

# **Pareto- Front Generation by Classical and Meta-heuristic Methods in Flexible Job Shop Scheduling with Multiple Objectives**

Maryam Ghasemi

Department of Computer Engineering,  
Shabestar Branch, Islamic Azad University  
Shabestar, Iran

Ali Farzan

Department of Computer Engineering,  
Shabestar Branch, Islamic Azad University  
Shabestar, Iran

## **ABSTRACT**

Planning and scheduling are as decision making processes which they have important roles in production systems and industries. According that, job shop scheduling is one of NP-hard problems to solve multi-objective decision making approaches. So, the problem is known as uncertain with many variables in optimal solution view. Finding optimal solutions are essential task in scheduling of jobs between machines in the industries. In this paper, we present classical sum weighted (WS) method and non-dominated sorting genetic algorithm II (NSGA-II) to solve flexible job shop scheduling problem (FJSSP) with multiple objectives and find Pareto-fronts: minimizing completion time of jobs and maximizing machine employment. To generate Pareto-fronts, a search algorithm uses mechanism of variable weights and random selection to change directions in search spaces. The experiment results indicate that NSGA-II solve the problem more acceptable than WS method with considering computing time and consuming memory.

## **Keywords**

Multi-Objective Optimization, Flexible Job Shop Scheduling Problem, Weighted Sum Method, NSGA-II, Production Systems.

## **1. INTRODUCTION**

The classical job shop problem (JSP) allows a job to be processed by any machine and FJSSP is an extension of JSP. The FJSSP is an optimization problem with many of variables and it is known as NP-hard problem [1, 2]. In the FJSSP operations of jobs are allocated to the resources at particular times. This is a field of computer and industrial engineering and operations research. The classical and flexible methods can solve FJSSP. Some of the classical and flexible methods are based on WS method [3]. The WS method can convert multi objective optimization problem into single objective by using objective functions [4] in the multi objective decision theory. In decision theory, WS method is the simplest multi objective classical decision making method in order to evaluate number of decision making measures/variables. In FJSSP, each job involves a set of operations which have to be processed by a set of machines. The machines are processing the processes at the specified time, consequently [5].

FJSSP is a sub-branch of resources management in the production systems. Scheduling and planning are two tasks of resource management in order to using flexibility as response mechanism to find Pareto-fronts. In the decades before, the classic methods were used for scheduling problem, but they did not have any flexibility in the application process. While, in the recent decades, intelligent and meta-heuristic methods to solve job scheduling problem are used as flexibility [6].

Intelligent based FJSSP solving methods are bee colony algorithm [7, 8], hybrid genetic algorithms and Tabu search [9], particle swarm optimization [10], gradual freezing algorithm and Tabu search algorithm [11] which they uses optimization methods to resolve scheduling problems.

The proposed method is trying to resolve FJSSP using two parts of chromosome structure and the comparison between classical and optimization methods are presented for job scheduling. Classical methods use WS method to find optimal non-dominated solutions in order to minimize completion time of jobs (Makespan) within maximizing workloads of unemployment machines by two parts of chromosome in WS method (2p-WS). The meta-heuristic approach in this paper uses NSGA-II to overcome scheduling problem with two parts of chromosomes (2p-NSGA II) in order to minimize Makespan with maximizing machines employment. The paper is structured as follows: Section 2 is introduced previous works in multi-objective job scheduling methods. Section 3 is problem definition and proposed method using WS method and NSGA-II algorithm such as initial population, chromosome structure and etc. Section 4 is presented proposed method and in the Section 5, the results and discussion of proposed method has been considered. Section 6 is concluded the paper.

## **2. PREVIOUS WORKS**

Using evolution algorithms for JSSP is started around 1980. Flexibility in scheduling is as a response mechanism in today's production systems for FJSSP in which more than one machine perform each operation are considered. Because, intelligent applicable scheduling jobs are used to optimize resource utilization and minimizing completion time of jobs and machines employment. Evolution algorithms work directly with real valued vectors, typically and genetic algorithms use typically strings of bits within dominant and non- dominant operators [12]. According that, Murata et al. [13] presented genetic algorithm to resolve JSSP. They used one objective known as minimizing Makespan. They indicated that two point crossover and shift change mutation are effective for this problem. The comparison between the genetic algorithm with other search algorithms such as local search, Tabu search and simulated annealing demonstrated high performance of their proposed genetic algorithm.

A hybrid genetic algorithm [9] is a new model has been presented to resolve FJSSP that combine genetic algorithm and Tabu search algorithm. The proposed method is used local and general search of genetic algorithm and Tabu search algorithm to explorations. The experiment results indicated that the proposed presented model had high performance regardless of the accuracy of solutions and computing time. In the recent years, there appeared meta-heuristic approaches to

overcome hardness of FJSSP. Such as Asadzadeh [14] presented a parallel artificial bee colony (pABC) to solve job scheduling and planning problem. The proposed method consisted of several colonies that located on different hosts with parallel manner and dynamic migration strategy to determine when they must communicate. This method improves the solution efficiency within very high convergence speed.

Recently, Jamili [15] proposed four methods to solve variant defined robust job scheduling problem. The proposed method used mathematical model, branch and bound algorithm, beam search algorithm and particle swarm optimization (PSO) algorithm. In first and second methods, the computing time increase when the size of population increases. For large scale problem, these methods have high required time to find optimal solutions; where, third and fourth methods are flexible to find optimal solutions in the large-scale problems within acceptable computing time. In our proposed model, classical WS and NSGA-II methods are used to solve job scheduling problem in production systems of industries. The proposed model presents two parts of chromosomes. The first part contains sequence of task operations and second part contains sequence of allocated machines to them.

### 3. PROBLEM DEFINITION

There are two challenges in the problem definition. The first challenge is about allocating machines to task operations and secondly is order of task operations which are allocated on each machine. Two parts of chromosome structure in WS (2p-WS) and NSGA II (2p-NSGA II) are used to resolve first challenge. There is no particular sequence of operations in each machine to resolve the second challenge; randomly selected from available free machines to process task operations. Problem definition consists of these assumptions:

- n is number of jobs and all jobs are independent together and is displayed as  $J_i$ .
- m is number of machines which are available and is displayed as  $M_k$ .
- All devices and operations are available at  $t = 0$ .
- At a specified time, the machine can only do one job: other jobs are available when this job is completed; which is now designated (resource constraints).
- Each job consists of a series of operations that must be carried out.
- Each operation has different running time on different machine.
- Running time of each operation ( $O_{ij}$ ) on each machine k is displayed as  $P_{i,j,k}$ .

Bi-objective functions are: minimize completion time of jobs (Makespan) and minimize unemployment time of machines (maximum machine employment). These objective functions are defined as follows: Job J spent processing time  $P_{ij}$  on the machine  $M_k$ . So,  $\sum P_{ij}$  for all  $J \in j_i$ .

$$\text{Makespan} = \text{Min} \sum P_{ij}$$

$W_k$ : workload of machine K

$W_T$ : workload of all machines

$$W_T \geq W_k, W_k \geq 0$$

Minimize  $Z_1 = \text{Makespan}$

$$\text{Maximize } Z_2 = WT = \sum_{k=1}^m W_k$$

In scheduling systems, completion time of task operations is important role [16]. Industry managers are interested to outperform tasks quickly with improve efficiency of machines as parallel working. On the other hand, maximizing or minimizing machine employment exerts a direct influence on first objective values ( $Z_1$ ).

### 3.1 Representation of Solutions

Mutation coding is related to chromosome structure. Ho et al. [17] presented a total survey of chromosome structures in job scheduling problems. They divided chromosome into two parts which first part indicated order of operations and second part indicated machines which they are shown by one array with binary values. In the proposed mutation coding mode, first row consists of processing times of task operations as string of bits (Chromosome X) and second row consists of allocated free machines to each operations of first row (Chromosome Y) which it is an array with mutation coding in the job scheduling problems. In this method, each chromosome is appeared as string of numbers. Table 1 shows mutation coding sample.

Table 1. Mutation Coding Sample

Chromosome X	3	1	2	3	1	3	1	2	2
Chromosome Y	4	2	1	2	4	3	1	3	2

This structure has two parts of Chromosomes. Machines are selected randomly between free machines in free machines list. This approach is used to solve job shop scheduling and overcome first challenge. The WS method works on the parameters and variables of problem with coding problem while NSGA-II method works on coding model of parameters and variables of problem instead of working on parameters or variables.

The initial population is a feature of evolutionary algorithms that affects the rate of convergence and quality of the final solution [18]. Each population or a generation of chromosomes has a size which is known as population size. Population size represents the number of chromosomes in the population or in a generation. In NSGA-II, population (individual) composed of sections as shown in Table 2.

Table 2. Population Components

Name	Description
Position	Position of each member
Cost	Required time to completion of this job on each machine.
Rank	Save ranking.
Domination Set ( $S_p$ )	Represents a set of population that this particular member dominates over them.
Dominated Count ( $n_p$ )	Number of times that this member of the population is dominated by others.
Crowding Distance	Function <i>CalccrowdingDistance</i> calculates crowding distances between obtained samples of solutions and sorting them based on crowding distance by using related equation.

### 3.2 Selection Operator

In the proposed method, binary tournament selection operator is used. The binary selection operator selects chromosomes to move them in the next generation. This operator applies crossover and mutation operators in the next step and creates new generations with new solutions. In the tournament binary method, two solutions will be selected randomly among of population and then a comparison is done between those and finally better solution will be selected. In the first, a selection criterion of NSGA-II is ranking of solution and in the second is crowding distance of the solution. Whatever the ranking of solution is lower and crowding distance is higher, it is more favorable.

### 3.3 Crossover and Mutation Operators

We use crossover operator to create new chromosomes (parents). According to the features of selected chromosome, this operator transfers these features to the new chromosomes by applying crossover operation. Single-point crossover is used to generate new chromosome. In the single-point crossover, a point is generated randomly and each chromosome will be cut from the generated point and the corresponding parts are combined with each other as shown in Figure 1.

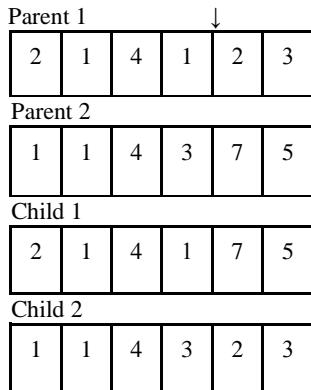


Fig 1: Single- Point Crossover Operator

The mutation operator is another operator of NSGA-II that increases diversity of population. This leads to creation of new population which does not exist in the total population, probably. Two times is used this operator: once for part of operation sequence and then for the part of allocated machines. In this algorithm, relocation mutation operator is used to the part of sequence operation in which two blocks of genes are selected and positions of them are relocated with together as shown in Figure 2.

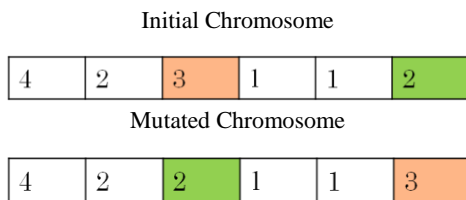


Fig 2: Mutation Operator

For mutation in the part of allocated machine, swap mutation operator is used in which one gene is selected randomly and value of it will be changed between a list of available machines. After crossover and mutation operations, new population selection is done among of initial population which are parents population and offspring population [P(t), Q(t)] in step 1. When generated populations are together, it is the time that sorting (Sort) will be used in the terms of non-dominated obtained solutions in step 2. After sorting and get non-dominated solutions, Pareto-fronts will obtain that include Pareto-front F<sub>1</sub>, Pareto-front F<sub>2</sub>, Pareto-front F<sub>3</sub> and so on to be continued in step 3. New population selection is done based on the priority of non-dominated solutions which obtained from Pareto-fronts in step 4 as shown in Figure 3.

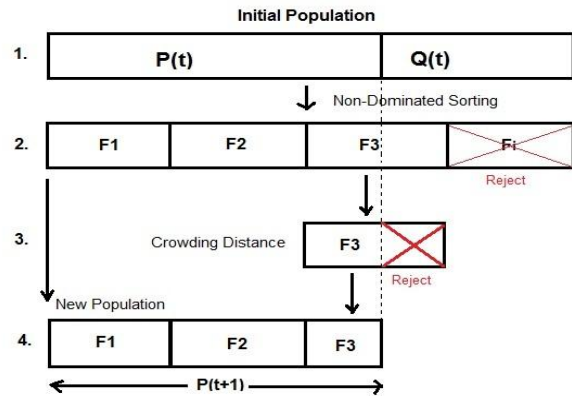


Fig 3: New Population Selection

As shown in Figure 3, F<sub>1</sub> and F<sub>2</sub> are selected as members of new population and the set of members of Pareto-fronts F<sub>3</sub> will selected which sorting of them is done based of calculating crowding distance. A set of members of Pareto-front with lower crowding distance will not selected as members of new population in the next stage and will be rejected (Reject Operation). Other obtained members from Pareto-front in section i is also rejected and because of existing better solutions in the previous stages with non-dominated status which are dominates these solution, then these are not members of in new population. Finally, new population P(t+1) will composed of members of Pareto-fronts F<sub>1</sub>, F<sub>2</sub> and some of members of F<sub>3</sub>.

Dominating is used to find non-dominated samples. Dominating concept or dominating that is used in Function dominates as shown in equation (1):

$$X \leq Y, X \text{ Dom } Y \quad (1)$$

It demonstrated that X dominates Y when value of x is equal or smaller than y and for all of i,  $x_i \leq y_i$  there is i<sub>0</sub> in which  $X_{i_0} < Y_{i_0}$  and x is better than y strictly at least from a review. This concept is used in Function *Dominate* to find non-dominated samples. Flowchart of non-dominated sorting algorithm in NSGA-II is shown in Figure 4.

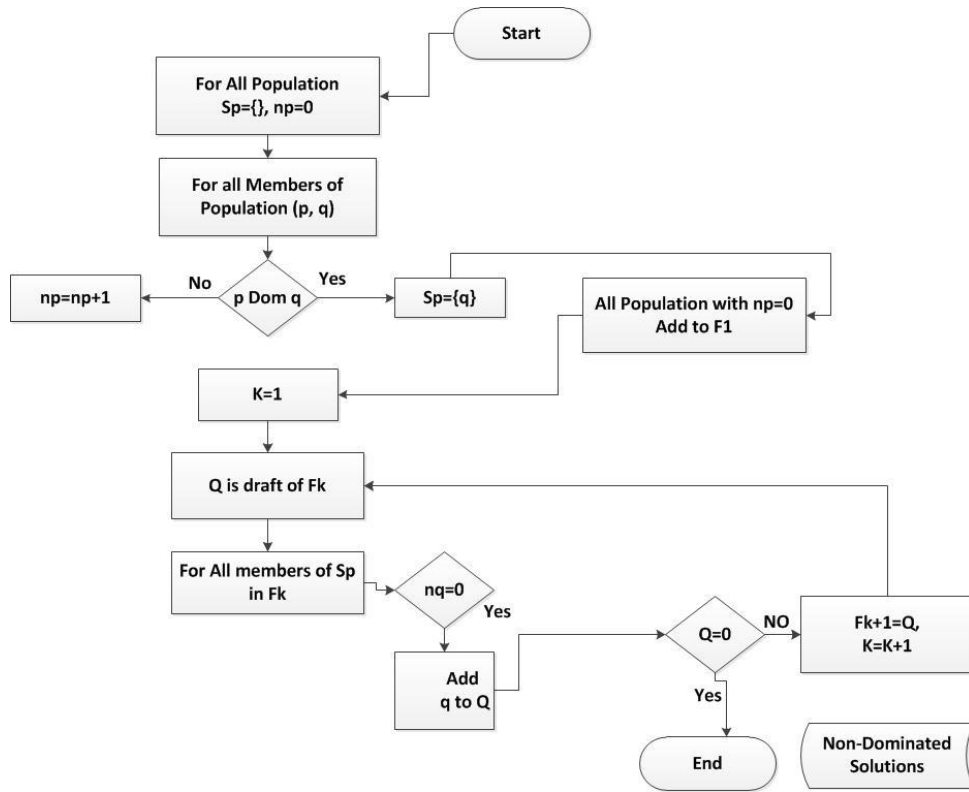


Fig 4: No-dominated Sorting Flowchart

### 3.4 Crowding Distance

One of the advantages of NSGA-II compared to GA algorithm is using crowding distance in sorting of solutions which are obtained. The greater crowding distance leads to increase covered area and diversity of solutions. There are two objective functions in the proposed method for job scheduling problem, vertical axis is Function  $f_2$  with aim to minimize completion time of jobs and horizontal axis is Function  $f_1$  with aim to maximize machine employment that finally with maximizing machine employment obviously. It leads to minimize completion time of jobs. Crowding distance is defined as shown in Figure 5.

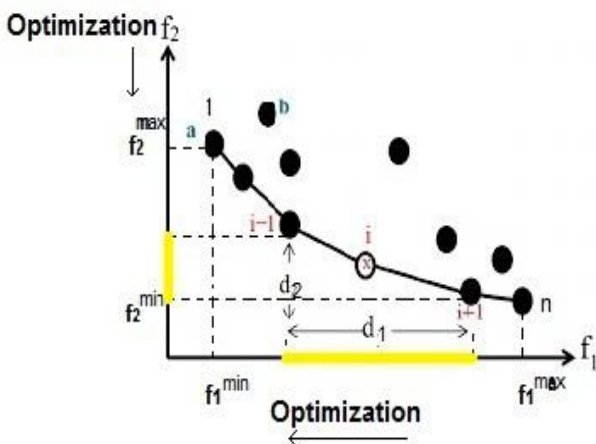


Fig 5: Crowding Distance in our Proposed Method

As shown in Figure 4, proposed model follows to get non-dominated solutions with minimum value of Function  $f_2$  and maximum value of function  $f_1$ . Generally, partial crowding distance is calculated for the great number of samples for the

point  $i$  corresponded to objective Function  $j$  as shown in Equation (2):

$$d_i^j = \frac{|f_j^{i+1} - f_j^{i-1}|}{f_j^{\max} - f_j^{\min}} \quad (2)$$

Total crowding distance is calculated with sum of partial crowding distances which are obtained corresponded to objective functions as shown in Equation (3):

$$d_i = d_i^1 + d_i^2 + \dots + d_i^m = \sum_{j=1}^m d_i^j \quad (3)$$

### 3.5 Bi-objectives in Weighted Sum

In the weighted sum approach, weight values ( $w_i$ ) are generated randomly between  $[0, 1]$  and they are optimized in each iteration. In equation (4), there are bi-objective functions:  $f_1(x)$  and  $f_2(x)$ .

$$\text{Min } F(x) = \sum_{i=1}^2 w_i f_i(x) \quad (4)$$

Where,  $F(x)$  progress in the minimum direction by changing in the weights  $w_1$  and  $w_2$ . Function  $F(x)$  is fitness function in the problem which using WS method that optimizes values obtained from scheduling with the objectives of minimizing machine unemployment and Makespan. The optimized values for job scheduling in MATLAB software by MATLAB Function *fminunc* (Find minimum of unconstrained multivariable function) is used to optimization problems and finding minimum values in Pareto-fronts. Non-dominated solutions in WS method are calculated by using function Dominated. This approach tries to find non-dominated solutions, because the objective is optimization of job scheduling by using two objective functions in order to minimize Makespan and machine unemployment. According to obtain objective functions, two solutions ( $x$  and  $y$ ) are compared with together and the solution overcome to the other which is known as optimized solution and it is sent to

the output of function. The quasi code of how find non-dominated solutions in WS method is shown in Table 3.

**Table 3. Find Non-dominated Solution**

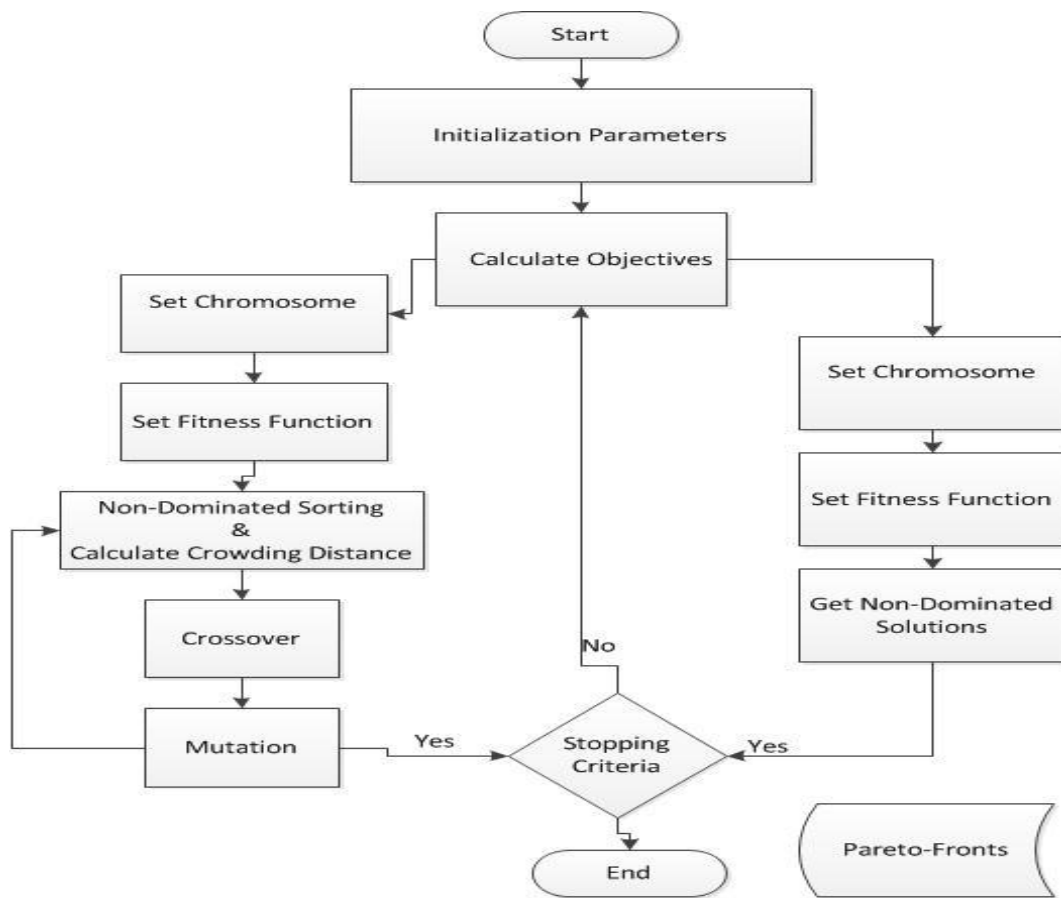
1-Input (Initial Parameters)
2- Get solution
3- For x=1 to Number of All Solutions
3-1- For y=x+1 to Number of All Solutions
3-2- IF Dom (solution(x), solution (y))
3-2-1- Dom (y)=True;
3-3- IF Dom (solution(y), solution (x))
3-3-1- Dom (x)=True;
4- Solution=Solution(~Dom)
5-Output

As shown in Table 4, in order to obtain non-dominated solutions, algorithm tries to achieve minimum Makespan and

maximum machine employment. In WS method, among of solutions in each generation, some of them are selected using fitness function. Evolution is started from a set of entirely randomized variable weights with fitness function and they to be satisfied and the best are selected in each generation. The solutions of each problem are shown by a list of parameters that they are called chromosomes. Chromosomes normally are displayed a simple string of data with two parts. The first part is presented sequence of operations and second part is presented sequence of allocated machines to relative jobs.

#### 4. PROPOSED MODEL

In the proposed model, certain iteration number of runs is as termination condition in WS and NSGA-II methods. Figure 6 shows flowchart of proposed model and how methods solve job scheduling problem.



**Fig 6: Flowchart of Proposed Model**

As start point, initialization parameters are given to the proposed model as shown in Figure 5. Then, bi-objectives ( $Z_1$ ,  $Z_2$ ) are calculated. The flowchart consists of two sections: one section runs NSGA-II and other is WS method to solve FJSSP. NSGA-II section sets chromosome structure using dataset and calculates fitness function. Then, the model utilizes non-dominated sorting and crowding distance to compute ranking of solutions and next crowding distances of them to find optimal samples. The crossover and mutation operators are used to generate next generation of variant solutions while stopping criteria to be satisfied. Also, section of WS method sets chromosomes for its samples and

calculates fitness function. Then, WS method tries to get non-dominated solution by dominating concept as. This procedure will be continued until stopping criteria to be satisfied. Get non-dominated solutions are common characteristic in both of NSGA-II and WS methods. Finally, remained solutions are known as members of Pareto-fronts and outputs of the system. None of Pareto-front solutions have not preferred over another and depending on the conditions, they can be considered as an optimal decision. Multi-objective optimization problems are a subset of those which are made multi-measure decision based approaches in the search space [19].

## 5. EXPERIMENT RESULTS AND DISCUSSION

There is benchmark dataset in literature to solve FJSSP. The processing systems with processor 2.83 GHz and RAM 4G is used in the personal laptop. All programming is performed in MALAB 2010 version. One of advantage of evolution algorithms is non-deterministic origin of them. Non-deterministic nature of evolution algorithms leads to make several runs to achieve significant results. Input parameters to achieve non-dominated solutions are shown in Table 4.

**Table 4. Input Parameters**

Parameters	Value
Population Number	50
Crossover Percent	0.7
Mutation Percent	0.3
Mutation Rate	0.03
Crossover Percent	0.7
w1	[0,1]
w2	[0,1]
Iteration Number	100

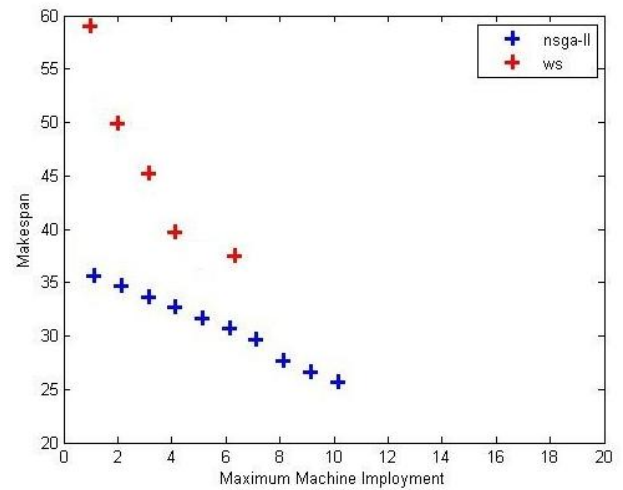
We use several runs of program and variable values of parameters in order to find good performance and optimal solutions in the output based on experiment works. Input parameters are in Table 5 and output of the system is results of processing time based on bi-objectives functions: Makespan and maximum machines employment. In the proposed model, population size is 50 members in the input parameters. Several runs shows that determining more value than 50 leads to high computing time and computational costs and vice versa, when the population size is less than 50 members, proposed algorithms cannot converge to the optimal solution. Researchers show that the population number between 50-100 members indicates the best solution between others [20]. Since, increasing number of population much more than it is to be un-useful and do not to help to solve the problem faster than it was. Table 5 is valid dataset of total FJSSP, because all operations must run on all machines necessarily.

**Table 5. Problem 8\*4**

	O	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>
J <sub>1</sub>	O <sub>11</sub>	3	6	5	4
	O <sub>12</sub>	3	9	3	2
	O <sub>13</sub>	2	3	4	5
J <sub>2</sub>	O <sub>21</sub>	10	5	2	10
	O <sub>22</sub>	5	10	2	3
	O <sub>23</sub>	5	6	3	7
J <sub>3</sub>	O <sub>31</sub>	3	8	2	3
	O <sub>32</sub>	4	5	6	1
	O <sub>33</sub>	3	9	1	10
J <sub>4</sub>	O <sub>41</sub>	8	5	6	3
	O <sub>42</sub>	5	10	6	6
	O <sub>43</sub>	3	7	5	7
J <sub>5</sub>	O <sub>51</sub>	4	4	10	1
	O <sub>52</sub>	9	8	10	1
	O <sub>53</sub>	2	5	7	2
J <sub>6</sub>	O <sub>61</sub>	8	2	7	5
	O <sub>62</sub>	7	9	10	9
	O <sub>63</sub>	7	8	1	1
J <sub>7</sub>	O <sub>71</sub>	8	6	5	10
	O <sub>72</sub>	7	7	9	9
	O <sub>73</sub>	6	2	3	9

J <sub>8</sub>	O <sub>81</sub>	5	1	2	10
	O <sub>82</sub>	6	8	5	1
	O <sub>83</sub>	7	2	1	6

To calculate the results, input parameters are given to the proposed method. The aim is using Ws method and NSGA-II to reach non-dominated solution in Pareto-fronts. Figure 7 shows one sample of bi-objective; Makespan in vertical axis and maximum machine employments in horizontal axis. When Makespan decreases while increasing machine employment leads to more optimization and achieve better non-dominated solutions.



**Fig 7: Bi-Objective Optimization of Problem 8\*4**

As shown in Figure 7, the proposed model solves FJSSP with bi-objective optimization with obtain solutions as descend and non-dominated positions. The proposed chromosome structure has two parts. First part contains processing time of task operations in each gene and second part chromosome structure are genes in which free machines allocated to each operations, randomly. The vertical axis is completion time of jobs (Makespan) and horizontal axis is maximum machines employment. These values are Pareto-front solutions are shown in Figure 6. The results of solution show that WS method is converged with the relatively high gradient to the better non-dominated solutions vs. NSGA-II exerts smooth gradient to the non-dominated solutions. Therefore, Makespan is started from 36 to down in NSGA-II method and it is started from 59 in WS method. Also, in the computing time and consuming memory views, NSGA-II is more performance than WS method with less computing time and memory. As a result, variable weights mechanism in the constraint classic WS method and random selection mechanism in NSGA-II is indicated acceptable performance to resolve FJSSP according to the results of outputs.

One of the disadvantages of WS method is only a Pareto-optimal solution is obtained; all Pareto-front solutions cannot be found and need to know more information than the problem, such as suitable weight, objective solution and etc. The main advantage of the classical methods is their ability to convert single-objective optimization to multi-objective optimization. In contrast, useful advantage of NSGA-II is its meta-heuristic approach to find non-dominated solutions not only one Pareto-front. These approaches of NSGA-II intelligently can change direction in search space more efficient than classical WS method.

## 6. CONCLUSION AND FUTURE WORKS

Job shop scheduling problems belong to the class of multi-objective scheduling problems. Multi-objective scheduling task overcomes to the resource restrictions and allocates machines to task operations of jobs. In this paper, NSGA-II and WS algorithms are used to resolve FJSSP. The objective functions are minimum completion time of jobs and maximum machines employment. The proposed two parts of chromosome structure in methods (2p-WS and 2p-NSGA II) in the face of first challenge; how allocate operations to machines. Also, there is no specified order for processing operations and randomly free machines have been allocated to processing task operations in order to solve next challenge. The variable weights in WS method and random selection mechanism in NSGA-II method are presented to change directions in the search space. The experiment results show that NSGA-II outperforms classical WS method. The random selection mechanism of NSGA-II has more high performance than variable weights mechanism in WS method. Also, WS consumes more computing time than NSGA-II to solve FJSSP while NSGA-II method use less consuming memory than WS to find Pareto-fronts. As future works, using additional objective functions such as minimizing completion time of latest job and calculating start time of jobs are considered.

## 7. REFERENCES

- [1] N. Kundakci, O. Kulak, "Hybrid Genetic Algorithms for Minimizing Makespan in Dynamic Job Shop Scheduling Problem", *Computers & Industrial Engineering*, Vol. 96, pp 31-51, 2016.
- [2] M. R.Garey, D. S. Johnson, R. Sethi, "The complexity of Flow Shop and Job Shop Scheduling", *Journal of Mathematics and Operations Research*, Vol. 1, No. 2, 1976.
- [3] Ryu, J. H., Kim, S. and Wan, H. 2009. Pareto Front Approximation with Adaptive Weighted Sum Method in Multi-Objective Simulation Optimization. *IEEE Proceedings of the 2009 Winter Simulation Conference*, pp. 623-633.
- [4] Kim, Y. and Weck, O. D. 2004. Adaptive weighted-sum method for bi-objective Optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*.
- [5] A. Motaghedi, K. Sabri-Laghaie, M. Heydari, "Solving Flexible Job Shop Scheduling with Multi Objective Approach", *International Journal of Industrial Engineering & Production Research*, Vol. 21, No. 4, 2010.
- [6] G. Zhang, L. Gao, Y. Y. Shi, "An effective genetic algorithm for the Flexible Job-Shop Scheduling problem", *Expert Systems with Applications*, Vol. 38, Vo. 4, pp. 3563-3573, 2011.
- [7] A. Thammano, A. Phuang, "A Hybrid Artificial Bee Colony Algorithm with Local search for Flexible Job-Shop Scheduling Problem", *Procedia Computer Science*, 20, pp. 96-101, 2013.
- [8] K. Z. Gao, P. N. Suganthan, Q. K. Pan, T. J. Chua, C. S. Chong, T. X. Cai, "An Improved Artificial Bee Colony Algorithm for Flexible Job-Shop Scheduling Problem with Fuzzy Processing Time", *Expert Systems with Applications*, Vol. 65, pp. 52-67, 2016.
- [9] X. Li, L. Gao, "An effective hybrid Genetic Algorithm and Tabu Search for Flexible Job Shop Scheduling Problem", *International Journal of production Economics*, Vol. 174, pp. 93-110, 2016.
- [10] G. Kanaani, R. Mogaddam, M. Tabari, Y. Zarandini, M. Arianezhad, "Resolving a New Multi-objective Scheduling Problem in cellular Production System by Using a Fusion Algorithm", *Production and Operation Management Scientific-Research Journal*, No. 2, pp. 1-28, 2012, in Persian.
- [11] Fattahi, P, Arkat, K. Salehi, M. 2012. Scheduling Flexible Job Shop Systems with Setup Time Dependent on the Job Families, 5<sup>th</sup> International Conference on Industrial Engineering, in Persian.
- [12] M. Milosevic, D. Lukic, M. Durdev, A. Antic, S. Borojevic, "An Overview Of Genetic Algorithms for Job Shop Scheduling Problems", *Journal of Production Engineering*, Vol. 8, No. 2, 2015.
- [13] T. Murata, H. Ishibuchi, H. Tanaka, "Genetic Algorithms for Flow-Shop Scheduling Problems", *Computers & Industrial Engineering*, Vol. 30, Iss. 4, pp.1061-1071, 1996.
- [14] L. Asadzadeh, "A Parallel artificial Colony Algorithm for the Job Shop scheduling Problem with a Dynamic Migration Strategy, *Computers & Industrial Engineering*", Vol. 102, pp. 359-367, 2016.
- [15] A. Jamili, "Robust Job Shop Scheduling Problem: Mathematical Models, Exact and Heuristic Algorithms", *Expert systems with Applications*, Vol. 55, pp. 341-350, 2016.
- [16] M. Beheshtiniya, N. Vakili, "Evaluation of Flexible Job Shop Scheduling algorithms and Comparing them with Bipartite Genetic Algorithm", *Modeling in Engineering Journal*, Vol. 13, No. 40, 2016, in Persian.
- [17] N. b. Ho, J. C. Tay, M. Edmund, K. Lai, "An Effective Architecture for Learning and Evolving Flexible Job Shop Schedules", *European Journal of Operational Research*, Vol. 179, pp. 316–333, 2007.
- [18] R. Shahryar, R. T. Hamid, M.A.S. Magdy, "A Novel Population Initialization Method for Accelerating Evolutionary Algorithms", *Computers and Mathematics with Application*, Vol. 53, pp. 1605–1614, 2007.
- [19] Y. Yuan, H. Xu, "Flexible Job Shop Scheduling Using Hybrid Differential Evolution Algorithms", *Computers & Industrial Engineering*, Vol. 65. Iss. 2, pp 246-260, 2013.
- [20] Ghaderi, H. Lotfi, S. and Esfahlan. M. 2011. An Introduction to Some Artificial Optimization Methods, First Edition, Islamic Azad University, Shabestar, Iran, in Persian.