

# Bi-objective Optimization Model for Integrated Preventive Maintenance and Flexible Job-shop Scheduling Problem

Javad Rezaeian<sup>1,\*</sup>, Farzaneh Mohammadpour<sup>2</sup>

<sup>1</sup>Department of Industrial Engineering, Mazandaran University of Science and Technology, Iran

<sup>2</sup>Industrial Management Institute, Iran

Copyright©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

**Abstract** This paper develops an integrated model for flexible job-shop scheduling problem with the maintenance activities. Reliability models are used to perform the maintenance activities. This model involves two objectives: minimization of the maximum completion time for flexible job-shop production part and minimization of system unavailability for the PM (preventive maintenance) part. To aim the objectives, two decisions must be taken at the same time: assigning  $n$  jobs on  $m$  machines in order to minimize the maximum completion time and finding the appropriate times to perform PM activities to minimize the system unavailability. These objectives are obtained with considering dependent machine setup times for operations and release times for jobs. In advance, the maintenance activity numbers and PM intervals are not fixed. Two multi objective optimization methods are compared to find the pareto-optimal front in the flexible job-shop problem case. Promising the obtained results, a benchmark with a large number of test instances is employed.

**Keywords** Job-shop Scheduling, Preventive Maintenance, Bi-objective

## 1. Introduction

Production scheduling and maintenance activities are two important problems in production planning. The flexible job-shop scheduling problem (FJSP) is an expansion of the job-shop scheduling problem. Machine assignment and operation sequence are two sub-problems of the FJSP problem, defined to obtain the minimum of maximum completion time. So, the FJSP has a complex nature and is also strongly NP-hard. Berrichi and et al. [1] expand an efficient maintenance planning to optimize the

objectives of the systems. In the most of the real manufacturing systems, production scheduling and maintenance activities are employed separately but there is strict interdependency between them [1]. In recent decades, the conflict relationship between production and maintenance has been observed in management decision [2]. Although, if the PM (preventive maintenance) activities do not be inserted in production planning, this will be a cause of violation in production system when PM activities or machine break-down are occurred. By considering this fact, in this paper an integrated bi-objective model allows us to have two main criteria related to PM activities and production plan.

In recent years, production scheduling and PM activities are studied rapidly. The objectives of production scheduling problems consisted in: minimization of the completion time of jobs, mean of flow time of processing. The single machine and permutation flow-shop problems with PM activities that maintenance periods fulfilled during the predefined intervals have been studied by [1]. Sioud and Cagne preserved a minimum level of reliability in permutation flow-shop scheduling models by defining the maintenance periods and proposed an integrated method in order to minimize the maximum completion time [6]. Reliability model to define the maintenance intervals used by [1] that their strategy is named rational strategy. They introduced a model for integrated PM and parallel machine scheduling. To our knowledge, the first study related to the integrated method in the flexible job-shop scheduling and PM activities with the machine unavailability is studied by Moradi et al. [2]. They combined some logical strategies with PM activities introduced by [1].

In this paper we proposed an integrated model for flexible job-shop scheduling problem with PM activities based on the proposed model by [2] with considering the sequence-dependent setup times for machines and release

dates for each operation of jobs and solved it in order to minimize the maximum completion time and machine unavailability.

To clarify the chosen strategy and policy we solved an example presented by [2]. In this case, we have assumed that there is a flexible job-shop scheduling system and PM activities are done every 20 time unites (TFPM=20), the duration of each activity is 2 times (DFPM=2). There are 2 machines and 2 jobs that each job gets a number of operations. Table 1 shows the processing times, release dates and setup times of operations on two machines. Fig 1 shows the Gantt-chart of FJSP without considering the PM policy.

The chosen PM policy and strategy increases the availability of system with providing the good operation conditions and decreasing the failures. The availability is

defined as: “the probability that a system or a component is performing its required function at a given point of time or over a stated period of time when operated and maintained in a prescribed manner”. So, the availability of machine M at a given point at time t is defined as [1]:

$$A(t)=P (M \text{ is operating at time } t) \tag{1}$$

Also, the unavailability which is opposite to the availability can be stated as [1]:

$$\bar{A}(t)= 1-A(t). \tag{2}$$

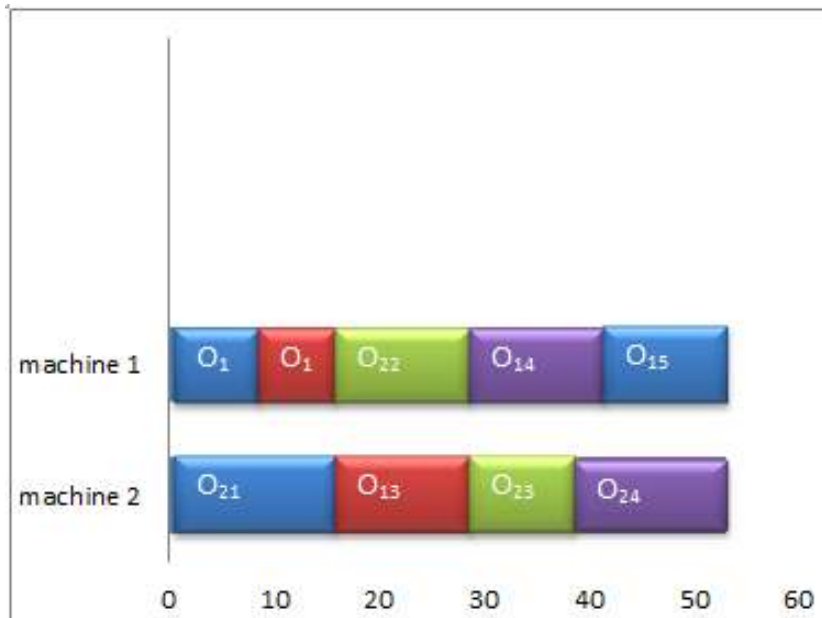
Fig.2 shows that the PM activities are done only in especial times 9, 16, 31, 41, 57. So, the unavailability is occurred at this time. Also, we supposed that the repair rate and the failure rate for machines are constant:  $\mu_i = 0.5, \lambda_i = 0.1$ . Table 2 shows the unavailability of machines and system.

**Table 1.** The processing times, release dates and setup times of operations

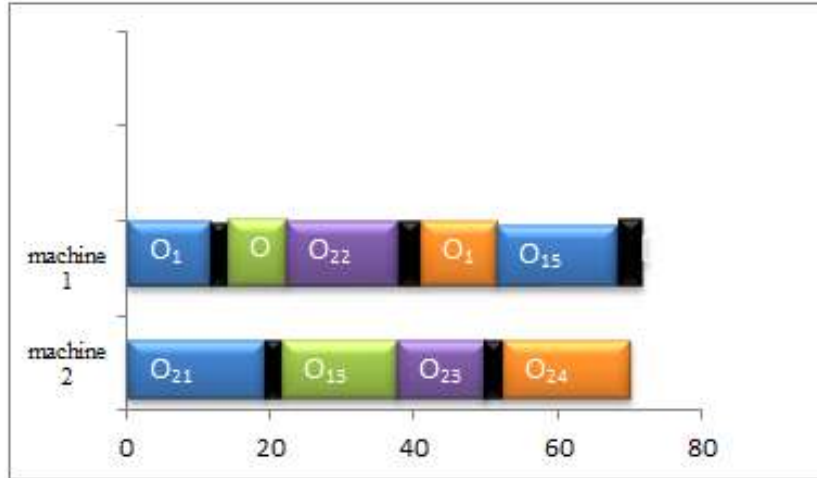
Jobs	Job 1					Job2			
Operations	o11	o12	o13	o14	o15	o21	o22	o23	o24
Processing times of machine 1	4	6	12	11	11	N	12	N	17
Processing times of machine 2	6	N	8	N	13	10	10	8	12
Release date of machine 1	3	0	0	0	0	4	0	0	0
Release date of machine 2	4	0	0	0	0	3	0	0	0
Setup times of machine 1	2	1	6	1	1	5	1	3	4
Setup times of machine 2	3	2	5	3	4	3	2	2	2

$O_{ij}$ : the operation  $kth$  of job  $ith$

N: machine cannot be able to process operation.



**Figure 1.** Gantt chart of FJSP without maintenance activities



**Figure 2.** Gantt chart of the integrated production and maintenance activities ( Machine unavailability for preventive maintenance activities)

**Table 2.** Best, average and worst number of obtained solutions for MOICA

The number of problem	MOICA		
	Best	Average	Worst
1	29	10.15	3
2	24	8.13	1
3	26	5.93	2
4	18	4.73	1
5	23	6.22	1
6	22	7.11	1
7	19	5.09	1
8	13	4.71	1
9	13	3.26	1

The remaining of study is organized as follows: The Multi-objective imperialist competitive algorithm is introduced in Section 2. Computational results are explained in Section 3. Finally, Section 4 contains the conclusions and future studies.

## 2. Multi-objective Imperialist Competitive Algorithm (MOICA)

For solving the proposed problem, ICA (imperialist competitive algorithm) for multi objective problems that is called MOICA (multi objective competitive algorithm) is applied.

### 2.1. Generating Initial Empires

Each solution in the imperialist competitive algorithm (ICA) is in a form of an array. Each array consists of variable values to be optimized. In the GA terminology, this array is called chromosome; however, in this paper, we use the term “country” for this array. In an  $N$ -dimensional optimization problem, a country is a  $1 \times N$  array. This array

is defined by:  $country = [p_1, p_2, p_3, \dots, p_N]$ , where  $p_i$  is the variable to be optimized. Each variable in a country denotes a socio-political characteristic of a country. From this point of view, the algorithm searches for the best country that is the country with the best combination of socio-political characteristics, such as culture, language and economic policy [3]. After generation of countries, a non-dominance technique and a crowding distance are applied to shape the fronts and rank member of each front, respectively. Then, the members of front one are saved in archive.

1) Non-dominance technique: Suppose that there are  $r$  objective functions. When the following conditions are satisfied, solution  $x_1$  dominates another solution  $x_2$ . If  $x_1$  and  $x_2$  do not dominate each other, they are placed in the same front [5].

- For all the objective functions, solution  $x_1$  is not worse than another solution  $x_2$ .
- For at least one of the  $r$  objective functions  $x_1$  is exactly better than  $x_2$ .

Front number 1 is made by all solutions that are not dominated by any other solutions. Also, front number 2 is built by all solutions that only dominated by solutions in front number 1. In this order, other fronts are made.

2) Crowding distance: The crowding distance, which is a measure of the density of solutions present an estimate of the density of solutions surrounding a particular solution [5].

The crowding distance measure used in the MOICA is shown in Equation (3). The solutions having a lower value of the crowding distance are preferred over solutions with a higher value of the crowding distance.

$$CD_i = \sum_{k=1}^r \frac{f_{k,i+1}^p - f_{k,i-1}^p}{f_{k,total}^{p,max} - f_{k,total}^{p,min}} \quad (3)$$

where  $r$  is the number of objective functions,  $f_{k,i+1}^p$  is the

$k$ th objective function of the  $(i+1)$ th solution and  $f_{k,i-1}^p$  is the  $k$ -th objective function of the  $(i-1)$ th solution after sorting the population according to crowding distance of the  $k$ -th objective function. Also,  $f_{k,total}^{p,max}$  and  $f_{k,total}^{p,min}$  are the maximum and minimum value of objective function  $k$ , respectively. Then, the best solutions in terms of the non-dominance and crowding distance are selected from population as the imperialists and the remaining countries are colonies. To calculate the cost value of each imperialist, the value of each objective function is obtained for each imperialist. Then, the cost value of each objective function is calculated by:

$$cost_{i,n} = \frac{|f_{i,n}^p - f_i^{p,best}|}{f_{i,total}^{p,max} - f_{i,total}^{p,min}} \quad (4)$$

Where  $cost_{i,n}$  the normalized value of objective is function  $i$  for imperialist  $n$ , and  $f_{i,n}^p$  is the value of the objective function  $i$  for imperialist  $n$ . Also,  $f_i^{p,best}$ ,  $f_{i,total}^{p,max}$  and  $f_{i,total}^{p,min}$  are the best, maximum and minimum values of objective function  $i$  in each iteration, respectively. Finally, the normalized cost value of each imperialist ( $Total Cost_n$ ) is obtained by:

$$Total Cost_n = \sum_{i=1}^r cost_{i,n} \quad (5)$$

Where  $r$  is the number of objective function. The power of each imperialist is calculated by equation 6 and the colonies distributed among the imperialist according to power of each imperialist country.

$$p_n = \left| \frac{Total Cost_n}{\sum_{i=1}^{N_{imp}} Total Cost_n} \right| \quad (6)$$

Then, the initial number of colonies of an empire will be as follows:

$$NC_n = round \{P_n \cdot N_{col}\} \quad (7)$$

Where  $NC_n$  is the initial number of colonies of the  $n$ th imperialist, and  $N_{col}$  is the number of all colonies.  $NC_n$  of colonies are selected randomly and are assigned to each imperialist. Imperialist with higher power has a greater number of colonies while imperialist with weaker power has less.

### 2.2. Total Power of an Empire

The total power of an empire is mainly affected by the power of the imperialist country; however, the power of the colonies of an empire has an indigent effect on the total power of that empire. Therefore, the equation of the total power of an empire shown below.

$$TP Emp_n = (Total Cost(imperialist_n) + \xi mean\{Total Cost(colonies of empire_n)\})(1 - QE_n) \quad (8)$$

Where  $TP Emp_n$  is the total power of the  $n$ th empire and zeta ( $\xi$ ) is a positive number which is considered to be less than 1. The total cost of imperialists and colonies are calculated by equations 58 and 59. In addition,  $QE_n$  is the quality of the  $n$ th empire determined as following. First, all of the imperialists and colonies are put together and then the non-dominated solutions are selected. The percentage of the non-dominated solution belonging to each empire is calculated as  $QE_n$ . This is necessary to mention that the total power of the empire to be determined by just the imperialist when the value of  $\xi$  is small and increase of it will increase the role of the colonies in determining the total power of an empire.

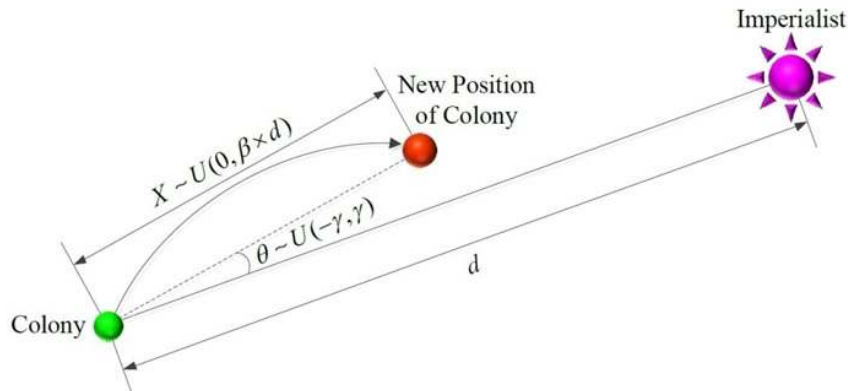


Figure 3. Moving colonies toward the imperialist with a random angle  $\theta$

**2.3. Moving the Colonies of an Empire toward the Imperialist (Assimilating)**

After dividing colonies between imperialists, colonies are moved toward their related imperialist. This movement is shown in Figure 3, in which  $d$  is the distance between colony and imperialist.  $X$  is a random variable with a uniform (or any proper) distribution between 0 and  $\beta \times d$  and  $\beta$  is a random number greater than one. Direction of the movement is shown by  $\theta$ , which is a random number between  $-\mathcal{V}$  and  $\mathcal{V}$ .

**2.4. Information Sharing between Colonies**

Moreover, colonies are sharing their information together by crossover to do themselves better. In this section, for this target one of these operators including one-point, two-point and continuous uniform crossover are selected randomly. The population percentage that is sharing information is shown by Pc. The best colonies have the more chance than others to share its information because colonies are selected in this section by the tournament selection, which is described below. A binary tournament selection procedure has been applied for selecting solutions for both the crossover and mutation operators. This procedure works as follow. First select two

solutions of the population size, and then the lowest front number is selected if the two populations are from different fronts. If they become from the same front, the solution with the highest crowding distance is selected.

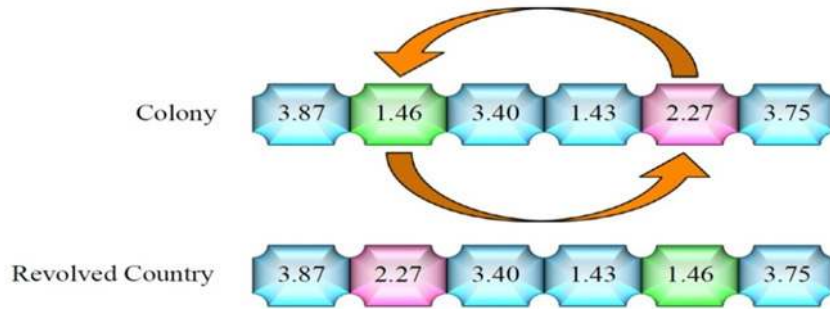
**2.5. Revolution**

In each decade, revolutions are performed on some of the colonies. For this target, swap operator is selected. The structures of these operators are described as follows. The revolution rate in this study is shown by  $P_r$ .

- Swap: for swapping, two numbers of a colony are first selected randomly (numbers 1.46 and 2.27 in Figure 4), then the positions of the selected numbers are exchanged, which is shown in Figure 4.

**2.6. Improve Imperialist**

In this step, the imperialist with minimum power will generate the maximum neighborhood ( $N_{max}$ ). Likewise, the imperialist with the maximum power will generate the minimum neighborhood ( $N_{min}$ ). The number of neighborhoods associated with each imperialist is generated depending on the value of the power and is produced using a linear function varying in a range between  $N_{min}$  and  $N_{max}$ , which is shown in Eq. 9.



**Figure 4.** Swap operator

$$Ne_n = floor(Ne_{min} + \frac{(Ne_{max} - Ne_{min})(TP Emp_{worst} - TP Emp_n)}{(TP Emp_{worst} - TP Emp_{best})}) \tag{9}$$

Where  $TP Emp_{best}$  is the value of total power of the strength empire and  $TP Emp_{worst}$  is the value of total power of the weakest power empire.

2.6.1. Colonies Updated

In each decade, the initial population of colonies, population of assimilating, population of Information sharing between colonies, population of revolution and population of improved imperialist are merged together for each empire as merged population. Afterward, archived updating is done on merged population. Then, the best of colonies based on non-dominating sorting and crowding distance by the population size of colonies for given empires (NC(i)) are selected for each imperialist.

2.6.2. Archive Adaption

Ranking and sorting is done by the non-dominated and crowding distance for the merged population. Then, the solutions of first front are selected in order to add to the archive. Finally, the solutions of first front are kept and others are deleted after ranking and sorting solutions in the archive. Also, the size of archive is equal to  $nArchive$ .

2.6.3. Exchanging Positions of the Imperialist and a Colony

First, the total cost of each imperialist is updated. Then, each imperialist with the best colony are merged together. Afterward, this population is sorted based on the non-dominated sorting and crowding distance. Finally, the best of them is selected as imperialist.

2.6.4. Imperialistic Competition

The power of a weaker empire will reduce, and the power of more powerful ones will rise in the imperialistic competition. All empires compete together to take the position of the weakest colony of the weakest empire. In other words, first choosing some (usually one) of the weakest colonies of the weakest empire and then the position of these colonies (or this colony) are given to the winner imperialist among all empires in the imperialistic competition. In this competition, the most powerful empires will not definitely possess these colonies; however, these empires will be more possible to possess them. This competition is modeled by just selecting one of the weakest colonies of the weakest empires and then calculating the possession probability of empires has the largest normalized total cost as follows.

$$NTP Emp_n = \max \{ TP Emp_i \} - TP Emp_n \quad (10)$$

Where  $NTP_n$  is the normalized total power of nth empire, and  $TP_n$  is the total power of nth empire. Having the normalized total power, the possession probability of each empire is calculated by:

$$PP_n = \left| \frac{NTP Emp_n}{\sum_{i=1}^{N_{imp}} NTP Emp_i} \right| \quad (11)$$

Then, the roulette wheel method is used for assigning the mentioned colonies to selected empire.

2.6.5. Eliminating the Powerless Empires

Powerless empires will collapse and their colonies will be distributed among other empires in the imperialistic competition. In this paper, when an empire loses all of its colonies, we consider it collapses.

2.7. Stopping Criteria

In this paper, the stopping criteria or end of imperialistic competition is considered when there is only one empire between all of empires. The convergence of the algorithm in three segments of iterations is illustrated in Figure 5.

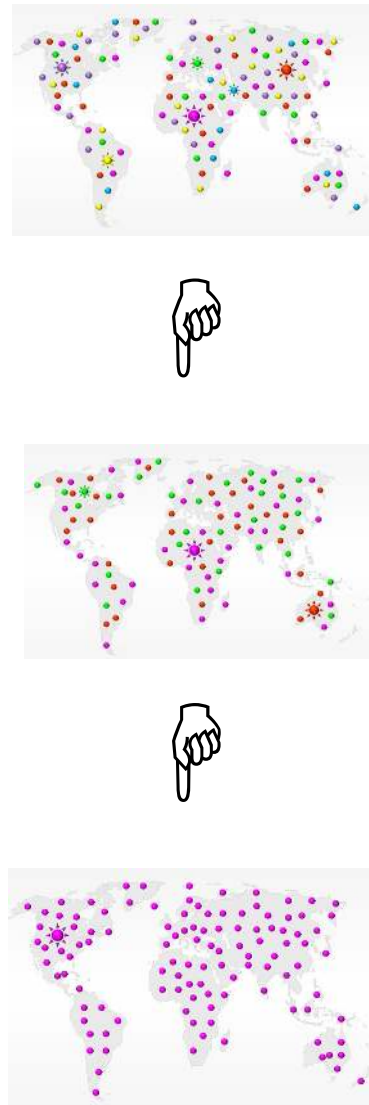


Figure 5. Convergence of the algorithm

### 3. Computational Results

In this section, some standard flexible job-shop test problems from [3] and [4] are considered. Several standard m-machines, n-jobs flexible job-shop test problems are presented. Using the number of jobs (n) and number of machines (m) noted by couple (n, m), they introduced test problems with minimum (n,m) of (6, 6) and maximum (n,m) of (30, 10). And for MOICA algorithm two comparison metrics are taken into account. It is noticeable that before calculation of each performance metric, both of objectives are normalized.

Diversification metric (DM): This metric measure the spread of the solution set and calculated by:

$$DM = \sqrt{(\max f_{1i} - \min f_{1i})^2 + (\max f_{2i} - \min f_{2i})^2} \tag{12}$$

Mean ideal distance (MID): The closeness between Pareto solutions and ideal point  $(f_1^{best}, f_2^{best})$  is determined by using the MID. The equation of the MID is defined by:

$$MID = \frac{\sum_{i=1}^n \sqrt{(f_{1i} - f_1^{best})^2 + (f_{2i} - f_2^{best})^2}}{n} \tag{13}$$

Where n is the number of non-dominated solutions and  $f_{i,total}^{max}$  and  $f_{i,total}^{min}$  are the maximum and minimum values of each fitness functions among the all non-dominated solutions obtained by the algorithms, respectively. Table 2 represents the best, average and worst number of obtained solutions. Table 3 gives the best, the average and the worst value of the C metric obtained over the solution sets. Table 4 illuminates the H metric obtained over solution sets.

**Table 3.** Best, average and worst values of the C metric obtained by MOICA

The number of problems	MOICA		
	Best	Average	Worst
1	1	0.29	0
2	0.89	0.32	0
3	1	0.24	0
4	1	0.36	0
5	1	0.38	0
6	1	0.30	0
7	0.76	0.26	0
8	1	0.44	0
9	1	0.31	0

**Table 4.** Best, average and worst values of the H metric obtained by MOICA

The number of problems	MOICA		
	Best	Average	Worst
1	34.72	291.15	607.83
2	29.46	167.52	503.57
3	33.09	159.30	347.13
4	67.62	311.44	829.76
5	29.39	126.16	385.03
6	48.74	347.94	698.94
7	61.22	383.01	1267.27
8	52.86	265.55	1044.35
9	57.77	391.37	1131.51

### 4. Conclusions and Future Studies

This paper is developed an integrated model to solve the PM and flexible job-shop scheduling problem, and optimized two manufacturing systems objectives. Two multi-objective evolutionary algorithms were adapted to our problem. Computational results with nine standard test problems have shown that the MOICA algorithm provided efficient solutions, based on two performance measures. As an offer for future studies, it would be considerable to employ other maintenance policies and strategies for this problem.

### REFERENCES

- [1] Berrichi, A., Amodeo, L., Yalaoui, F., Châtelet, E., & Mezghiche, M., 2008. Bi-objective optimization algorithms for joint production and maintenance scheduling: Application to the parallel machine problem. *Journal of Intelligent.*
- [2] E. Moradi, S.M.T. Fatemi Ghomi, M. Zandieh., 2011. Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem. *Expert Systems with Applications*, 38, 7169-7178.
- [3] E. Atashpaz-Gargari and C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competitive. *IEEE Congress on Evolutionary computation*, Singapore, 2007.
- [4] Ishibuchi, H., Yoshida, T., & Murata, T., 2003. Balance between genetic search and local search in memetic algorithms for multi-objective permutation flow-shop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2), 204-223.

- [5] Kaabi, J., Varnier, C., & Zerhouni, N., 2003. Genetic algorithm for scheduling production and maintenance in a flow-shop. France: Laboratory of Automatic of Besancon. In French.
- [6] Sioud A.G.M. and Cagne, C., 2012. A hybrid genetic algorithm for a single machine scheduling problem with sequence dependent setup times. *Computers and Operations Research*, 39, 2415-2424.