

# A reactive approach for flexible job shop scheduling problem with tardiness penalty under uncertainty

Behnam Ayyoubzadeh<sup>a</sup>, Sadoullah Ebrahimnejad<sup>b,\*</sup>, Mahdi Bashiri<sup>c</sup>, Vahid Baradaran<sup>d</sup>, Seyed Mohammad Hassan Hosseini<sup>e</sup>

<sup>a</sup> Industrial Engineering Department, Faculty of Engineering, Islamic Azad University-North Tehran Branch, Tehran, Iran

E-mail: b.ayyoubzadeh@iau-tnb.ac.ir

<sup>b</sup> Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran

E-mail: ibrahimnejad@kiau.ac.ir

<sup>c</sup> School of Strategy and Leadership, Faculty of Business and Law, Coventry University, Coventry, UK

E-mail: mahdi.bashiri@coventry.ac.uk

<sup>d</sup> Industrial Engineering Department, Faculty of Engineering, Islamic Azad University-North Tehran Branch, Tehran, Iran

E-mail: v\_baradaran@iau-tnb.ac.ir

<sup>e</sup> Industrial Engineering and Management Department, Shahrood University of Technology, Shahrood, Iran

E-mail: sh.hosseini@shahroodut.ac.ir

---

## Abstract

Flexible job shop scheduling under uncertainty plays an important role in real-world manufacturing systems. This paper deals with the flexible job shop scheduling problem to minimize the sum of jobs' tardiness considering machines breakdown and order due date modification as two important disruptions in this production system. To this end, the problem is formulated as a mixed-integer linear programming model. In addition, two strategies are proposed including allocating multiple machines to each job and selecting the best alternative process from other jobs to handle these disruptions. Since the problem is well-known strongly NP-hard, a hybrid metaheuristic algorithm based on the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) is proposed to solve the real-sized instances of these problems. Numerical experiments are used to evaluate the performance and effectiveness of the proposed hybrid algorithm. Obtained results for the small-sized instances show that the proposed algorithm provides proper solutions in terms of optimality and CPU Time. In addition, results for the medium- and large-sized scales validate the efficiency of the proposed algorithm and indicate that the proposed hybrid solution approach outperformed the classic GA in terms of the objective function value and the CPU time.

**Keywords:** Flexible Job Shop, Reactive Scheduling, Uncertainty, Tardiness

---

## 1. Introduction

The job shop scheduling problem (JSSP), which was proposed in 1976 by Coffman and Bruno [1], is one of the most popular scheduling problems that is receiving increasing attention in the field of academic research and manufacturing enterprise. The flexible job shop scheduling problem (FJSSP) as a generalization of JSSP is a classical combinatorial optimization problem in real-world manufacturing systems. In FJSSP, an operation can be processed by more than one machine, i.e., alternative routings are allowed. This kind of production system has expanded access to machines for all of the operations; consequently, it can be an efficient method for overcoming the existing uncertainty [2-6].

Uncertainties such as machine breakdowns and modifications or cancellations of the orders cause usually failure in initial scheduling. These unexpected disruptions can lead to changes in order modification and the subsequent changes in product delivery date, and even jobs inaccessibility in the start time of operations on machines, may lead to the stochastic start time of operations [7].

If significant disruptions occur in a job shop, there is no way instead of complete rescheduling; but in the case of insignificant disruptions, using an efficient approach can adapt the initial scheduling to the new condition. For this purpose, reactive scheduling is well-known as a proper approach to deal with these kinds of disruptions. Most of the present reactive scheduling methods are based on a real-time dispatching strategy.

To repair the suggested schedule by the predictive scheduling in real-time due to internal disruptions (e.g. machine breakdown) and external disruptions (e.g. postponing the delivery date of orders), reactive scheduling uses intelligent technology, mainly Affected Operation Rescheduling (AOR) and Right Shift Rescheduling (RSR). Fig. 1 shows the way reactive scheduling deals with disruptions during production.

Please insert here Fig. 1

Reactive scheduling is divided into two categories: (a) rescheduling, and (b) repair scheduling. Repair scheduling would be a proper approach to adapt the original scheduling to the new status. This approach attempts to improve the original scheduling so that the best results are achieved in terms of unexpected internal and external disruptions. This study aims to conduct the FJSSP by considering machines breakdown and order due date modification as the most important disruptions in this production system. In this way, repair scheduling is used to respond to these disruptions during running the original predictive scheduling. To tackle the problem, a predefined probability distribution is assumed for the start time and processing time.

The remaining sections of this study are organized as follows: A literature review is provided in section (2). Problem definition and the proposed mathematical model are presented in section (3). Section (4) is devoted to the solution approach that is a hybrid procedure based on GA and PSO. Numerical instances and result analysis are presented in section (5). Finally, section (6) provides a conclusion and future researches.

## 2. Literature review

The FJSSP has many applications in manufacturing industries, and hence, has received more attention of researchers in last years [8-10]. However, little attention has been paid to address the problem under uncertainty. Jackson [11] distinguished the notion of static and dynamic scheduling from each other. This section presents the literature on scheduling problems under uncertainty with a focus on reactive approaches in solving them.

Sabuncuoglu and Bayiz [12] used a reactive scheduling approach in the production environment, they analyzed different scheduling policies related to the condition of machines' unavailability in the classic job shop. Chryssolouris and Subramaniam [13] proposed a GA for reactive scheduling in a job shop scheduling problem (JSSP). They considered two effective criteria for performance including average job's tardiness and average job's cost. Albers and Schmidt [14] they studied the allocation problem of a jobs' set in online scheduling on several identical machines to minimize the makespan. They assumed that machines are not permanently available due to breakdown. Qi et al. [15] studied a single machine scheduling problem wherein, jobs have distinct due dates. They examined the single machine case in detail in order to find the optimal sequence of jobs. Suwa and Sandoh [16] suggested a scheduling policy for job-shop reactive scheduling under the circumstances of machines' unavailability. Hao and Lin [17] studied one of the classic scheduling in a dynamic environment that emphasizes responsiveness to environmental changes. Adibi et al. [18] provided an event-driven policy JSSP in which jobs arrive stochastically. In the presence of machine breakdown, they proposed an artificial neural network for searching parameters in a rescheduling point. Kianfar et al. [19] considered an FJSSP with sequence-dependent setup times under uncertainty wherein, jobs arrive dynamically. He and Sun [20] investigated the FJSSP while machine breakdowns can happen. They provided a multi-strategies approach to make a robust and stable schedule. Zhang et al. [21] developed a new hybrid intelligent algorithm; they considered two kinds of disruptions, such as random jobs arrival, and machine breakdown. De Freitas Rodrigues et al. [22] studied a scheduling problem with identical parallel machines. They considered that some specific jobs are not able to be processed on some specific machines. He et al. [23] studied a single machine system with parallel batching where rejection is allowed. They proposed dynamic programming algorithm and approximation algorithm in order to minimize the makespan and the total rejection cost. Zinder and Walker [24] proposed a new method for scheduling on parallel identical machines with integer preemptions.

Literature review indicates that there are many studies in tackling the disruption issue in scheduling problems. However, a few efforts have investigated disruption in JSP and FJSP using reactive approaches. Rey et al. [25] studied the FJSSP in a just-in-time (JIT) production system. They used two meta-heuristic (GA and PSO) for managing job release times. Rohaninejad et al. [26] addressed a bi-objective FJSSP under uncertainty considering machines' capacity constraints. Their objectives were minimizing the makespan and overtime costs of machines. They formulated the problem using a nonlinear integer programming model. Moreover, they developed a hybrid meta-heuristic algorithm to solve problem on practical scales. Shen and Zhu

[27] studied a JSSP under uncertainty in processing time and cost. Xie and Chen[28] have solved a JSS model under a new uncertainty for minimizing the completion time. Also, they have examined the gray processing time interval for these problems.

Since the FJSSP is well-known strongly NP-hard, many studies used heuristic or metaheuristic algorithms to solve the FJSSP [29]. Tavakkoli-Moghaddam et al. [30] suggested a hybrid approach based on an artificial neural network and a two-step annealing algorithm to solve a stochastic JSSP. Subramaniam et al. [31] provided a heuristic in order to adjust randomly occurred multiple disruptions. Their proposed method showed proper solutions and outperformed other methods. Xiuli [32] considered an FJSSP and suggested a multi-objective optimization method based on GA. Fattahi and Fallahi [33] used a dynamic programming model for the FJSSP and suggested two functions to establish a trade-off between efficiency and reliability using a GA. Lei [34] developed an efficient decomposed-integration GA for minimizing the maximum fuzzy completion time in a flexible job shop. Hatami et al. [35] proposed a three-stage assembly scheduling model with sequence-dependent setup time. They designed two meta-heuristics namely, simulated annealing and tabu search. Liu et al. [36] studied a flexible job shop dynamic scheduling and proposed a multi-objective model that includes tardiness penalty and construction time. They adapted GA for strategic scheduling of cycle-driven. Jolai et al. [37] developed a novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop system; their main objective is to minimize the maximum completion time. Hokim et al. [38] implemented a tire manufacturing process scheduling using an efficient PSO algorithm with the aim of minimizing the makespan. Naderi-Beni et al.[39] studied an unrelated parallel machines scheduling problem. They proposed a fuzzy bi-objective mixed-integer linear programming model. They addressed sequence-dependent setup times, machine eligibility restrictions and release dates. Shen et al. [40] provided a FJSS model with setup time to minimize makespan. In this paper, they defined a tabu search algorithm with a specific neighborhood. Jamrus et al. [41] developed a hybrid approach integrating a PSO with a cauchy distribution and GA (HPSO+GA) for solving an FJSS problem by finding a job sequence that minimizes the makespan with uncertain processing time. Soleimani nia and Mehdizadeh [42] presented a mathematical model for FJSS problem with reverse flows and solved it using GA. Huang et al. [43] proposed a teaching-learning-based hybrid genetic-particle swarm optimization algorithm to address multi-objective an FJSSP. Thereinafter, researchers used a hybrid procedure based on GA and glowworm swarm optimization algorithm with a green transport heuristic strategy (GA-GSO-GTHS) [44].

Although several studies have dealt with the FJSSP under uncertainty, there is a lack of efficient solution procedures for the FJSSP under uncertainty considering machine breakdowns and modification in order delivery date. So, the novelty of this paper can be stated as formulating and proposing a new reactive scheduling approach to handle two aforementioned disruptions in the FJSSP. In this way, two strategies are proposed including allocating multiple machines to each job and selecting the best alternative process from other jobs as a reactive approach.

### **3. Problem definition and mathematical model**

The FJSSP by considering machines breakdown and order due date modification as two common kinds of disruption is investigated and solved in this paper. The objective is to minimize the sum of tardiness. This section aims to formulate the problem as MIP model to better understand. Firstly, assumptions of the problem are addressed as follows:

- Precedence relationship among operations of a job is predefined
- Each operation for each job is being allocated to one machine
- Operations sequence on jobs have priority
- Jobs sequence on machines have priority
- None of the operations can be stopped (preemptive)
- The setup time of machines that includes the process time is independent for operations process
- The start time and operation process are random with a uniform probability distribution

#### Indices and sets:

$i, h$	Indices for jobs, $i, h \in \{1, 2, \dots, J\}$
$j, k$	Indices for operations, $j, k \in \{1, 2, \dots, N_i\}$
$m$	Indices for machines, $m \in \{1, 2, \dots, M_c\}$
$c$	Indices for stages, $c \in \{1, 2, \dots, C\}$

#### General parameters:

$J$	Number of jobs
$C$	Number of stages
$N_i$	Number of operations of $i^{th}$ job
$M_c$	Number of the similar machines in the $c^{th}$ stage
$O_{ij}$	$j^{th}$ operation of $i^{th}$ job
$\alpha_{cij}$	A binary parameter that takes value 1 if $j^{th}$ operation of $i^{th}$ job requires to the $c^{th}$ stage; 0 otherwise
$M$	A very large number

#### Parameters relevant to the original schedule of the $\tau^{th}$ period:

$J^\tau$	Jobs' set of the $\tau^{th}$ period
$ep_{cijm}^\tau$	The earliest processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the time-interval of the $\tau^{th}$ period
$lp_{cijm}^\tau$	The latest processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the time-interval of the $\tau^{th}$ period

$\xi p_{cijm}^\tau$	Processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the $(ep_{cijm}^\tau, lp_{cijm}^\tau)$ time-interval of the $\tau^{th}$ period (random parameter)
$d_i^\tau$	Delivery date of $i^{th}$ job of the $\tau^{th}$ period
$et_{ij}^\tau$	The earliest start time of $j^{th}$ operation of $i^{th}$ job in the time-interval of the $\tau^{th}$ period
$lt_{ij}^\tau$	The latest start time of $j^{th}$ operation of $i^{th}$ job in the time-interval of the $\tau^{th}$ period
$\xi t_{ij}^\tau$	Start time of $j^{th}$ operation of $i^{th}$ job in the $(et_{ij}^\tau, lt_{ij}^\tau)$ time-interval of the $\tau^{th}$ period (random parameter)
$\xi D^\tau$	Occurrence time of disruption of the $\tau^{th}$ period (random parameter)
$b_{cijm}^\tau$	A binary parameter that takes value 1 if $m^{th}$ machine is allocated to the $O_{ij}$ in the $c^{th}$ stage of the $\tau^{th}$ period; 0 othrewise

---

**Parameters relevant to the reactive scheduling of the  $\gamma^{th}$  period:**

---

$J^\gamma$	Correction of jobs' set of the $\gamma^{th}$ period
$ep_{cijm}^\gamma$	The earliest processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the time-interval of the $\gamma^{th}$ period
$lp_{cijm}^\gamma$	The latest processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the time-interval of the $\gamma^{th}$ period
$\xi p_{cijm}^\gamma$	Processing time of $O_{ij}$ on the $m^{th}$ machine in the $c^{th}$ stage in the $(ep_{cijm}^\gamma, lp_{cijm}^\gamma)$ time-interval of the $\gamma^{th}$ period (random parameter)
$d_i^\gamma$	Correction of delivery date of $i^{th}$ job of the $\gamma^{th}$ period
$T_{cm}^\gamma$	Inaccessibility time of $m^{th}$ machine in the $c^{th}$ stage of the $\gamma^{th}$ period

---

**Decision Variables:**

---

$v_{ij}^\tau$	A binary parameter that takes value 1 if the strat time of $O_{ij}$ occurs sooner than disorder's outbreaking time of the $\tau^{th}$ period; 0 othrewise
$x_{cijm}^\gamma$	A binary parameter that takes value 1 if $m^{th}$ machine is allocated to the $O_{ij}$ in the $c^{th}$ stage of the $\gamma^{th}$ period; 0 othrewise
$\xi t_{ij}^\gamma$	Start time of $j^{th}$ operation of $i^{th}$ job in the $(et_{ij}^\gamma, lt_{ij}^\gamma)$ time-interval of the $\gamma^{th}$ period (random variable)

$z_{cjh}^\gamma$ 

A binary parameter that takes value 1 if  $i^{th}$  job precedes to the  $h^{th}$  job, and both of them require the  $c^{th}$  stage of the  $\gamma^{th}$  period; 0 otherwise

In the following section, the proposed mathematical model has been defined:

$$\text{Min} \sum_{i=1}^{J^\gamma} \text{Max} (0, (\xi t_{iN_i}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_i m}^\gamma x_{CiN_i m}^\gamma - d_i^\gamma)) \quad (1)$$

s.t.:

$$\sum_{c=1}^C \sum_{m=1}^{M_c} x_{cijm}^\gamma \cdot \alpha_{cij} = 1 - v_{ij}^\tau, \quad \forall i, j \quad (2)$$

$$\begin{cases} \xi t_{ij}^\tau \leq \xi D^\tau + M(1 - v_{ij}^\tau) & , \quad \forall i, j \\ \xi D^\tau \leq \xi t_{ij}^\tau + M v_{ij}^\tau & , \quad \forall i, j \end{cases} \quad (3)$$

$$\begin{cases} lt_{ij}^\tau + \sum_{c=1}^C \sum_{m=1}^{M_c} lp_{cijm}^\tau b_{cijm}^\tau \leq et_{i(j+1)}^\gamma + M(1 - v_{ij}^\tau) + M v_{i(j+1)}^\tau \\ , \forall i, j = \{1, 2, \dots, (N_i - 1)\} \\ lt_{ij}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} x_{cijm}^\gamma \cdot lp_{cijm}^\gamma \leq et_{i(j+1)}^\gamma + M v_{ij}^\tau, \forall i, j = \{1, 2, \dots, (N_i - 1)\} \end{cases} \quad (4)$$

$$\xi D^\tau + T_{cm}^\gamma \leq et_{ij}^\gamma + M(1 - x_{cijm}^\gamma) + M v_{ij}^\tau \quad \forall i, j, c, m \quad (5)$$

$$\begin{cases} \xi t_{ij}^\tau + \sum_{c=1}^C \sum_{m=1}^{M_c} b_{cijm}^\tau \cdot \xi p_{cijm}^\tau \leq \xi t_{i(j+1)}^\gamma + M(1 - v_{ij}^\tau) + M v_{i(j+1)}^\tau \\ , \forall j = \{1, 2, \dots, (N_i - 1)\}, i \\ \xi t_{ij}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} x_{cijm}^\gamma \cdot \xi p_{cijm}^\gamma \leq \xi t_{i(j+1)}^\gamma + M v_{ij}^\tau, \forall i, j = \{1, 2, \dots, (N_i - 1)\} \end{cases} \quad (6)$$

$$\alpha_{chk} \cdot (\xi t_{hk}^\tau + b_{chkm}^\tau \cdot \xi p_{chkm}^\tau) \leq \xi t_{ij}^\gamma \cdot \alpha_{cij} + M(1 - \alpha_{cij}) + M(1 - x_{cijm}^\gamma) + M(1 - b_{chkm}^\tau) + M(1 - v_{hk}^\tau) + M v_{ij}^\tau, \quad \forall h, j, i, k, c, m \quad (7)$$

$$\begin{cases} \alpha_{chk} \cdot (\xi t_{hk}^\gamma + x_{chkm}^\gamma \cdot \xi p_{chkm}^\gamma) \leq \xi t_{ij}^\gamma \cdot \alpha_{cij} + M z_{cjh}^\gamma + M(1 - \alpha_{cij}) + M(1 - x_{cijm}^\gamma) \\ + M(1 - x_{chkm}^\gamma) + M v_{hk}^\tau, \quad \forall i, \forall h > i, k, j, c, m \\ \alpha_{cij} \cdot (\xi t_{ij}^\gamma + x_{cijm}^\gamma \cdot \xi p_{cijm}^\gamma) \leq \xi t_{hk}^\gamma \cdot \alpha_{chk} + M(1 - z_{cjh}^\gamma) + M v_{cij}^\tau + M(1 - \alpha_{chk}) \\ + M(1 - x_{chkm}^\gamma) + M(1 - x_{cijm}^\gamma), \quad \forall i, \forall h > i, k, j, c, m \\ z_{cjh}^\gamma \leq \max_{j=1}^{N_i} \{\alpha_{cij}\} \cdot \max_{k=1}^{N_i} \{\alpha_{chk}\} \end{cases} \quad (8)$$

$$\begin{cases} \xi t_{ij}^{\gamma} \leq lt_{ij}^{\gamma} \\ et_{ij}^{\gamma} \leq \xi t_{ij}^{\gamma} \end{cases}, \quad \forall i, j \quad (9)$$

$$\begin{cases} \xi t_{ij}^{\gamma} \leq \xi t_{ij}^{\tau} + M(1 - v_{ij}^{\tau}) \\ \xi t_{ij}^{\tau} \leq \xi t_{ij}^{\gamma} + M(1 - v_{ij}^{\tau}) \end{cases}, \quad \forall i, j \quad (10)$$

$$x_{cijm}^{\gamma} \leq \alpha_{cij}, \quad \forall i, j, c, m \quad (11)$$

$$\begin{aligned} \xi t_{ij}^{\gamma} \geq 0, \quad x_{cijm}^{\gamma}, z_{cjh}^{\gamma}, v_{ij}^{\tau} \in [0, 1], \quad \forall i, j, k, m, c \\ \xi t_{ij}^{\gamma}, \xi t_{ij}^{\tau} \geq 0, \xi p_{cijm}^{\tau}, \xi p_{cijm}^{\gamma} \geq 0, \alpha_{cij}, b_{cijm}^{\tau}, x_{cijm}^{\gamma}, z_{cjh}^{\gamma}, v_{ij}^{\tau} \in [0, 1], \quad \forall i, j, k, m, c \end{aligned} \quad (12)$$

Eq. (1) minimizes the sum of jobs' tardiness as the objective function. Eq. (2) indicates that each operation has to be allocated to a machine in each stage of the sequence. Eq. (3) ensures that there is a specified boundary between the original and the reactive scheduling. Eq. (4) shows that the intervals of each operation's start time of a job do not have overlap. Eq. (5) guarantees that the operation's start time can be affected by disruptions occurrence. Eq. (6) displays that the operation sequence of a job has priority. Eqs. (7) and (8) show that similar operations of each job on a machine have priority. Eq. (9) indicates that the operation's start time of a job is a uniform random variable which is between the earliest and latest start time. Eq. (10) represents that operations' start times of jobs, which are carried out in the original scheduling time-interval, are not allowed to be changed in the reactive scheduling. Eq. (11) assures that a machine will be allocated only to the operation of a job that requires to be processed. Eq. (12) indicates that the start time of operations can acquire positive values, and a series of variables are binary.

According to the proposed linearization approach by Bagheri and Bashiri [45],  $\beta_i$  is defined as (13).

$$\beta_i = \max(0, \xi t_{iN_i}^{\gamma} + \sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_i m}^{\gamma} x_{CiN_i m}^{\gamma} - d_i^{\gamma}), \quad \forall i \quad (13)$$

The constraints (14) and (15) should be added to the model.

$$\beta_i \geq 0 \quad \forall i \quad (14)$$

$$\beta_i \geq \xi t_{iN_i}^{\gamma} + \sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_i m}^{\gamma} x_{CiN_i m}^{\gamma} - d_i^{\gamma}, \quad \forall i \quad (15)$$

Finally, the linearized model will be as bellow, and Eq. (1) will be converted to (16).

$$\text{Min} \sum_{i=1}^{J^{\gamma}} \beta_i \quad (16)$$

Now a numerical instance is solved by the exact method applying LINGO software. Assume a factory that produces and repairs household appliances by receiving orders. This factory has four



main stations including machining, milling, pressing, and casting stations, and uses a job shop system to produce and repair components. In the first half of January 2020, it has received an order of manufacturing four components; according to the contract, this factory is supposed to deliver the order within 22 days (from 2020/02/01 up to 2020/02/22). Table 1 provides more details about this order.

Please insert here Table 1

The mentioned factory has six machines in the pressing station, the machining station has consisted of two parallel identical machines, and each milling and casting station includes one machine. Fig. 2 illustrates the layout of the aforementioned stations.

Please insert here Fig. 2

For instance,  $m_{21}$  refers to the first machine of the second stage in the above figure. There are four operations in the discussed factory that each of them has to be performed by its specified machine. According to the demand, some operations have to be performed on each component. Therefore, the processing route of each component may be different compared to other components. Operations sequence on components have priority and is defined as follows:

$$i_1 : C_1 - C_2 - C_3 - C_4$$

$$i_2 : C_2 - C_4 - C_3$$

$$i_3 : C_4 - C_3 - C_2$$

$$i_4 : C_1 - C_3 - C_2$$

The setup times during changing the components are independent and include process time. Each operation of each job is allocated to a machine in a specified stage, and the job sequence on machines has priority, but the job is on the priority that has closer delivery time. By considering experts' opinions, the deterministic start times and operations processes are provided in Tables 2 and 3.

Please insert here Table 2

Please insert here Table 3

To predict in the original scheduling, some other binary parameters are required which have to be specified by the experts' opinion. Since reactive scheduling depends on the original scheduling and the way that disruptions occur, efficient and practical scheduling is affiliated with the correct decisions which are made by experts. The parameters, which have to be determined by experts, are as follows:

- Does the  $j^{th}$  operation need to the  $i^{th}$  job in the  $c^{th}$  stage?
- Is the  $m^{th}$  machine allocated to the  $j^{th}$  operation of  $i^{th}$  job in the  $c^{th}$  stage?

The answers to the above questions are provided in Tables 4 and 5. It should be noted that if their answers are positive, they will get 1, and otherwise will get 0.

Please insert here Table 4

Please insert here Table 5

The manufacturing and repairing process of this factory may face two fundamental internal and external disruptions:

Internal environment: Due to misuse of  $m_{31}$  machine by an operator, this machine will be broken and requires two complete days of maintenance. For this purpose, the inaccessibility time of 1<sup>st</sup> machine in the 3<sup>rd</sup> stage is indicated in Table 6 as follows:

Please insert here Table 6

External environment: Customers of first and third orders want to receive their orders on the fourth day, but the factory has changed their delivery dates to 12<sup>th</sup> and 17<sup>th</sup> days, respectively. Given the two abovementioned disruptions, the processing time of  $O_{ij}$  on  $m^{th}$  machine in the  $c^{th}$  stage, when the machine  $m^{th}$  is not available or time-interval  $(ep_{cijm}^{\gamma}, lp_{cijm}^{\gamma})$  has revised in the  $\gamma^{th}$  period, as has been shown in Table 7.

Please insert here Table 7

Now, a small-sized instance is illustrated and solved using the proposed model as follows:

Please insert here Fig. 3

$$J = 4 \rightarrow i, h : 1, 2, 3, 4, C = 4 \rightarrow c : 1, 2, 3, 4$$

$$N_1 = 4, N_2 = 3, N_3 = 3, N_4 = 3$$

$$M_1 = 1, M_2 = 2, M_3 = 1, M_4 = 2$$

$$\tau = 22 \rightarrow j^{22} = \{i_1, i_2, i_3, i_4\}$$

Orders' delivery date of  $\tau^{th}$  period are as Table 1.  $\xi t_{ij}^{\tau}$  and  $\xi p_{ijm}^{\tau}$  are two random parameters that have a uniform distribution.

Please insert here Table 8

$$\xi t_{ij}^{\tau} = et_{ij}^{\tau} + (lt_{ij}^{\tau} - et_{ij}^{\tau})R \quad (17)$$

$$\xi p_{cijm}^{\tau} = ep_{cijm}^{\tau} + (lp_{cijm}^{\tau} - ep_{cijm}^{\tau})R \quad (18)$$

Eq. (17) shows that operation start times of  $\tau^{th}$  period are a random variable.  $et_{ij}^\tau$  and  $lt_{ij}^\tau$  show the earliest and latest operation start time that will be replaced by values of Table 2. Eq. (18) indicates that the operation process time of  $\tau^{th}$  period is a random variable;  $ep_{cijm}^\tau$  and  $lp_{cijm}^\tau$  represent the earliest and latest process time of operation, respectively;  $ep_{cijm}^\tau$  and  $lp_{cijm}^\tau$  will be replaced based on the values of Table 3. In this section, random values, which are between 0 and 1 with a uniform distribution are generated as like as Table 8.

For simpler scheduling, the integer part of random variables of the operation's start time and process time will be considered. Given that  $\xi t_{ij}^\tau$  and  $\xi p_{cijm}^\tau$  are two random variables between deterministic interval-time of  $\tau^{th}$  period, and by considering that start time and operation process are random variables, two following policies are adopted:

- One time, the original scheduling will be predicted by using the earliest operation start time and operation process which exist in Tables 2 and 3, respectively.
- One another time, the original scheduling will be predicted by using the latest operation start time and operation process which exist in Tables 2 and 3, respectively.

In order to deal with the machine breakdown and the modification of orders' delivery date, the following steps are applied:

**Step 1:** Consider that two above disruptions occurred on the 4<sup>th</sup> day, all of the operations of jobs had occurred according to the earliest scheduling up to the 4<sup>th</sup> day of the  $\tau^{th}$  period; due to stopping the processing operations are not allowed, all of the operation, which their start time is before the 4<sup>th</sup> day, will not be stopped. It should be noted that machines will become free when the operation processes finish. For instance,  $O_{22}$ ,  $O_{34}$ ,  $O_{41}$  and  $O_{11}$  have been processed based on the original scheduling, because their start times were before the 4<sup>th</sup> day.

**Step 2:** For the operation, that their start times are after the fourth day and 2 above disruptions have occurred, process time will be changed due to machine breakdown.

**Step 3:** For the remained operations of each job in the reactive scheduling, multiple machine allocation process to each job is considered in order to allocate one another machine in the same step instead of the broken machine; it is worth mentioning that machines allocation to operations is conducted by considering the new delivery dates and objective function. In this instance, the subsequent of operation process for each job will be considered for remaining operations:

- Various modes for machines allocating to remained operations of the first job:

$$i_1 : \begin{cases} m_{21} - m_{31} - m_{41} \\ m_{22} - m_{31} - m_{41} \\ m_{21} - m_{31} - m_{42} \\ m_{22} - m_{31} - m_{42} \end{cases}$$

- Various modes for machines allocating to remained operations of the second job:

$$i_2 : \begin{cases} m_{41} - m_{31} \\ m_{42} - m_{31} \end{cases}$$

- Various modes for machines allocating to remained operations of the third job:

$$i_3 : \begin{cases} m_{31} - m_{21} \\ m_{31} - m_{22} \end{cases}$$

- Various modes for machines allocating to remained operations of the fourth job:

$$i_4 : \begin{cases} m_{31} - m_{21} \\ m_{31} - m_{22} \end{cases}$$

**Step 4:** In the case of happening disruption in the machines allocation process to each job, another high priority process can be replaced.

Based on the obtained results, Fig. 4 shows the earliest time scheduling of the  $\tau^{th}$  period and the dealing way of reactive scheduling to the happened disruptions.

Please insert here Fig. 4

Based on Fig. 4, total tardiness is equal to 4. In addition, Fig. 5 demonstrates the Gantt chart for the latest time scheduling of the  $\tau^{th}$  period and the dealing way of reactive scheduling to the happened disruptions.

Please insert here Fig. 5

## 4. Solution approach

### 4.1. The proposed GA

Since the discussed problem is strongly NP-Hard, it can not be solved on medium and large scales using the exact method in a reasonable time. Therefore, the following hybrid procedure are proposed based on the GA to solve the problem on the practical scales.

**Step 1:** one of the most important issues in designing a meta-heuristic algorithm is the way that its constraints are defined. This hybrid approach forms the structure of the chromosomes according to some of the mathematical model's constraints. This method decreases the CPU time of the algorithm due to preventing the generating of defective chromosomes using a penalty strategy. In the other words, the obtained infeasible solutions during the search process are considered in the penalty strategy; consequently, a fitness function will be created which includes the main objective function and penalty function, and the problem will be converted to an optimization problem without constraints.

$$\left. \begin{array}{l} \text{Min } Z \\ \text{s.t. :} \\ \text{Constraints} \end{array} \right\} \rightarrow \begin{array}{l} \text{Min } \hat{Z} = Z + p(v) \\ p(v) : \text{Penalty function} \end{array} \quad (19)$$

There are lots of penalty functions; multiplicative penalty function is used in this problem which is defined as Eq. (20):

$$\hat{z} = z \cdot (1 + \beta v) \quad (20)$$

The existing  $\beta$  in the above equation is the coefficient of penalty which is a constant number; it is determined based on the importance degree of constraints. In the following section, the penalty's coefficient of each defined mathematical model's constraints is determined; therefore, a specified amount of penalty will be added to the fitness function while a specific constraint is not satisfied. The penalty of each constraint is defined by the following equations:

$$\text{Violation}\{g = g_0\} = \left| \frac{g}{g_0} - 1 \right| \quad (21)$$

$$\text{Violation}\{g \leq g_0\} = \max \left( \frac{g}{g_0} - 1, 0 \right) \quad (22)$$

$$\text{Violation}\{g \geq g_0\} = \max \left( 1 - \frac{g}{g_0}, 0 \right) \quad (23)$$

Provided Eqs. (A1) to (A19) in the appendix is added to the main objective function, and as mentioned, the concerning problem will be converted to a problem without constraint. Then, the fitness function has to be maximized. As mentioned, the multiplicative penalty function is used in this problem that by considering Eq. (20) will be written as follows:

$$\text{Min}[\sum_{i=1}^{J^\gamma} \text{Max}(0, (\xi t_{iN_i}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_i m}^\gamma x_{CiN_i m}^\gamma - d_i^\gamma))] [1 + 1000(\text{Violation1} + \dots + \text{Violation19})]$$

In order to convert the minimum fitness function into the maximum fitness function, the bellow steps have to be considered:

$$\text{Max} \frac{1}{[\sum_{i=1}^{J^\gamma} \text{Max}(0, (\xi t_{iN_i}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_i m}^\gamma x_{CiN_i m}^\gamma - d_i^\gamma))] [1 + 1000(\text{Violation1} + \dots + \text{Violation19})]}$$

**Step 2:** The existing decision variables in the mathematical model, such as  $z_{cih}^\gamma$ ,  $\xi t_{ij}^\gamma$ ,  $x_{cijm}^\gamma$  and  $v_{ij}^\tau$ , form the structure of chromosomes as two-dimensional, three-dimensional, and four-dimensional matrices. It should be noted that all of the structure of the chromosomes has to be

converted into two-dimensional matrices. In this section, the chromosome structure of the current problem will be defined.

- **Structure of chromosome  $v_{ij}^\tau$**

Chromosome  $v_{ij}^\tau$  is formed of a random  $4 \times 4$  matrix that includes binary numbers. Reactive scheduling is applied in order to respond to the original schedule's disruptions. As a result, reactive scheduling is always run after predictive scheduling. In this case, if a gene obtained a random value of 1, all of the prior genes have to obtain 1. Table 9 shows a relevant matrix of chromosome  $v_{ij}^\tau$ .

Please insert here Table 9

- **Structure of chromosome  $x_{cilm}^\gamma$**

Chromosome  $x_{cilm}^\gamma$  is formed of a random  $16 \times 6$  matrix that includes binary numbers. By considering the assumed sequence, the second, third, and fourth jobs do not have a fourth operation. Therefore,  $O_{24}, O_{34}$  and  $O_{44}$  genes have to be equal to zero in order to not allocate machines to them. The probability that a gene gets 1 is dependent to the  $\alpha_{cij}$ . Each operation for each job needs only one step; consequently, that operation gets 1, and others get 0. While there is only a machine in a step and by considering that the operation requires the discussed step, the aforementioned machine obtains 1, and the rest of the machines acquire 0. However, when there is more than one machine in a step and by considering that the operation requires the discussed step, a machine has to be allocated to that operation; so if the first gene acquires the value of 1 randomly, the second gene of the same step has to be 0. Due to being processed, none of the operations that have been allocated in the main scheduling must have to be considered in the reactive scheduling. Due to being processed according to the latest process time as Table 10 shows, no machines are not allocated to the  $O_{21}, O_{22}, O_{31}$  and  $O_{41}$  operations in the structure of chromosome  $x_{cilm}^\gamma$ .

Please insert here Table 10

- **Structure of chromosome  $z_{cjh}^\gamma$**

Chromosome  $z_{cjh}^\gamma$  is formed of a random  $16 \times 4$  matrix that includes binary numbers. Considering the sequence of the jobs, the relevant genes to the jobs, which have the same stages, have to be set to 0; but in lieu of  $h > i$ , genes of them can get 0 and 1 randomly. While two abovementioned jobs require to be processed in the same stage and at the same time, the genes acquire 1; otherwise, the aforementioned genes obtain 0. In lieu of  $i \geq h$ , all of genes acquire 0. Table 11 demonstrates the structure of chromosome  $z_{cjh}^\gamma$ .  $S_{ih}$  indicates that  $i^{th}$  job is preferred to  $h^{th}$ .

Please insert here Table 11

- **Structure of chromosome  $\xi t_{ij}^\gamma$**

Chromosome  $\xi t_{ij}^\gamma$  is formed of a random  $4 \times 4$  matrix that includes binary numbers. The specified intervals are considered based on probabilistic intervals in order to prevent generating random numbers out of control limits. A random interval between (0, 5) has to be generated for the operation start time of each job, and the operation start time of the remained jobs have to acquire the following value:

*operation start time of the remained jobs = (a random integer number of previous operation start time + process time of previous operation, a random integer number of previous operation start time + process time previous operation + 5)*

Since genes of chromosome  $v_{ij}^\tau$  acquire 1, it shows that predictive scheduling is running. Therefore, the relevant start time of this chromosome has to be calculated based on the predefined values of Table 2 and relevant process time has to be calculated according to Table 3. However, if genes of  $v_{ij}^\tau$  chromosome acquire 0, it indicates that reactive scheduling is running; consequently, related start time will not change and process time has to be calculated based on Table 7. It should be noted that the generated chromosome  $\xi t_{ij}^\gamma$  is provided in Table 12.

Please insert here Table 12

**Step 3:** In this step, the way that the initial population will be generated is described. It should be noted that the number of initial population is one of the most important parameters; on the other hand, it depends on several different criteria. One of the most important criteria, which determines the initial population, is the length of the longest Chromosome. The number of initial population in the proposed GA is calculated based on Eq. (24):

$$P_0 = 2^{bl} \quad (24)$$

Due to the repetition of genes in some chromosomes, it is better to consider the chromosomes' number of population less than the number of permutations of 0 and 1 in a chromosome. Consequently, there are two modes for each gene. If  $l$  shows the length of the chromosome, which is calculated by multiplying the number of variables in the objective function and the number of genes that can obtain 1 and zero, is equal to  $16 \times 2$  in the aforesaid

instance; on the other hand,  $b$  is a fixed parameter that is equal to 0.2 in this instance. Therefore,  $P_0$  value will be calculated as follows:

$$P_0 = 2^{0.2 \times 32} = 2^6 = 64$$

As a result, 64 chromosomes will be generated randomly for each chromosome; but before entering the main body of the proposed algorithm, some operations have to be carried out. All of these chromosomes are generated 5 times and the chromosomes which impose the penalty to the objective function, are eliminated. Then, feasible chromosomes, which have been produced in the initial population, will be enhanced by using genetic operators in order to find an optimal solution.

**Step 4:** The roulette wheel is used in this study as the following:

First, the fitness percentage of feasible chromosomes is measured through Eq. (25):

$$([v_{ij}^{\tau}, x_{cijm}^{\gamma}, \xi t_{ij}^{\gamma}, z_{cih}^{\gamma}] \frac{\text{Fitness function value for each feasible chromosome}}{\text{sum of all of the function values of the feasible chromosomes}}) \times 100 \quad (25)$$

After that, the numbers of involved chromosomes in the crossover operator are determined. It should be noted that some of these chromosomes must participate in the crossover operator. The number of involved chromosomes in the crossover operator is calculated through Eq. (26):

$$N_c = 2 \times \text{round} \left( \frac{P_0 \times P_c}{2} \right) \quad (26)$$

$P_0$  = The number of the initial population.

$P_c$  = The crossover operator ratio (it is usually about 0.6).

$N_c$  = The number of participated chromosomes in the crossover operator.

$N_s$  = Counter of selected chromosomes for crossover operator.

The chromosomes that participate in the crossover operator: As the next generation is supposed to be better than the previous generation, the chromosomes, which have a higher percentage of fitness, have a higher probability to participate in the crossover operator. Therefore, the roulette wheel method is selected to determine which chromosomes are involved in transplant reproduction operations. In accordance with the fitness value of each chromosome, chromosomes will participate in the crossover operator.

**Step 5:** The proposed algorithm uses a simple one-point crossover to create new offspring. If they are feasible, they will be kept for the next step; otherwise, they will be eliminated.

**Step 6:** The considered mutation operator is like the mutation in genetic science, which rarely occurs in some chromosomes. In this study, offspring chromosome genes that have been created from the crossover operator will be mutated with the probability of 0.2. Due to being binary  $x_{cijm}^{\gamma}$



,  $v_{ij}^t$ ,  $z_{cjh}^y$  chromosomes are binary, the mutation operator will be performed; but in order to mutate the  $\xi_{ij}^y$  chromosome, which is an integer, a new random number has to be generated according to the existing intervals in Table 12. Finally, the mutant chromosome must be checked; the defective invalid chromosomes have to be removed.

**Step 7:** In this step, members' new population will be selected. All feasible chromosomes are classified based on their fitness values in descending. Therefore, the first  $p$  chromosomes of the combined population, that have higher fitness values, will be selected as the new population ( $p$  is equal to the number of each generation's population). The abovementioned operators are repeated up to where the optimal solution will be acquired. Fig. 6. demonstrates the proposed GA steps in a chart as follows.

Please insert here Fig. 6

## 4.2. The proposed GA-PSO

This section represents the hybrid proposed solution approach based on Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) the so-called GA-PSO in this paper. These two evolutionary algorithms, which are based on exploration and exploitation, have been proposed for different types of stochastic JSS problems [46]. One-point crossover operator of GA will be combined with the operator of PSO in order to form the evolutionary operator; the GA's operator is based on exploration, and the PSO's operator is based on exploitation. It should be noted that the generated chromosomes in the exploitation-based operator will be replaced by the parent ones if generated ones are better than the parent chromosomes. The evaluation process has been provided as follows.

**Step 1:**  $(vel.)_{ij}^t$  shows the velocity for  $i^{th}$  gene of  $j^{th}$  chromosome in the  $t^{th}$  generation will be determined as follows:

$$(vel.)_{ij}^t = \varphi(vel.)_{ij}^{t-1} + \mathcal{G}_1(g_i - x_{ij}^{t-1}) + \mathcal{G}_2(l_i^{t-1} - x_{ij}^{t-1}), \quad \forall i, j \quad (27)$$

$\varphi$ ,  $b_l$  and  $b_g$  are predefined parameters in the Eq. (27).  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are equal to  $r_1 \cdot b_g$  and  $r_2 \cdot b_l$ , respectively. It should be noted that  $r_1$  and  $r_2$  are random numbers between 0 and 1.  $g_i$  is equal to the  $i^{th}$  gene's value of the best chromosome, and  $l_i^t$  is equal to the  $i^{th}$  gene's value of the best chromosome in the  $t^{th}$  generation.

**Step 2:** A new chromosome of  $j'$  will be generated by the following equation:

$$x_{ij'}^t = x_{ij}^t + v_{ij}^t \quad \forall i, j \quad (28)$$

**Step 3:** In this step, the fitness function evaluates the  $j'$  chromosome. If  $j'$  chromosome's fitness value is better than the fitness value of  $j$  chromosome,  $j'$  chromosome will be replaced

instead of  $j$  chromosome by using  $(\text{vel.})_{ij}^t = \left[ (\text{vel.})_{ij}^t \right]$ . As mentioned, exploitation-based operators, and exploration-based operators are considered in the present investigation.

In the proposed GA-PSO, parents are selected through the roulette wheel. Offspring selection can be deterministic (i.e. selection of the best offspring) or as like as the proposed method in the previous section. All of the other steps of the proposed GA-PSO are exactly the same as the steps of the proposed GA. Fig. 7 shows the steps of the proposed GA-PSO in a chart in the following.

Please insert here Fig. 7

## 5. Test instances and results analysis

Given that the small-sized instances solved using LINGO software, the proposed GA and GA-PSO have been coded and applied in MATLAB to solve the same instances for verification. Moreover, the proposed GA and GA-PSO solve medium-size and large-size instances to evaluate their effectiveness and efficiency. Before that, each parameter was tested in three levels: low, medium and high by the Taguchi analysis method. The test values and the final value of each parameter have been presented in Table 13.

Please insert here Table 13

All of the instances have been solved 30 times by a computer with a dual-core processor with 2.1 GHz and 2 GB of Ram and the best results have been reported for 10 problems. We have investigated the problem at hand by solving 5 small-sized instances, 3 medium-sized instances, and 2 large-size instances. The obtained results of small-size instances are provided in Table 14.

Please insert here Table 14

Moreover, the acquired results of medium size and large size instances are provided in Table 15.

Please insert here Table 15

According to the obtained results in Table 14, it is obvious that the three aforementioned methods solve small instances as the same as each other in terms of the objective function. It should be noted that both of the proposed algorithms can solve small size instances 35 percentage (on average) faster than the exact method. In addition, the proposed GA-PSO solves small size instances 1 percentage faster than the proposed GA.

On the other hand, Table 15 shows that the proposed GA-PSO solves instances 10 percentage (on average) better than the proposed GA. Moreover, the proposed GA-PSO can solve medium size and large size instances 4.5 percentage (on average) faster than the proposed GA.

Please insert here Fig. 8

Please insert here Fig. 9

As Figures 8 and 9 show, the proposed GA and GA-PSO can provide efficient solutions and better solution times compared to the exact method. However, the proposed GA-PSO outperformed GA in solving the medium- and large-sized instances in terms of the objective function and CPU Time.

Now, two proposed algorithms are statistically compared via the two-sample T-test at a 95% confidence level. For this purpose, each of instances 1 to 10 has been solved thirty times by GA and GA-PSO and the tardiness value and CPU time are calculated as the performances measure of each algorithm. The outputs provided in Table 16 reveal that the performance of the proposed GA-PSO algorithm is significantly better than the classic GA.

Please insert here Table 16

## 6. Conclusion

In real-world condition, different kinds of disruptions may occur during production that can deteriorate the initial schedule. This paper addressed a FJSSP to minimize the sum of jobs' tardiness considering two common kinds of disruptions including machines breakdown and order due date modification. The problem was formulated as an MILP model based on the reactive approach. Furthermore, two strategies, such as being able to allocate multiple machines to each job, and being able to select the best alternative process from the other job while some disruptions would be created in the processes of a job, were used for dealing with the aforementioned disruptions. Since the discussed problem is strongly NP-Hard, a GA and a GA-PSO were proposed to solve the problem on the practical scales. Obtained results indicated that two proposed metaheuristics can obtain optimum solutions like the mathematical model for the small-sized instances. Moreover, the proposed GA-PSO provided better solutions for medium- and large-sized instances compared to the proposed GA.

Possible extensions of this work include considering more real-world issues and operational constraints such as buffers with specific capacities, transportation constraints, and transportation time and demand. In addition, it is suggested to apply other efficient heuristic or meta-heuristic algorithms for solving the considered problem.

## Reference

- [1] Coffman, E. G. & Bruno, J. L. "Computer and job-shop scheduling theory", John Wiley & Sons (1976).
- [2] Gan, H.-S. & Wirth, A. "Comparing deterministic, robust and online scheduling using entropy", *International journal of production research*, 43, 2113-2134 (2005).
- [3] Gao, J., Gen, M. & Sun, L. "Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm", *Journal of Intelligent Manufacturing*, 17, 493-507 (2006).
- [4] Gao, J., Gen, M., Sun, L. & Zhao, X. "A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems", *Computers & Industrial Engineering*, 53, 149-162 (2007).
- [5] Gao, J., Sun, L. & Gen, M. "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems", *Computers & Operations Research*, 35, 2892-2907 (2008).
- [6] Gen, M., Gao, J. & Lin, L. "Multistage-based genetic algorithm for flexible job-shop scheduling problem", *Intelligent and evolutionary systems*, 187, 183-196 (2009).
- [7] Herroelen, W. & Leus, R. "Project scheduling under uncertainty: Survey and research potentials", *European journal of operational research*, 165, 289-306 (2005).
- [8] Ayyoubzadeh, B., Ebrahimnejad, S., Bashiri, M., Baradaran, V., & Hosseini, S.M.H., "Modeling and an improved NSGA-II algorithm for sustainable manufacturing systems with energy conservation under environmental uncertainties: a case study", *International Journal of Sustainable Engineering*, 14(3), 255-279 (2021).  
<https://doi.org/10.1080/19397038.2021.1923083>
- [9] Ayyoubzadeh, B., Ebrahimnejad, S., Bashiri, M., Baradaran, V., & Hosseini, S.M.H., "Energy-aware reactive flexible job shop scheduling with timely delivery under uncertainty: a case study", *Journal of Industrial Engineering and Management Studies*, 8(2), 26-53 (2021).  
10.22116/JIEMS.2022.138123
- [10] Hasani, A., & Hosseini S.M.H., "Auxiliary resource planning in a flexible flow shop scheduling problem considering stage skipping." *Computers & Operations Research* 138 (2022): 105625.
- [11] Jackson, J. R. "Simulation research on job shop production", *Naval Research Logistics Quarterly*, 4, 287-295 (1957).
- [12] Sabuncuoglu, I. & Bbyiz, M. "Analysis of reactive scheduling problems in a job shop environment", *European Journal of operational research*, 126, 567-586 (2000).
- [13] Chryssolouris, G. & Subramaniam, V. "Dynamic scheduling of manufacturing job shops using genetic algorithms", *Journal of Intelligent Manufacturing*, 12, 281-293 (2001).

- [14] Albers, S. & Schmidt, G. "Scheduling with unexpected machine breakdowns". *Discrete Applied Mathematics*, 110, 85-99 (2001).
- [15] Qi, X., Yu, G. & Bard, J. F. "Single machine scheduling with assignable due dates", *Discrete Applied Mathematics*, 122, 211-233 (2002).
- [16] Suwa, H. & Sandoh, H. "Capability of cumulative delay based reactive scheduling for job shops with machine breakdowns", *Computers & Industrial Engineering*, 53, 63-78 (2007).
- [17] Hao, X. & Lin, L. "Job shop rescheduling by using multi-objective genetic algorithm", *The 40th International Conference on Computers and Industrial Engineering, Japan*, 1-6 (2010).
- [18] Adibi, M., Zandieh, M. & Amiri, M. "Multi-objective scheduling of dynamic job shop using variable neighborhood search", *Expert Systems with Applications*, 37, 282-287 (2010).
- [19] Kianfar, K., Ghomi, S. F. & Jadid, A. O. "Study of stochastic sequence-dependent flexible flow shop via developing a dispatching rule and a hybrid GA", *Engineering Applications of Artificial Intelligence*, 25, 494-506 (2012).
- [20] He, W. & Sun, D.-H. "Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies", *The International Journal of Advanced Manufacturing Technology*, 66, 501-514 (2013).
- [21] Zhang, L., Gao, L. & Li, X. "A hybrid intelligent algorithm and rescheduling technique for job shop scheduling problems with disruptions", *The International Journal of Advanced Manufacturing Technology*, 65, 1141-1156 (2013).
- [22] De Freitas Rodrigues, R., Dourado, M. C. & Szwarcfiter, J. L. "Scheduling problem with multi-purpose parallel machines, . *Discrete Applied Mathematics*, 164, 313-319 (2014).
- [23] He, C., Leung, J. Y.-T., Lee, K. & Pinedo, M. L. "Scheduling a single machine with parallel batching to minimize makespan and total rejection cost", *Discrete Applied Mathematics*, 204, 150-163 (2015).
- [24] Zinder, Y. & Walker, S. "Algorithms for scheduling with integer preemptions on parallel machines to minimize the maximum lateness", *Discrete Applied Mathematics*, 196, 28-53 (2015).
- [25] Rey, G. Z., Bekrar, A., Trentesaux, D. & Zhou, B.-H. "Solving the flexible job-shop just-in-time scheduling problem with quadratic earliness and tardiness costs", *The International Journal of Advanced Manufacturing Technology*, 81, 1871–1891 (2015).
- [26] Rohaninejad, M., Kheirkhah, A., Fattahi, P. & Vahedi-Nouri, B. "A hybrid multi-objective genetic algorithm based on the electre method for a capacitated flexible job shop scheduling problem", *The International Journal of Advanced Manufacturing Technology*, 77, 51-66 (2015).
- [27] Shen, J. & Zhu, Y. "Chance-constrained model for uncertain job shop scheduling problem", *Soft Computing*, 20, 2383–2391 (2016).

- [28] Xie N., Chen N. "Flexible job shop scheduling problem with interval grey processing time", *Applied Soft Computing*, 70, 513-524 (2018).
- [29] Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S. & Xiong, J. "A knowledge-based ant colony optimization for flexible job shop scheduling problems", *Applied Soft Computing*, 10, 888-896 (2010).
- [30] Tavakkoli-Moghaddam, R., Jolai, F., Vaziri, F., Ahmed, P. & Azaron, A. "A hybrid method for solving stochastic job shop scheduling problems", *Applied Mathematics and Computation*, 170, 185-206 (2005).
- [31] Subramaniam, V., Raheja, A. & Rama Bhupal Reddy, K. "Reactive repair tool for job shop schedules", *International Journal of Production Research*, 43, 1-23 (2005).
- [32] Xiuli, W. "Research on flexible job shop dynamic scheduling problem", *Journal of System Simulation*, 20, 3828-3832 (2008).
- [33] Fattahi, P. & Fallahi, A. "Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability", *CIRP Journal of Manufacturing Science and Technology*, 2, 114-123 (2010).
- [34] Lei, D. "A genetic algorithm for flexible job shop scheduling with fuzzy processing time", *International Journal of Production Research*, 48, 2995-3013 (2010).
- [35] Hatami, S., Ebrahimnejad, S., Tavakkoli-Moghaddam, R. et al. "Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times", *Int J Adv Manuf Technol*, 50, 1153-1164 (2010).
- [36] Liu, A.-J., Yang, Y., Xing, Q.-S., Lu, H., Zhang, Y.-D., Zhou, Z.-Y., Wu, G.-H. & Zhao, X.-H. "Dynamic scheduling on multi-objective flexible Job Shop", *Computer Integrated Manufacturing Systems*, 17, 2629-2637 (2011).
- [37] Jolai, F., Rabiee, M. & Asefi, H. "A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times", *International Journal of Production Research*, 50, 7447-7466 (2012).
- [38] Hokim., H., Gyum., K., Young., J. and Park., S. "The mixing process scheduling using particle swarm optimization", *Computers and industrial engineering*, 110, 333-343 (2017).
- [39] Naderi-Beni, M., Ghobadian, E., Ebrahimnejad, S., Tavakkoli-Moghaddam, R. "Fuzzy bi-objective formulation for a parallel machine scheduling problem with machine eligibility restrictions and sequence-dependent setup times", *International Journal of Production Research*, 52, 5799-5822, (2014).
- [40] Shen, L., Dauzere-Pares, S., Neufeld, J.S. "Solving the flexible job shop scheduling problem with sequence-dependent setup times", *European Journal of Operational Research*, 265, 503-516 (2018).

- [41] Jamrus, T., Chien, C., Gen, M. and Sethanan, K. "Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing", IEEE Transactions on Semiconductor Manufacturing, 31(1), 32-41, (2018).
- [42] Soleimani Nia, F., & Mehdizadeh, E. "Presenting a Mathematical Model for Flexible job-shop Production Scheduling Problem with Reverse Flows and Solving it by Using Genetic Algorithm", Journal of Industrial Engineering (Journal of Faculty of Engineering), 52(1), 87-96, (2018).
- [43] Huang, X., Guan, Z., Yang, L. "An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem", Advances in Mechanical Engineering, 10(9), 1-14, (2018).
- [44] Liu, Z., Guo, S., & Wang, L. "Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption", Journal of Cleaner Production, 211, 765-786 (2019).
- [45] Bagheri, M. & Bashiri, M. "A new mathematical model towards the integration of cell formation with operator assignment and inter-cell layout problems in a dynamic environment", Applied Mathematical Modelling, 38, 1237-1254 (2014).
- [46] Lin, L. & Gen, M. "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation", Soft Computing, 13, 157-168 (2009).

## Appendix

Based on the provided Eqs. (21)-(23), penalty of the mathematical model's constraints are determined as follows:

$$Violation1\{g = g_0\} = \left| \frac{\sum_{c=1}^C \sum_{m=1}^{M_c} x_{cijm}^{\gamma} \cdot \alpha_{cij}}{1 - v_{ij}^{\tau}} - 1 \right| \quad (A1)$$

$$Violation2\{g < g_0\} = \max \left( \frac{\xi t_{ij}^{\tau}}{\xi D^{\tau} + M \cdot (1 - v_{ij}^{\tau})} - 1, 0 \right) \quad (A2)$$

$$Violation3\{g \leq g_0\} = \max \left( \frac{\xi D^{\tau}}{\xi t_{ij}^{\tau} + M \cdot v_{ij}^{\tau}} - 1, 0 \right) \quad (A3)$$

$$Violation4\{g \leq g_0\} = \max \left( \frac{lt_{ij}^{\tau} + \sum_{c=1}^C \sum_{m=1}^{M_c} lp_{cijm}^{\tau} b_{cijm}^{\tau}}{et_{i(j+1)}^{\gamma} + M \cdot (1 - v_{ij}^{\tau}) + M \cdot v_{i(j+1)}^{\tau}} - 1, 0 \right) \quad (A4)$$

$$Violation5\{g \leq g_0\} = \max \left( \frac{lt_{ij}^\gamma + \sum_{c=1}^C \sum_{m=1}^{M_c} x_{cijm}^\gamma \cdot lp_{cijm}^\gamma}{et_{i(j+1)}^\gamma + M \cdot v_{ij}^\tau} - 1, 0 \right) \quad (A5)$$

$$Violation6\{g \leq g_0\} = \max \left( \frac{\xi D^\tau + T_{cm}^\gamma}{et_{ij}^\gamma + M \cdot (1 - x_{cijm}^\gamma) + M \cdot v_{ij}^\tau} - 1, 0 \right) \quad (A6)$$

$$Violation7\{g \leq g_0\} = \max \left( \frac{\xi t_{ij}^\tau + \sum_{c=1}^C \sum_{m=1}^{M_c} b_{cijm}^\tau \cdot \xi p_{cijm}^\tau}{\xi t_{i(j+1)}^\gamma + M \cdot (1 - v_{ij}^\tau) + M \cdot v_{i(j+1)}^\tau} - 1, 0 \right) \quad (A7)$$

$$Violation8\{g \leq g_0\} = \max \left( \frac{\xi t_{ij}^\gamma + \sum_{m=1}^{M_c} x_{cijm}^\gamma \cdot \xi p_{cijm}^\gamma}{\xi t_{i(j+1)}^\gamma + M \cdot v_{ij}^\tau} - 1, 0 \right) \quad (A8)$$

$$Violation9\{g \leq g_0\} = \max \left( \frac{\alpha_{chk} \cdot (\xi t_{hk}^\tau + b_{chkm}^\tau \cdot \xi p_{chkm}^\tau)}{\xi t_{ij}^\gamma \cdot \alpha_{cij} + M \cdot (1 - \alpha_{cij}) + M \cdot (1 - x_{cijm}^\gamma) + M \cdot (1 - b_{chkm}^\tau) + M \cdot (1 - v_{hk}^\tau) + M \cdot v_{ij}^\tau} - 1, 0 \right) \quad (A9)$$

$$Violation10\{g \leq g_0\} = \max \left( \frac{\alpha_{chk} \cdot (\xi t_{hk}^\gamma + x_{chkm}^\gamma \cdot \xi p_{chkm}^\gamma)}{\xi t_{ij}^\gamma \cdot \alpha_{cij} + M \cdot z_{cih}^\gamma + M \cdot (1 - \alpha_{cij}) + M \cdot (1 - x_{cijm}^\gamma) + M \cdot (1 - x_{chkm}^\gamma) + M \cdot v_{hk}^\tau} - 1, 0 \right) \quad (A10)$$

$$Violation11\{g \leq g_0\} = \max \left( \frac{\alpha_{cij} \cdot (\xi t_{ij}^\gamma + x_{cijm}^\gamma \cdot \xi p_{cijm}^\gamma)}{\xi t_{hk}^\gamma \cdot \alpha_{chk} + M \cdot (1 - z_{cih}^\gamma) + M \cdot v_{cij}^\tau + M \cdot (1 - \alpha_{chk}) + M \cdot (1 - x_{cijm}^\gamma) + M \cdot (1 - x_{chkm}^\gamma)} - 1, 0 \right) \quad (A11)$$

$$Violation12\{g \leq g_0\} = \max \left( \frac{z_{cih}^\gamma}{\max_{j=1}^{N_i} \{\alpha_{cij}\} \cdot \max_{k=1}^{N_i} \{\alpha_{chk}\}} - 1, 0 \right) \quad (A12)$$

$$Violation13\{g \leq g_0\} = \max \left( \frac{\xi t_{ij}^\gamma}{lt_{ij}^\gamma} - 1, 0 \right) \quad (A13)$$

$$Violation14\{g \leq g_0\} = \max \left( \frac{et_{ij}^\gamma}{\xi t_{ij}^\gamma} - 1, 0 \right) \quad (A14)$$

$$Violation15\{g \leq g_0\} = \max \left( \frac{\xi t_{ij}^\gamma}{\xi t_{ij}^\tau + M \cdot (1 - v_{ij}^\tau)} - 1, 0 \right) \quad (A15)$$

$$Violation16\{g \leq g_0\} = \max \left( \frac{\xi t_{ij}^\tau}{\xi t_{ij}^\gamma + M \cdot (1 - v_{ij}^\tau)} - 1, 0 \right) \quad (A16)$$



$$Violation17\{g \leq g_0\} = \max \left( \frac{x_{cijm}^\gamma}{\alpha_{cij}} - 1, 0 \right) \quad (A17)$$

$$Violation18\{g \leq g_0\} = \max \left( 1 - \frac{\beta_i}{\sum_{c=1}^C \sum_{m=1}^{M_c} \xi p_{CiN_im}^\gamma x_{CiN_im}^\gamma - d_i^\gamma}, 0 \right) \quad (A18)$$

$$Violation19\{g \leq g_0\} = \max (-\beta_i, 0) \quad (A19)$$

## Biography

**Behnam Ayyoubzadeh** received his Ph.D. degree from the Islamic Azad University, North Tehran Branch in the industrial engineering field in 2021. He received his B.Sc. in Industrial Engineering from the Islamic Azad University, Karaj Branch in 2011. Also, He received his M.Sc. in Industrial Engineering from the Islamic Azad University, Karaj Branch in 2013. His main areas of research are the modelling of scheduling (especially job shop) and production planning and control. He published more than 10 papers in reputable academic journals and conferences.

**Sadoullah Ebrahimnejad** is an Associate Professor of Industrial Engineering at Islamic Azad University – Karaj Branch in Iran. He received his B.Sc. degree from Iran University of Science & Technology in 1986, M.Sc. degree from Amirkabir University of Technology in 1993, and Ph.D. degree from Islamic Azad University - Science & Research Branch in Tehran in 2001. His research interests are Fuzzy MADM/MODM, SCM, Operations management, Optimization models, Risk management, Multi-criteria network optimization. He published more than 90 papers in reputable academic journals and conferences.

**Mahdi Bashiri** is a Professor of Industrial Engineering at Shahed University. He holds a B.Sc. in Industrial Engineering from Iran University of Science and Technology, M.Sc. and Ph.D. from Tarbiat Modares University. He is a recipient of the 2013 young national top scientist award from the Academy of Sciences of the Islamic Republic of Iran. He serves as the Editor-in-Chief of Journal of Quality Engineering and Production Optimization published in Iran and the Editorial Board member of three reputable academic journals. His research interests are facilities design, meta-heuristics, and multi-response optimization. He has published about 10 books and more than 190 papers in reputable academic journals and conferences.

**Vahid Baradaran** received his Ph.D. degree from Tarbiat Modares University in the industrial engineering field in 2010. He is an associate professor at Islamic Azad University, Tehran North Branch. Since he worked in university, he taught many courses in Operation Research, Statistics, Decision Making, Multivariate Statistical Analysis, and Data Mining in Master Science, and PhD courses.

**Seyed Mohammad Hassan Hosseini** is currently principal and assistant professor of Industrial Engineering College at Shahrood University of Technology, where he has been since 2013. He received a B.Sc. from Iran University of Science and Technology in 2000, and an M.Sc. from the Amirkabir University of Technology. He received his Ph.D. in Industrial Engineering from the Payam-e-Noor University of Tehran in 2012. From 2002 to 2008 he worked at Iran Khodro Company (IKCO) as the most famous car manufactures in Middle East as an expert of quality control and also CRM department.

His main areas of research are the modelling of scheduling (especially flow shop) and production planning and control, quality engineering and management, and decision making techniques. He has published over than 40 papers in different international journals such as Journal of applied mathematical modelling, The International Journal of Advanced Manufacturing Technology, International Journal of Industrial Engineering Computations, International Journal of Supply and Operations Management, Journal of Optimization in Industrial Engineering, Journal of Industrial and Systems Engineering, Journal of applied mathematics and computations, Environment Systems and Decisions, International Journal of Productivity and Quality Management and so on.

As of 2014, Google Scholar reports over than 200 citations to his work. He has given numerous invited talks and tutorials, and is a founder of and consultant to companies involved in car making.

## Tables

**Table 1.** Components delivery date to customers.

**Table 2.** Deterministic interval of operation start time for the period of  $\tau^{th}$ .

**Table 3.** Deterministic interval of operation process time for the period of  $\tau^{th}$ .

**Table 4.** Pre-need steps of the jobs' operation.

**Table 5.** Machines allocation to the jobs' operation for the period of  $\tau^{th}$ .

**Table 6.** Machines' unavailability duration in each period of  $\gamma^{th}$ .

**Table 7.** Deterministic interval of operation process for the period of  $\gamma^{th}$ .

**Table 8.** Random Values of R.

**Table 9.** Formation of  $v_{ij}^{\tau}$  chromosomes.

**Table 10.** Formation of  $x_{cijm}^{\gamma}$  chromosomes.

**Table 11.** Formation of  $z_{cjh}^{\gamma}$  chromosomes.

**Table 12.** Formation of  $\xi t_{ij}^{\gamma}$  chromosomes.

**Table 13.** Parameters setting of GA-PSO algorithm.

**Table 14.** Obtained results for small size problem through Exact Method, GA and GA-PSO.

**Table 15.** Obtained results for medium size and large scale problem through Exact Method, GA and GA-PSO.

**Table 16.** Results of statistical compare performances.

## Figures

**Fig. 1.** Reactive scheduling process.

**Fig. 2.** Machines' placement over factory.

**Fig. 3.** Repair and production process of  $\tau^{th}$  period.

**Fig. 4.** Reactive scheduling in contrast with disruptions during production based on earliest time scheduling.

**Fig. 5.** Reactive scheduling in contrast with disruptions during production based on latest time scheduling.

**Fig. 6.** The proposed GA.

**Fig. 7.** The proposed GA-PSO.

**Fig. 8.** Comparison of the obtained tardiness function value through Exact Method, GA and GA-PSO.  
**Fig. 9.** Comparison of Exact Method, GA and GA-PSO in terms of CPU Time.

### Tables

Table 1. Components delivery date to customers.

i	1	2	3	4
$d_i^\tau$	22	12	13	18

Table 2. Deterministic interval of operation start time for the period of  $\tau^{th}$ .

i \ j	1	2	3	4
1	(2,6)	(8,9)	(11,13)	(16,18)
2	(1,2)	(4,6)	(9,10)	-
3	(0,1)	(4,5)	(8,9)	-
4	(0,1)	(6,8)	(10,11)	-

Table 3. Deterministic interval of operation process time for the period of  $\tau^{th}$ .

c	1	2		3	4	
<div><div>m</div><div><math>O_{ij}</math></div></div>	1	1	2	1	1	2
$O_{11}$	(1,2)	-		-	-	
$O_{12}$	-	(1,2)	(1,2)	-	-	
$O_{13}$	-	-		(2,3)	-	

$O_{14}$	-	-	-	(2,3)	(2,3)
$O_{21}$	-	(1,2)	(1,2)	-	-
$O_{22}$	-	-	-	(1,3)	(1,3)
$O_{23}$	-	-	(1,3)	-	-
$O_{31}$	-	-	-	(2,3)	(2,3)
$O_{32}$	-	-	(2,3)	-	-
$O_{33}$	-	(3,4)	(3,4)	-	-
$O_{41}$	(2,5)	-	-	-	-
$O_{42}$	-	-	(1,2)	-	-
$O_{43}$	-	(2,3)	(2,3)	-	-

Table 4. Pre-need steps of the jobs' operation.

$O_{ij} \backslash c$	1	2	3	4
$O_{11}$	1	0	0	0
$O_{12}$	0	1	0	0
$O_{13}$	0	0	1	0
$O_{14}$	0	0	0	1
$O_{21}$	0	1	0	0
$O_{22}$	0	0	0	1
$O_{23}$	0	0	1	0
$O_{31}$	0	0	0	1
$O_{32}$	0	0	1	0
$O_{33}$	0	1	0	0
$O_{41}$	1	0	0	0
$O_{42}$	0	0	1	0
$O_{43}$	0	1	0	0

Table 5. Machines allocation to the jobs' operation for the period of  $\tau^{th}$ .

c	1	2	3	4
$O_{ij} \backslash m$	1	1   2	1	1   2
$O_{11}$	1	-	-	-
$O_{12}$	-	1   0	-	-

$O_{13}$	-	-	1	-
$O_{14}$	-	-	-	1   0
$O_{21}$	-	1   0	-	-
$O_{22}$	-	-	-	1   0
$O_{23}$	-	-	1	-
$O_{31}$	-	-	-	1   0
$O_{32}$	-	-	1	-
$O_{33}$	-	0   1	-	-
$O_{41}$	1	-	-	-
$O_{42}$	-	-	1	-
$O_{43}$	-	1   0	-	-

Table 6. Machines' unavailability duration in each period of  $\gamma^{th}$ .

m \ c	1	2	3	4
1	0	0	2	0
2	-	0	-	0

Table 7. Deterministic interval of operation process for the period of  $\gamma^{th}$ .

c	1	2	3	4
m \ $O_{ij}$	1	1   2	1	1   2
$O_{11}$	(1,2)	-	-	-
$O_{12}$	-	(1,2)   (1,2)	-	-
$O_{13}$	-	-	(3,5)	-
$O_{14}$	-	-	-	(2,3)   (2,3)
$O_{21}$	-	(2,4)   (2,4)	-	-
$O_{22}$	-	-	-	(2,4)   (2,4)
$O_{23}$	-	-	(1,3)	-
$O_{31}$	-	-	-	(3,4)   (3,4)
$O_{32}$	-	-	(3,4)	-
$O_{33}$	-	(2,3)   (2,3)	-	-
$O_{41}$	(2,5)	-	-	-
$O_{42}$	-	-	(3,4)	-

$O_{43}$	-	(2,3)	(2,3)	-	-
----------	---	-------	-------	---	---

Table 8. Random Values of  $\mathbf{R}$ .

0.875	0.7	0.611	0.777	0.587
0.218	0.271	0.784	0.112	0.503
0.247	0.558	0.001	0.571	0.117
0.318	0.097	0.512	0.651	0.557
0.244	0.214	0.761	0.348	0.672

Table 9. Formation of  $v_{ij}^{\tau}$  chromosomes.

m \ c	1	2	3	4
1	1	1	1	0
2	1	0	0	0
3	1	0	0	0
4	1	1	0	0

Table 10. Formation of  $x_{cijn}^{\gamma}$  chromosomes.

c	1	2		3	4	
m \ $O_{ij}$	1	1	2	1	1	2
$O_{11}$	<u>1</u>	0	0	0	0	0
$O_{12}$	0	<u>1</u>	<u>0</u>	0	0	0
$O_{13}$	0	0	0	<u>1</u>	0	0
$O_{14}$	0	0	0	0	<u>0</u>	<u>1</u>
$O_{21}$	0	0	0	0	0	0
$O_{22}$	0	0	0	0	0	0
$O_{23}$	0	0	0	<u>1</u>	0	0
$O_{24}$	0	0	0	0	0	0
$O_{31}$	0	0	0	0	0	0
$O_{32}$	0	0	0	<u>1</u>	0	0

$O_{33}$	0	<u>1</u>	<u>0</u>	0	0	0
$O_{34}$	0	0	0	0	0	0
$O_{41}$	0	0	0	0	0	0
$O_{42}$	0	0	0	<u>1</u>	0	0
$O_{43}$	0	<u>1</u>	<u>0</u>	0	0	0
$O_{44}$	0	0	0	0	0	0

Table 11. Formation of  $z_{cjh}^{\gamma}$  chromosomes.

$S_{ij} \backslash c$	1	2	3	4
$S_{11}$	0	0	0	0
$S_{12}$	0	<u>1</u>	<u>1</u>	<u>1</u>
$S_{13}$	0	<u>1</u>	<u>1</u>	<u>1</u>
$S_{14}$	<u>1</u>	<u>1</u>	<u>1</u>	0
$S_{21}$	0	0	0	0
$S_{22}$	0	0	0	0
$S_{23}$	0	<u>1</u>	<u>1</u>	<u>1</u>
$S_{24}$	0	<u>1</u>	<u>1</u>	0
$S_{31}$	0	0	0	0
$S_{32}$	0	0	0	0
$S_{33}$	0	0	0	0
$S_{34}$	0	<u>1</u>	<u>1</u>	0
$S_{41}$	0	0	0	0
$S_{42}$	0	0	0	0
$S_{43}$	0	0	0	0
$S_{44}$	0	0	0	0



Table 12. Formation of  $\xi t_{ij}^\gamma$  chromosomes.

i \ j	1	2	3	4
1	(0,5)	$(t_{11} + p_{11}, t_{11} + p_{11} + 5)$	$(t_{12} + p_{12}, t_{12} + p_{12} + 5)$	$(t_{13} + p_{13}, t_{13} + p_{13} + 5)$
2	(0,5)	$(t_{21} + p_{21}, t_{21} + p_{21} + 5)$	$(t_{22} + p_{22}, t_{22} + p_{22} + 5)$	-
3	(0,5)	$(t_{31} + p_{31}, t_{31} + p_{31} + 5)$	$(t_{32} + p_{32}, t_{32} + p_{32} + 5)$	-
4	(0,5)	$(t_{41} + p_{41}, t_{41} + p_{41} + 5)$	$(t_{42} + p_{42}, t_{42} + p_{42} + 5)$	-

Table 13. Parameters setting of GA-PSO algorithm

Parameters	Test values			The best value
	Level 1	Level 2	Level 3	
Iteration	3000	4000	5000	5000
b	0.2	0.3	0.5	0.2
Population size	$5 \times P_0$	$7 \times P_0$	$10 \times P_0$	$5 \times P_0$
$p_c$	0.3	0.6	0.8	0.6
$p_M$	0.2	0.3	0.4	0.2

Table 14. Obtained results for small size problem through Exact Method, GA and GA-PSO.

Scale of problem	Specifications of problem							CPU Time (hour)			Tardiness Value		
	J	$N_i$	C	$M_c$	$\xi D^r$	$T_{cm}^\gamma$	Jobs Sequence	Exact Method	GA	GA-PSO	Exact Method	GA	GA-PSO
Small size	4	$N_1 = 4$ $N_2 = 3$ $N_3 = 3$ $N_4 = 3$	4	$M_1 = 1$ $M_2 = 2$ $M_3 = 1$ $M_4 = 2$	4	$T_{31}^\gamma = 2$	$i_1 : C_1 - C_2 - C_3 - C_4$ $i_2 : C_2 - C_4 - C_3$ $i_3 : C_4 - C_3 - C_2$ $i_4 : C_1 - C_3 - C_2$	0.73	0.98	0.96	4	4	4
	5	$N_1 = 4$	5	$M_1 = 1$	4	$T_{31}^\gamma = 2$	$i_1 : C_1 - C_2 - C_3 - C_4$	1.55	1.09	1.08	6	6	6

		$N_2=3$		$M_2=2$		$T_{51}^{\gamma}=3$	$i_2: C_2-C_4-C_3$ $i_3: C_4-C_3-C_2$ $i_4: C_1-C_3-C_2$ $i_5: C_5-C_4-C_2-C_3$										
		$N_3=3$		$M_3=1$													
		$N_4=3$		$M_4=2$													
		$N_5=4$		$M_5=1$													
	6	$N_1=4$	5	$M_1=1$	4	$T_{31}^{\gamma}=2$	$i_1: C_1-C_2-C_3-C_4$	5.35	3.35	3.30	20	20	20				
		$N_2=3$		$M_2=2$		$T_{51}^{\gamma}=3$	$i_2: C_2-C_4-C_3$ $i_3: C_4-C_3-C_2$ $i_4: C_1-C_3-C_2$ $i_5: C_5-C_4-C_2-C_3$										
		$N_3=3$		$M_3=1$													
		$N_4=3$		$M_4=2$													
		$N_5=4$		$M_5=1$													
		$N_6=3$					$i_6: C_4-C_5-C_2$										
	6	$N_1=4$	5	$M_1=1$	4	$T_{31}^{\gamma}=2$	$i_1: C_1-C_2-C_3-C_4$	5.37	3.36	3.32	21	21	21				
		$N_2=3$		$M_2=2$		$T_{51}^{\gamma}=3$	$i_2: C_2-C_4-C_3$ $i_3: C_4-C_3-C_2$ $i_4: C_1-C_3-C_2$ $i_5: C_5-C_4-C_2-C_3$										
		$N_3=3$		$M_3=1$		$T_{11}^{\gamma}=1$											
		$N_4=3$		$M_4=2$													
		$N_5=4$		$M_5=1$			$i_6: C_4-C_5-C_2$										
	6	$N_1=4$	5	$M_1=1$	4	$T_{31}^{\gamma}=2$	$i_1: C_1-C_2-C_3-C_4$	5.39	3.38	3.33	24	25	24				
		$N_2=3$		$M_2=2$		$T_{51}^{\gamma}=3$	$i_2: C_2-C_4-C_3$										
		$N_3=4$		$M_3=1$		$T_{11}^{\gamma}=1$	$i_3: C_4-C_3-C_2-C_5$										
						$T_{42}^{\gamma}=2$	$i_4: C_1-C_3-C_2$										

		$N_4 = 3$		$M_4 = 2$			$i_5 : C_5 - C_4 - C_2 - C_3$						
		$N_5 = 4$		$M_5 = 1$			$i_6 : C_4 - C_5 - C_2$						
		$N_6 = 3$											

Table 15. Obtained results for medium size and large scale problem through Exact Method, GA and GA-PSO.

Scale of problem	Specifications of problem							CPU Time (hour)			Tardiness Value		
	J	$N_i$	C	$M_c$	$\xi D^r$	$T_{cm}^r$	Jobs Sequence	Exact Method	GA	GA-PSO	Exact Method	GA	GA-PSO
Medium size	8	$N_1 = 5$	5	$M_1 = 1$ $M_2 = 2$ $M_3 = 1$ $M_4 = 2$ $M_5 = 3$	4	$T_{31}^r = 2$ $T_{51}^r = 3$	$i_1 : C_1 - C_2 - C_3 - C_4 - C_5$ $i_2 : C_2 - C_4 - C_3 - C_5$ $i_3 : C_4 - C_3 - C_2$ $i_4 : C_1 - C_3 - C_2$ $i_5 : C_5 - C_4 - C_2 - C_3$ $i_6 : C_3 - C_5 - C_2 - C_4 - C_1$ $i_7 : C_5 - C_1 - C_4$ $i_8 : C_1 - C_2 - C_4 - C_5$	-	4.98	4.86	-	32	30
		$N_2 = 4$											
		$N_3 = 3$											
		$N_4 = 3$											
		$N_5 = 4$											
		$N_6 = 5$											
		$N_7 = 3$											
		$N_8 = 4$											
	9	$N_1 = 5$	5	$M_1 = 1$ $M_2 = 2$ $M_3 = 1$ $M_4 = 2$ $M_5 = 3$	4	$T_{31}^r = 2$ $T_{51}^r = 3$	$i_1 : C_1 - C_2 - C_3 - C_4 - C_5$ $i_2 : C_2 - C_4 - C_3 - C_5$ $i_3 : C_4 - C_3 - C_2$ $i_4 : C_1 - C_3 - C_2$ $i_5 : C_5 - C_4 - C_2 - C_3$ $i_6 : C_3 - C_5 - C_2 - C_4 - C_1$ $i_7 : C_5 - C_1 - C_4$ $i_8 : C_1 - C_2 - C_4 - C_5$	-	5.21	4.99	-	35	32
		$N_2 = 4$											
		$N_3 = 3$											
		$N_4 = 3$											
		$N_5 = 4$											

[illegible]

		$N_7 = 3$					$i_{11} : C_5 - C_2 - C_4 - C_3 - C_6$ $i_{12} : C_6 - C_3 - C_4 - C_5 - C_1 - C_2$						
		$N_8 = 4$											
		$N_9 = 4$											
		$N_{10} = 6$											
		$N_{11} = 5$											
		$N_{12} = 6$											
		$N_1 = 5$											
		$N_2 = 4$											
		$N_3 = 3$					$i_1 : C_1 - C_2 - C_3 - C_4 - C_5$						
		$N_4 = 3$	$M_1 = 1$				$i_2 : C_2 - C_4 - C_3 - C_5$ $i_3 : C_4 - C_3 - C_2$						
		$N_5 = 4$	$M_2 = 3$				$i_4 : C_1 - C_3 - C_2$ $i_5 : C_5 - C_4 - C_2 - C_3$						
		$N_6 = 5$	$M_3 = 1$			$T_{31}^\gamma = 2$	$i_6 : C_3 - C_5 - C_2 - C_4 - C_1$						
13		$N_7 = 3$	6		4	$T_{51}^\gamma = 3$	$i_7 : C_5 - C_1 - C_4$ $i_8 : C_1 - C_2 - C_4 - C_5$ $i_9 : C_6 - C_3 - C_1 - C_5$	-	6.13	5.82	-	40	36
		$N_8 = 4$	$M_4 = 2$			$T_{61}^\gamma = 3$	$i_{10} : C_1 - C_2 - C_4 - C_5 - C_6 - C_3$ $i_{11} : C_5 - C_2 - C_4 - C_3 - C_6$						
		$N_9 = 4$	$M_5 = 3$				$i_{12} : C_6 - C_3 - C_4 - C_5 - C_1 - C_2$ $i_{13} : C_6 - C_5 - C_1 - C_3$						
		$N_{10} = 6$	$M_6 = 1$										
		$N_{11} = 5$											
		$N_{12} = 6$											
		$N_{13} = 4$											

Measurment	Hypothesis	T-Value	P_Value
Tardiness value	H1: $\mu_{GA-PSO} < \mu_{GA}$	2.57	0.005
CPU time	H1: $\mu_{GA-PSO} < \mu_{GA}$	2.82	0.002

Table 16. Results of statistical compare performances.

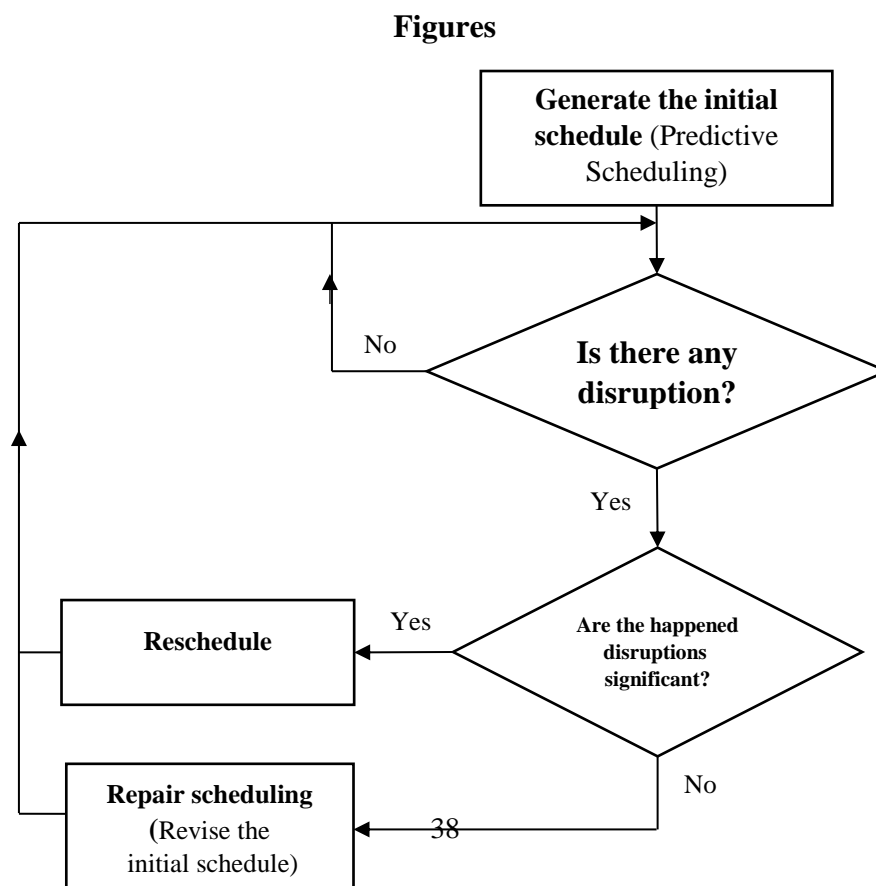


Fig. 1. Reactive scheduling process.

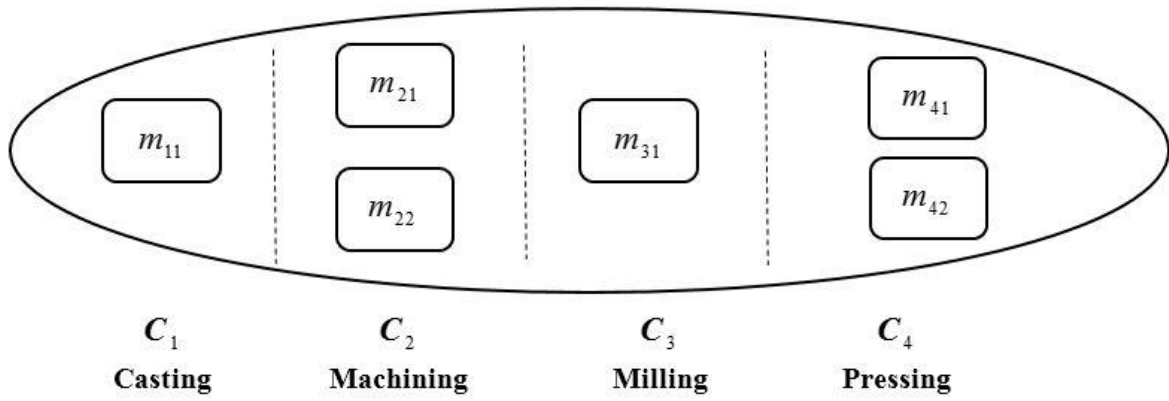


Fig. 2. Machines' placement over factory.

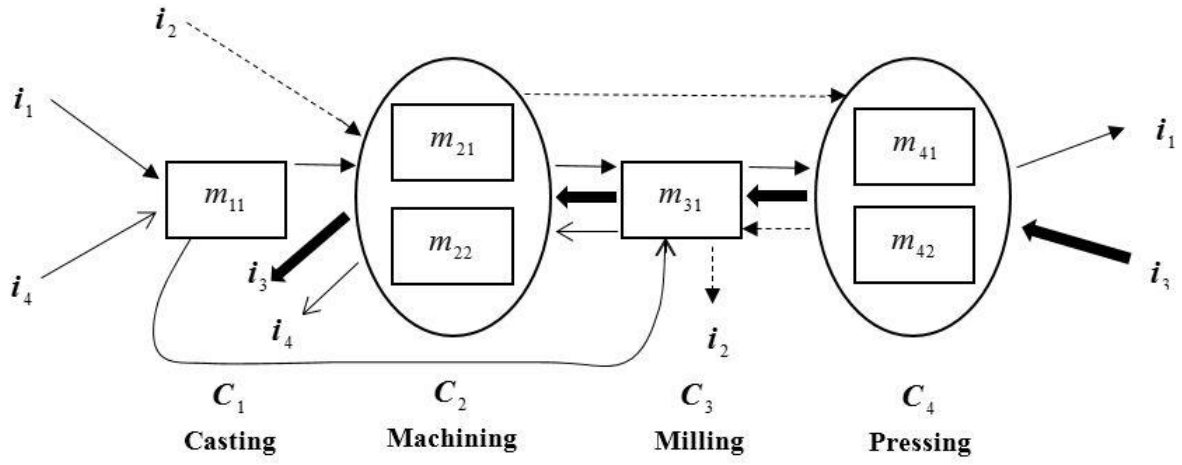


Fig. 3. Repair and production process of  $\tau^{\text{th}}$  period.

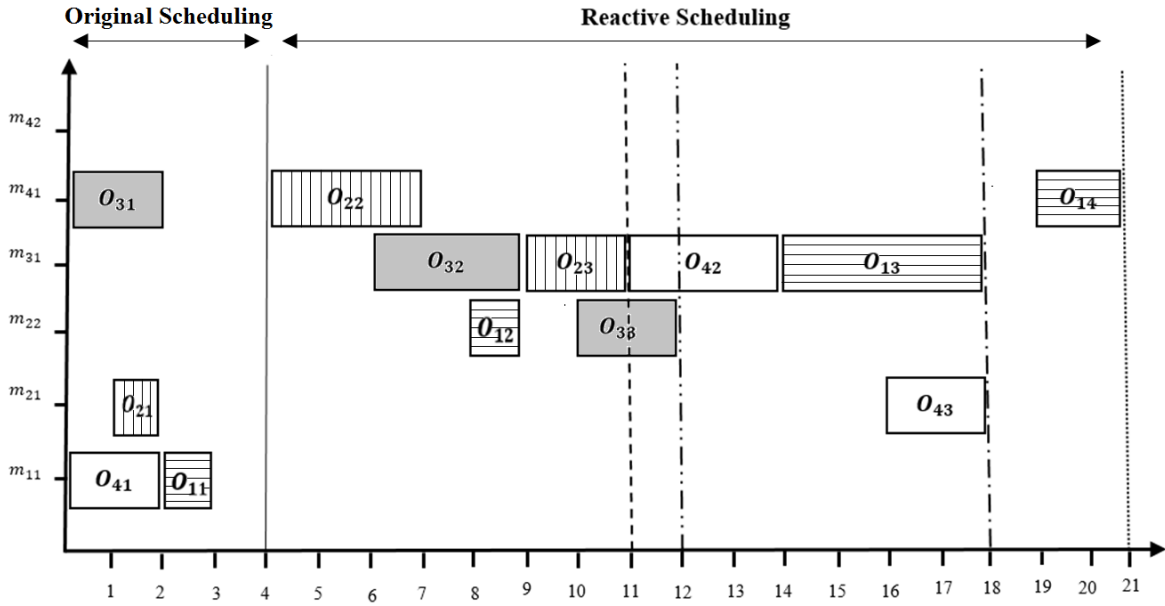


Fig. 4. Reactive scheduling in contrast with disruptions during production based on earliest time scheduling.



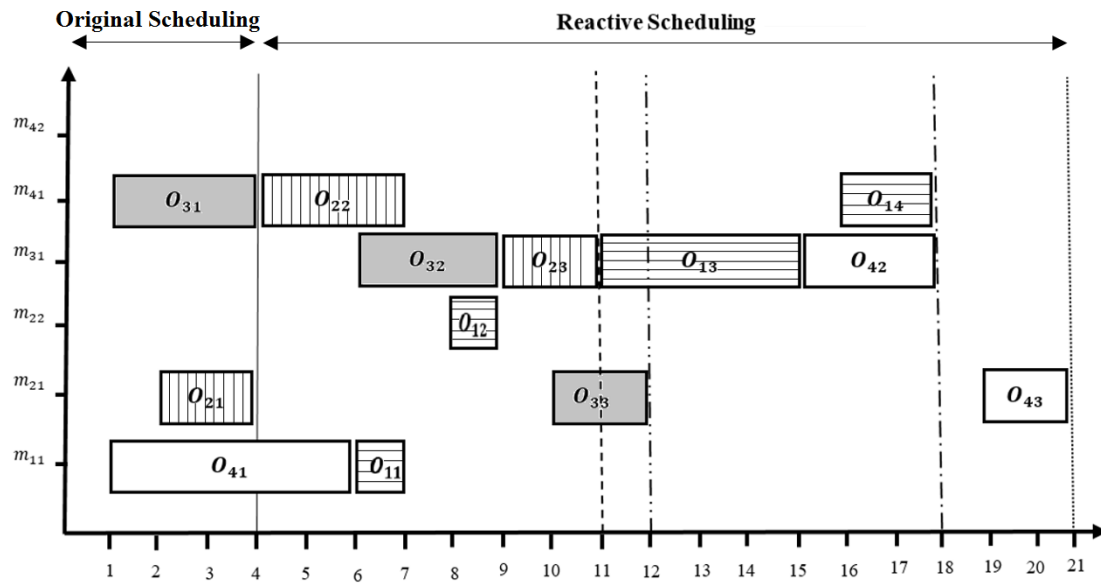


Fig. 5. Reactive scheduling in contrast with disruptions during production based on latest time scheduling.

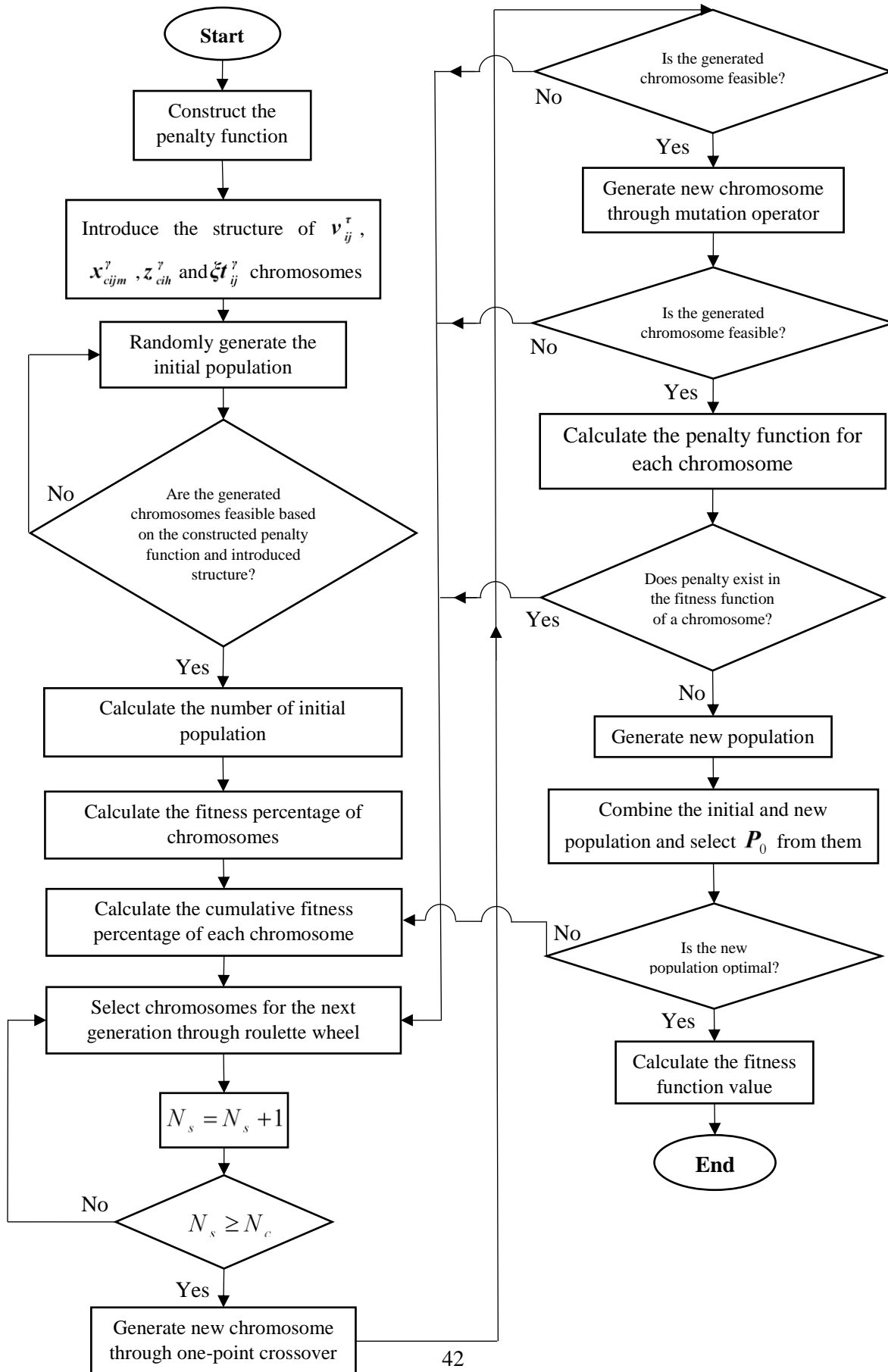
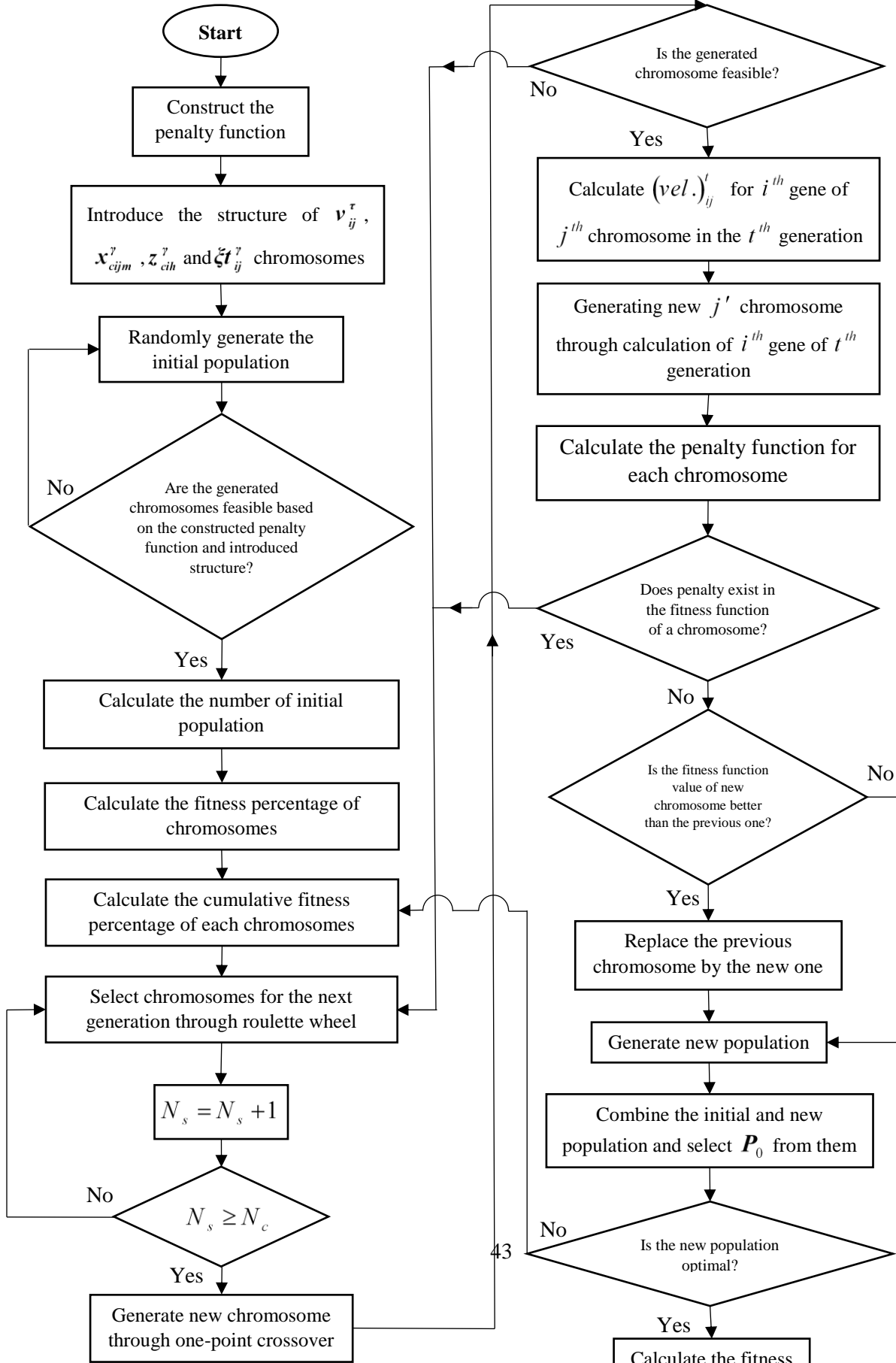


Fig. 6. The proposed GA



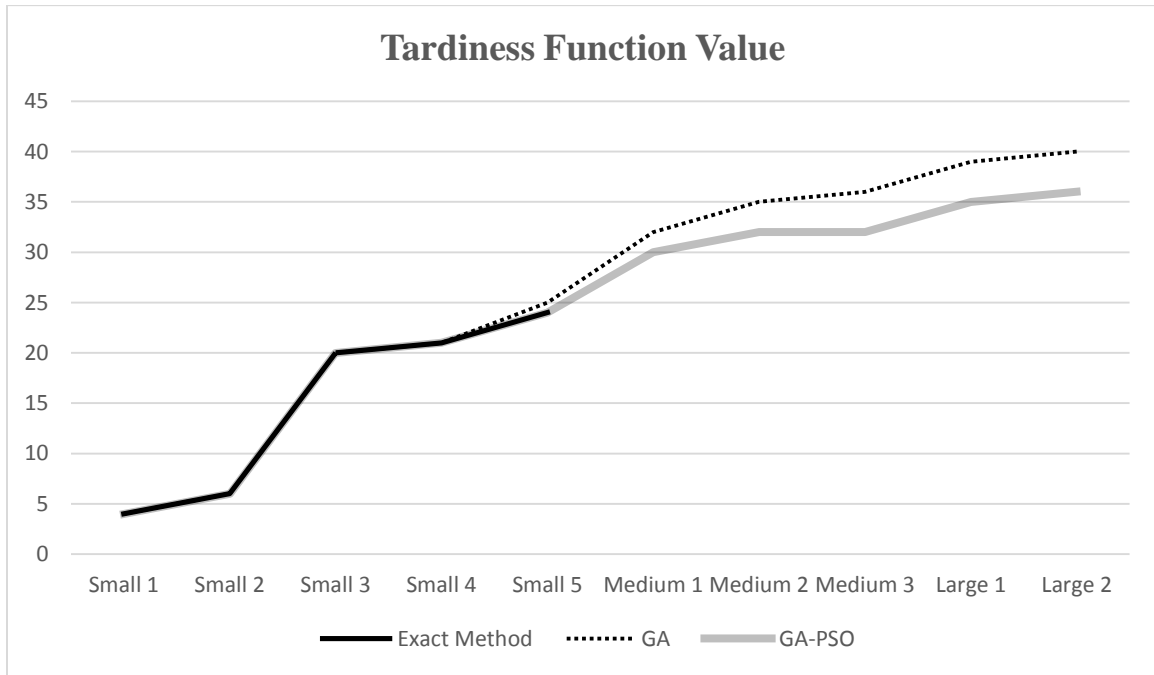


Fig. 8. Comparison of the obtained tardiness function value through Exact Method, GA and GA-PSO.

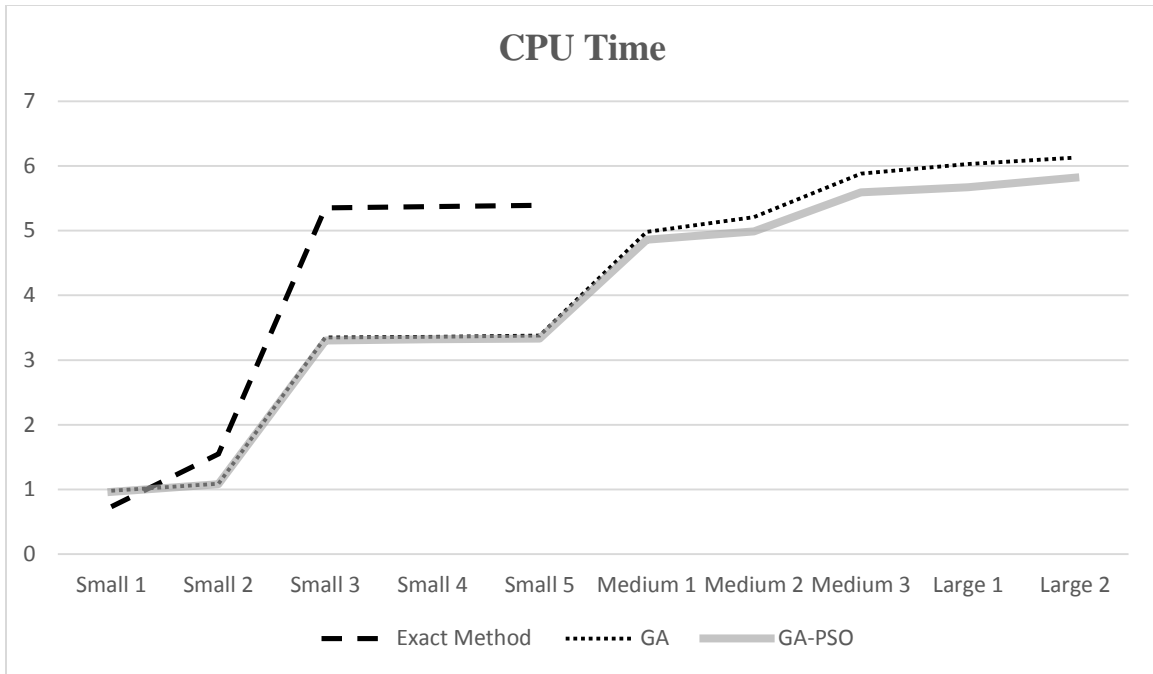


Fig. 9. Comparison of Exact Method, GA and GA-PSO in terms of CPU Time.