left(S^*_{ij}\right)\}$.

3. Results

This section presents two numerical examples of flow shop scheduling of an auto port supplier with job-dependent transportation and set-up effects and compares the result of their related crisp and fuzzy makespans.

TABLE 1. The fuzzy time parameters for the nine jobs (min).

Job		$\widetilde{\mathcal{L}}_{\mathrm{I}}$	$\widetilde{\mathcal{T}}_{I,M_1}$	$\widetilde{\mathcal{L}}_{M_1}$	$\widetilde{\mathcal{L}}_{M_2}$	$\widetilde{\mathcal{Q}}_{M_1}$	$\widetilde{\mathcal{Q}}_{M_2}$	$\widetilde{\mathcal{P}}_{M_1}$	$\widetilde{\mathcal{P}}_{M_2}$	$\widetilde{\mathcal{T}}^{e}_{M_2,I}$
1	$\overline{x}_1$	0.08	0.13	0.07	0.11	0.18	0.11	9.12	7.99	0.09
	σ^1_L	0.01	0.03	0.02	0.02	0.03	0.02	0.73	0.84	0.01
	σ^1_R	0.02	0.02	0.01	0.03	0.04	0.04	0.67	0.73	0.02
2	$\overline{x}_2$	0.06	0.09	0.07	0.09	0.12	0.14	7.88	4.09	0.08
	σ^2_L	0.01	0.01	0.02	0.01	0.04	0.03	0.58	0.24	0.01
	σ^2_R	0.02	0.02	0.03	0.02	0.02	0.04	0.55	0.29	0.02
3	$\overline{x}_3$	0.08	0.18	0.09	0.07	0.09	0.11	9.49	2.59	0.07
	σ^3_L	0.01	0.03	0.02	0.01	0.01	0.01	0.61	0.14	0.02
	σ^3_R	0.02	0.04	0.02	0.02	0.02	0.02	0.68	0.17	0.02
4	$\overline{x}_4$	0.04	0.09	0.11	0.14	0.17	0.15	7.38	11.79	0.06
	σ^4_L	0.01	0.02	0.01	0.03	0.02	0.04	0.45	1.04	0.01
	σ^4_R	0.02	0.03	0.03	0.04	0.03	0.04	0.49	1.09	0.01
5	$\overline{x}_5$	0.07	0.21	0.12	0.10	0.15	0.14	6.39	6.19	0.09
	σ^5_L	0.02	0.04	0.02	0.01	0.03	0.02	0.50	0.74	0.02
	σ^5_R	0.03	0.05	0.03	0.03	0.04	0.03	0.45	0.79	0.03
6	$\overline{x}_6$	0.06	0.19	0.05	0.08	0.16	0.11	6.46	2.89	0.05
	σ^6_L	0.01	0.05	0.02	0.02	0.02	0.02	0.61	0.14	0.01
	σ^6_R	0.02	0.04	0.03	0.03	0.01	0.01	0.71	0.19	0.02
7	$\overline{x}_7$	0.07	0.08	0.08	0.04	0.13	0.17	6.42	7.49	0.06
	σ^7_L	0.02	0.02	0.02	0.01	0.01	0.03	0.55	0.34	0.01
	σ^7_R	0.01	0.01	0.03	0.01	0.02	0.04	0.62	0.37	0.01
8	$\overline{x}_8$	0.09	0.16	0.13	0.15	0.17	0.19	6.65	9.29	0.08
	σ^8_L	0.02	0.02	0.02	0.03	0.03	0.04	0.72	0.64	0.02
	σ^8_R	0.02	0.03	0.03	0.02	0.03	0.03	0.81	0.69	0.01
9	$\overline{x}_9$	0.06	0.11	0.09	0.06	0.09	0.07	7.76	5.99	0.07
	σ^9_L	0.01	0.02	0.01	0.01	0.01	0.01	0.85	0.54	0.01
	σ^9_R	0.01	0.03	0.01	0.01	0.02	0.01	0.94	0.63	0.02

TABLE 2. The fuzzy job-independent parameters (min).

	$\widetilde{\mathcal{T}}^{e}_{M_1,M_2}$	$\widetilde{\mathcal{U}}_{M_2}$	$\widetilde{\mathcal{U}}_{M_1}$	$\widetilde{\mathcal{T}}_{M_1,M_2}$	$\widetilde{\mathcal{T}}_{M_2,O}$	$\widetilde{\mathcal{U}}_{O}$	$\widetilde{\mathcal{T}}^{e}_{O,M_1}$
$\overline{x}$	0.05	0.08	0.09	0.07	0.09	0.06	0.07
σ_L	0.01	0.02	0.02	0.01	0.01	0.01	0.02
σ_R	0.01	0.03	0.02	0.01	0.02	0.02	0.03

3.1. Fuzzy example

Table 1 presents time parameters of nine jobs for the sample computation in the form of a Gaussian fuzzy number with different left-hand and right-hand spreads. Moreover, the job-independent parameters of the model are presented in Table 2.

The presented data are transformed by above mentioned formulas into integrated parameters $\tilde{A}(j)$, $\tilde{B}(j)$, $\tilde{z}'(j)$ and $\tilde{v}'(j)$ (see Tab. 3). According to the modified fuzzy Gilmore and Gomory algorithm, we have to repeat Steps 4–11 seventy-two times for each pair of indices, i and j, $i \neq j$, $i, j = 1, \ldots, 9$, being excluded. For this numerical example, the results of the algorithm for a fixed pair $(6, 8)$ are presented. For the other pairs, a similar process should be repeated. The auxiliary pair of fuzzy numbers, $(\tilde{A}(10), \tilde{B}(10) = ((14, 0.9, 0.95), (14, 0.8, 0.85))$ are added to previous data. Next, we find "a partial sorting," φ_{68}, which is obtained from the sorting at Step 2

TABLE 3. Integrated parameters (min).

Job		$\widetilde{A}$	$\widetilde{B}$	$\widetilde{v}$	$\widetilde{z}'$
1	$\bar{x}_1$	9.5	8.3	7.99	9.4
	σ_L^1	0.8	0.9	0.84	0.79
	σ_R^1	0.87	0.99	0.88	0.84
2	$\bar{x}_2$	8.2	4.4	4.09	8.1
	σ_L^2	0.6	0.3	0.24	0.59
	σ_R^2	0.64	0.32	0.26	0.63
3	$\bar{x}_3$	9.9	2.9	2.59	9.84
	σ_L^3	0.7	0.2	0.14	0.68
	σ_R^3	0.74	0.23	0.18	0.76
4	$\bar{x}_4$	7.7	12.1	11.79	7.62
	σ_L^4	0.5	1.1	1.04	0.49
	σ_R^4	0.51	1.13	1.10	0.55
5	$\bar{x}_5$	6.9	6.5	6.19	6.79
	σ_L^5	0.6	0.8	0.74	0.58
	σ_R^5	0.71	0.88	0.79	0.61
6	$\bar{x}_6$	6.8	3.2	2.89	6.76
	σ_L^6	0.7	0.2	0.14	0.69
	σ_R^6	0.71	0.22	0.16	0.71
7	$\bar{x}_7$	6.7	7.8	7.49	6.65
	σ_L^7	0.6	0.4	0.34	0.59
	σ_R^7	0.62	0.44	0.37	0.61
8	$\bar{x}_8$	7.1	9.6	9.29	7.03
	σ_L^8	0.8	0.7	0.64	0.78
	σ_R^8	0.83	0.75	0.68	0.81
9	$\bar{x}_9$	8.1	6.3	5.99	8.02
	σ_L^9	0.9	0.3	0.54	0.89
	σ_R^9	0.93	0.31	0.57	0.92

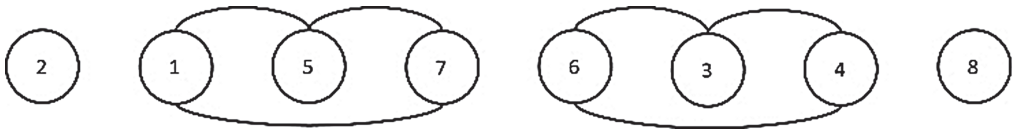

FIGURE 1. Undirected graph with four components (Step 5).

by excluding $\tilde{A}(\varphi(6))$ and $\tilde{A}(\varphi(8))$. Then, we compute the fuzzy cost of each arc using equation (2.5). Table 4 presents the results of Steps 1–4. Next, we construct an undirected graph with 8 nodes and undirected arcs $(i, \varphi(i)), i = 1, 2, \ldots, 8$. By referring to columns, job and q of Table 4, we draw the undirected edges $i - \varphi(i)$ on the graph (see Fig. 1). Now, we select the smallest value of $\tilde{C}_{i,i+1}$ such that i is in one component and $i + 1$ in another one. In the case of a tie for smallest, we choose arbitrarily. The graph in Figure 1 has four components. Node 2 is in the first component, nodes 1, 5, and 7 are in the second component, nodes 3, 4, and 6 are in the third component, and node 8 is in the fourth component. For this example, all edges have a cost of zero. Thus, we add undirected arcs (1, 2), (6, 7), and (7, 8) to the graph to obtain a single-component graph. Steps 6 and 7 are now completed. Finally, we can see the final result in Figure 2.

An inspection of Table 4 shows that the condition $I\left(\tilde{A}(\varphi(i))\right) \geq I\left(\tilde{B}(i)\right)$ is satisfied by the arcs (6, 7) and (7, 8). Hence, from Step 8, we obtain two groups: Group 1 containing the arcs (6, 7) and (7, 8), and Group 2

TABLE 4. Results of Steps 1–4.

Job		$\widetilde{A}$	$\widetilde{B}$	q	$\widetilde{C}_{i,i+1}$
1	$\bar{x}_1$	6.7	7.8		
	σ_L^1	0.6	0.4	5	$\tilde{0}$
	σ_R^1	0.62	0.44		
2	$\bar{x}_2$	6.8	3.2		
	σ_L^2	0.7	0.2	2	$\tilde{0}$
	σ_R^2	0.71	0.88		
3	$\bar{x}_3$	6.9	6.5		
	σ_L^3	0.6	0.8	4	$\tilde{0}$
	σ_R^3	0.71	0.88		
4	$\bar{x}_4$	7.1	9.6		
	σ_L^4	0.8	0.7	6	$\tilde{0}$
	σ_R^4	0.83	0.75		
5	$\bar{x}_5$	7.7	12.1		
	σ_L^5	0.5	1.1	7	$\tilde{0}$
	σ_R^5	0.51	1.13		
6	$\bar{x}_6$	8.2	4.4		
	σ_L^6	0.6	0.3	3	$\tilde{0}$
	σ_R^6	0.64	0.32		
7	$\bar{x}_7$	9.9	2.9		
	σ_L^7	0.7	0.2	1	$\tilde{0}$
	σ_R^7	0.74	0.23		
8	$\bar{x}_8$	14	14		
	σ_L^8	0.9	0.8	8	–
	σ_R^8	0.95	0.85		

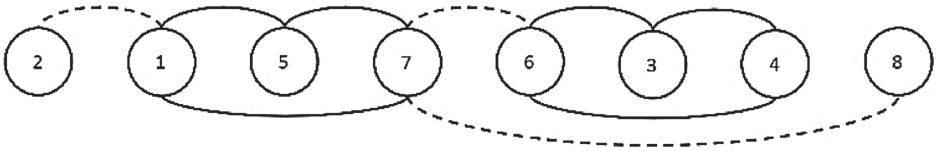

FIGURE 2. A Single-Component Graph (Steps 6 and 7).

TABLE 5. An optimal permutation for $\varphi_{6,8}$.

j	1	2	3	4	5	6	7	8
$\Psi^*(j)$	2	7	6	3	1	5	8	4

containing arc (1, 2). Finally, the optimal permutation for the partial sorting, φ_{68} is obtained by $\Psi^*(i)$, where

$$\Psi^*(i) = \varphi_{\alpha_{6,7}\ \alpha_{7,8}\ \alpha_{1,2}}\ \ (i).$$

For example, $\Psi^*(1) = \varphi(\alpha_{6,7}(\alpha_{7,8}\,(\alpha_{1,2}\,(1)))) = 2$. Thus, job 1 follows job 2. Similarly, the other computations can be done as can be seen in Table 5. Eventually, an optimal permutation $(J_1, J_2, J_7,\ J_8,\ J_4, J_3, J_6, J_5)$ for $\varphi_{6,8}$ can be obtained from the pairs $(j,\ \Psi^*(j))$.

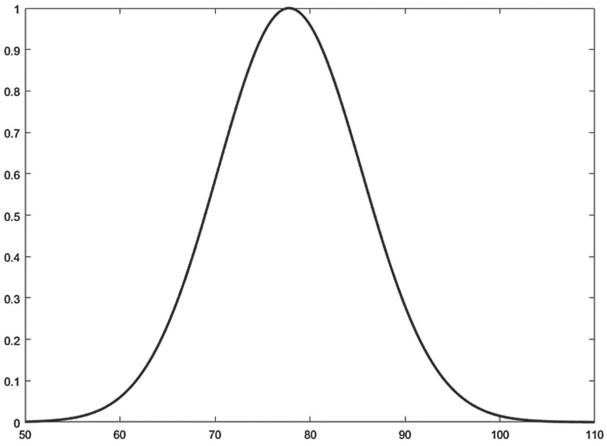

FIGURE 3. The fuzzy makespan obtained by the fuzzy algorithm.

Now, we denote the optimal permutations obtained at Step 11 by $S^*_{6,8}$. At Step 12, we compute makespan $\widetilde{\mathcal{M}}'\left(S^*_{ij}\right) = \tilde{z}'(i) + \tilde{v}'(j) + \tilde{w}'\left(S^*_{ij}\right)$ for each optimal permutation, $S^*_{i,j}$. Based on our computations, the makespan value for $S^*_{6,8}$ is $\widetilde{\mathcal{M}}'\left(S^*_{68}\right) = (85.40, 8.02, 8.19)$. Finally, the permutation S^* can be found by minimizing $\widetilde{\mathcal{M}}\left(S^*_{ij}\right) : \widetilde{\mathcal{M}}\ (S^*) = \min_{i,j}\{\widetilde{\mathcal{M}}\left(S^*_{ij}\right)\}$. The optimal permutation obtained in Step 13 is as follows: $S^*_{\text{fuzzy}} = (J_8, J_1, J_2,\ J_7,\ J_9, J_5, J_6, J_4, J_3)$ with $\widetilde{\mathcal{M}}\left(S^*_{\text{fuzzy}}\right) = (77.79, 7.502, 7.663)$. Figure 3 demonstrates the fuzzy makespan of this problem. Now, defuzzified makespan can be computed using BADD defuzzification method. If we consider $\beta = 5$, the following defuzzified makespan can be obtained:

$$\text{DF}\left(\widetilde{\mathcal{M}}\left(S^*_{\text{fuzzy}}\right)\right) = 77.848.$$

3.2. Crisp example

For the crisp example, we defuzzified the fuzzy time parameters for nine jobs (Tab. 1) and the fuzzy job-dependent parameters (Tab. 2). Table 6 shows time parameters of nine jobs for the sample computation. Besides, the job-independent parameters of the model are presented in Table 7.

Since the steps for this problem are similar, we just explain the final results. Using modified Gilmore and Gomory algorithm, the optimal sequence of jobs and its related makespan are as follows:

$$S^*_{\text{crisp}} = (J_8, J_1, J_9,\ J_6,\ J_5, J_2, J_7, J_4, J_3) \quad \mathcal{M}\left(S^*_{\text{crisp}}\right) = 78.237.$$

4. DISCUSSION

In most of scheduling problems, we need to define some parameters such as loading times, processing times, transportation times, and so on. Moreover, these parameters have a huge impact on the final results of models. Furthermore, most of the time, finding the accurate values of these parameters are difficult (if not impossible). In real-world applications, managers need to decide on the parameters of models. Since these parameters have an uncertain nature and they can change due to the occurrence of unexpected incidents, considering crisp values will not be the best option. In other words, if we use a crisp number for these parameters, we will lose valuable information.

TABLE 6. Time parameters for the nine jobs (min).

Job	$\mathcal{L}_{\mathrm{I}}$	$\mathcal{T}_{I,M_1}$	$\mathcal{L}_{M_1}$	$\mathcal{L}_{M_2}$	$\mathcal{Q}_{M_1}$	$\mathcal{Q}_{M_2}$	$\mathcal{P}_{M_1}$	$\mathcal{P}_{M_2}$	$\mathcal{T}^{e}_{M_2,I}$
1	0.081	0.137	0.069	0.114	0.178	0.112	9.321	8.003	0.088
2	0.062	0.093	0.071	0.093	0.119	0.137	7.683	3.992	0.081
3	0.077	0.191	0.088	0.068	0.087	0.121	9.529	2.621	0.072
4	0.044	0.092	0.111	0.138	0.172	0.143	7.473	11.902	0.058
5	0.067	0.221	0.109	0.102	0.154	0.144	6.299	6.206	0.092
6	0.067	0.187	0.052	0.084	0.157	0.109	6.576	2.903	0.055
7	0.068	0.078	0.084	0.041	0.132	0.172	6.492	7.407	0.061
8	0.091	0.163	0.131	0.146	0.171	0.188	6.715	9.311	0.077
9	0.062	0.121	0.089	0.063	0.093	0.071	7.811	6.002	0.073

TABLE 7. The job-independent parameters of the model (min).

$\mathcal{T}^{e}_{M_1,M_2}$	$\mathcal{U}_{M_2}$	$\mathcal{U}_{M_1}$	$\mathcal{T}_{M_1,M_2}$	$\mathcal{T}_{M_2,O}$	$\mathcal{U}_{O}$	$\mathcal{T}^{e}_{O,M_1}$
0.053	0.082	0.089	0.068	0.091	0.062	0.073

Fuzzy logic is often perceived as a powerful tool to deal with imprecise data. Using the capability of fuzzy logic to represent approximate values, we can represent these parameters without losing valuable information. To that end, managers can define the time parameters of all jobs on each machine using fuzzy L-R numbers. We chose L-R fuzzy numbers since their definition is very general and allows quantification of quite different types of information.

To demonstrate the application of our algorithm in providing managerial insight, we solved two problems with crisp and fuzzy time parameters, respectively. As we saw in the previous section, our results demonstrate that the defuzzified makespan $\mathrm{DF}\left(\widetilde{\mathcal{M}}\left(S^{*}_{\mathrm{fuzzy}}\right)\right) = 77.848$ is rather smaller than its related crisp makespan $\mathcal{M}\left(S^{*}_{\mathrm{crisp}}\right) = 78.237$. This makespan reduction is not the main advantage of our model. In fact, what we achieved using the fuzzy algorithm is more reliable than the crisp version because we considered different types of uncertainties in our parameters. Thus, even if the parameters slightly change during the process our result will still be robust and reliable. Depending on the nature of a time parameter, managers are able to define different levels of uncertainty. Clearly, if we define the fuzzy parameters precisely (*e.g.*, small σ_L and σ_R in Gaussian fuzzy numbers), the uncertainty in our final makespan will be reduced. Moreover, the optimal sequence of jobs for the crisp problem is $(J_8, J_1, J_9,\ J_6,\ J_5, J_2, J_7, J_4, J_3)$ and for the fuzzy version is $(J_8, J_1, J_2,\ J_7,\ J_9, J_5, J_6, J_4, J_3)$. As you can see there are minor changes in these two job sequences. However, these changes can improve the performance of the factory and increase productivity.

More importantly, it can help managers avoid financial penalties. As mentioned above, if they estimate a parameter slightly higher or lower than the actual value, the crisp model will increase production costs. The fuzzy version can mitigate this issue and evade this risk. Therefore, it will increase cost-effectiveness and service efficiencies in the long run.

The computational cost of the fuzzy algorithm is higher than the crisp one. However, although the computational complexity is increased by the employment of fuzzy numbers, it is a small price to pay for obtaining satisfactory results.

Finally, here we just solve one sample of the problems that can be solved using this model. In other words, we can similarly use fuzzy numbers for other parameters of models. Since Gilmore and Gomory algorithm has been used in many scheduling problems, we can further optimize those models using our fuzzy version of this algorithm.

5. Conclusions and future research

This paper has presented the fuzzy scheduling of the flexible manufacturing cell with job-dependent processing and material-handling operations. Since transportation and set-up times are a large portion of the production time in a flexible manufacturing cell, ignoring these parameters may cause significant errors in determining the optimal makespan. Furthermore, determining the exact values of these time parameters is a challenging task. To overcome this problem, we represented these parameters using fuzzy numbers. Using the capability of fuzzy numbers to represent approximate values, we can represent these parameters without losing valuable information. In this way, we convert our original problem into a fuzzy one. Then, to solve the fuzzy problem, we generalized Gilmore and Gomory algorithm into fuzzy Gilmore and Gomory algorithm.

The proposed algorithm was tested with a small real-world problem. We compared the results of the proposed fuzzy method with those of the crisp ones. Our computations indicate that the proposed fuzzy approach has a better performance compared to the crisp algorithm.

Although our model is quite promising, it also suffers from some limitations that can be addressed in future work. First, in the defuzzification step, we used a simple method. Since this step has a significant effect on the final result, we can further improve this algorithm using more complicated defuzzification techniques. Furthermore, an interesting future direction is to adopt deferent types of fuzzy numbers. Using a comparative study, we analyze the advantages and disadvantages of various fuzzy numbers. Additionally, in the second step of the algorithm, we need to use a ranking method to sort fuzzy numbers. This step is the most computationally expensive part of our algorithm. By applying a Computationally efficient ranking method, we can further optimize our model and reduce its computational cost.

References

[1] M. Allahverdi and A. Allahverdi, Algorithms for four-machine flowshop scheduling problem with uncertain processing times to minimize makespan. *RAIRO:OR* **54** (2020) 529–553.

[2] K. Amrouche and M. Boudhar, Two machines flow shop with reentrance and exact time lag. *RAIRO:OR* **50** (2016) 223–232.

[3] E. Babaee Tirkolaee, A. Goli and G.W. Weber, 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.* **28** (2020) 2772–2783.

[4] K.R. Baker and D. Trietsch, Principles of Sequencing and Scheduling. John Wiley & Sons (2013).

[5] R.E. Burkard, V.G. Deineko, R. van Dal, J.A.A. van der Veen and G.J. Woeginger, Well-solvable special cases of the traveling salesman problem: a survey. *SIAM Rev.* **40** (1998) 496–546.

[6] M.H. Fazel Zarandi, A.A. Sadat Asl, S. Sotudian and O. Castillo, A state of the art review of intelligent scheduling. *Artif. Intell. Rev.* **53** (2020) 501–593.

[7] D.P. Filev and R.R. Yager, A generalized defuzzification method *via* bad distributions. *Int. J. Intell. Syst.* **6** (1991) 687–697.

[8] P.C. Gilmore and R.E. Gomory, Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. *Oper. Res.* **12** (1964) 655–679.

[9] A. Goli, E.B. Tirkolaee and M. Soltani, A robust just-in-time flow shop scheduling problem with outsourcing option on subcontractors. *Prod. Manuf. Res.* **7** (2019) 294–315.

[10] H. Golpîra and E. B. Tirkolaee, Stable maintenance tasks scheduling: A bi-objective robust optimization model. *Comput. Ind. Eng.* **137** (2019) 106007.

[11] M. Hanss, Applied Fuzzy Arithmetic: An Introduction with Engineering Applications. Springer, Berlin-Heidelberg (2005).

[12] H. Kise, T. Shioyama and T. Ibaraki, Automated two-machine flowshop scheduling: a solvable case. *IIE Trans.* **23** (1991) 10–16.

[13] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Sequencing and scheduling: Algorithms and complexity. In: Logistics of Production and Inventory, edited by S.S. Graves, A.H.G. Rinnooy Kan, P. Zipkin. Vol. 4 of *Handbooks in Operations Research and Management Science*. North-Holland (1993) 445–522.

[14] L. Lei, R. Armstrong and S. Gu, Minimizing the fleet size with dependent time-window and single-track constraints. *Oper. Res. Lett.* **14** (1993) 91–98.

[15] J. Leung and G. Zhang, Optimal cyclic scheduling for printed circuit board production lines with multiple hoists and general processing sequence. *IEEE Trans. Robot. Autom.* **19** (2003) 480–484.

[16] E.V. Levner, Optimal planning of parts' machining on a number of machines. *Autom. Remote Control* **12** (1969) 1972–1978.

[17] E. Levner, K. Kogan and O. Maimon, Flowshop scheduling of robotic cells with job-dependent transportation and set-up effects. *J. Oper. Res. Soc.* **46** (1995) 1447–1455.

[18] W. Li, J. Li, K. Gao, Y. Han, B. Niu, Z. Liu and Q. Sun, Solving robotic distributed flowshop problem using an improved iterated greedy algorithm. *Int. J. Adv. Robot. Syst.* **16** (2019) 1729881419879819.

[19] E.N. Nasibov and A. Mert, On methods of defuzzification of parametrically represented fuzzy numbers. *Autom. Control Comput. Sci.* **41** (2007) 265–273.

[20] M. Rabbani, M. Samavati, M.S. Ziaee and H. Rafiei, Reconfigurable dynamic cellular manufacturing system: a new bi-objective mathematical model. *RAIRO:OR* **48** (2014) 75–102.

[21] A.A. Sadat Asl, M.H. Fazel Zarandi, S. Sotudian and A. Amini, A fuzzy capacitated facility location-network design model: a hybrid firefly and invasive weed optimization (FIWO) solution. *Iran. J. Fuzzy Syst.* **17** (2020) 79–95.

[22] H.I. Stern and G. Vitner, Scheduling parts in a combined production-transportation work cell. *J. Oper. Res. Soc.* **41** (1990) 625–632.

[23] S. Sotudian, M.H.F. Zarandi and I.B. Turksen, From Type-I to Type-II fuzzy system modeling for diagnosis of hepatitis. *Int. J. Comput. Inf. Eng.* **10** (2016) 1280–1288.

[24] A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri and B. Sadeghpour-Gildeh, Computing a fuzzy shortest path in a network with mixed fuzzy arc lengths using alpha-cuts. *Comput. Math. Appl.* **60** (2010) 989–1002.

[25] P. Wanke, C.P. Barros and A. Emrouznejad, A comparison between stochastic DEA and fuzzy DEA approaches: revisiting efficiency in Angolan banks. *RAIRO:OR* **52** (2018) 285–303.