International Journal of Engineering works
Kambohwell Publisher Enterprises
ISSN: 2409-2770 Vol. 1, Issue. 2, PP 26-31, Nov. 2014
www.kwpublisher.com

A New Method for Solving Dynamic Flexible Job Shop Scheduling Problems Integrating Genetic Algorithm and Priority Rules

M. Sanatizadeh, M.A. Badamchizadeh, S. Khanmohammadi, D. Ashtiani Haghighi

***Abstract*—** Dynamic flexible job shop scheduling problems has been one of the important and strongly NP-hard problem of manufacturing systems for many years. Most of the proposed algorithms are based on priority rules; By using these rules, the arrived jobs go to a long queue of waited jobs and sometimes it takes a long time for a job to be processed. In this paper a new approach, integrating of priority rules and genetic algorithm is presented, by decomposition of a dynamic problem to smaller dynamic and static problems. A module converts the queue of dynamic jobs to static, and then a genetic algorithm has been used to improve some objective functions.

Keywords- dynamic flexible job shop, priority rules, genetic algorithm, static

I. Introduction

Scheduling is the allocation of resources for performing a set of tasks. Resources may be machines in shop floor, runway in an airport, crews at a construction site or processing units in a computing environment. Tasks may be operations in shop floor, takeoffs and landing in an airport, stages in a construction project or computer programs to be executed [1].

Job shop scheduling is a branch of the production scheduling, which is well known as combinatorial optimization problem. The job shop scheduling problem is to determine a schedule of jobs that have pre-specified operation sequences in a multi-machine environment. In the classical job shop scheduling problem (JSP), n job are processed for completion on m unrelated machines. For each job, technology constraints specify a complete distinct routing which is fixed and known in advance. Flexible job shop scheduling problem (FJSP) is an extension of the classical JSP which allows an operation to be processed by any machine from a given set. The scheduling problem of FJSP consists of a routing sub-problem that is assigning each operation to a machine out of a set of capable machines and the scheduling sub-problem, which consists of sequencing the assigned operations on all machines in order to obtain a feasible schedule, minimizing a predefined objective function. The FJSP is a much more complex version of the JSP, so the FJSP is strongly NP-hard and combinatorial [2]. Job scheduling can be classified into two groups, static (offline) and dynamic (online). In static JSPs all jobs are ready at time zero and in dynamic JSPs, job release times are not fixed at a single point, that is, jobs arrive at various times. Dynamic JSPs can be further classified as deterministic or stochastic based on the manner of specification of job release times. Deterministic JSPs assume that the job release times are known in advance. In stochastic JSPs, job release times are random variables described by a known probability distribution [3]. The scheduling based on static variables performs weak in real situation so a large number of papers have used different priority rules to overcome the problem of unforeseen incidents but the priority rules procedure is a time consuming method and sometimes results in poor performance. On the other hand, in static scheduling by ignoring nondeterministic parameters and unpredictable events a near optimal solution can be achieved.

We will present compromise between online and offline systems in the form of a priority rule-based optimization system that combines strength of both methods [4]. L. De Giovanni and F. Pezzella [5] proposed an improved genetic algorithm for flexible job shop scheduling in distributed environment. F. Pezzella, G. Morganti and G. Ciaschetti [6] presented a genetic algorithm and proved the efficiency of their method with respect to other genetic algorithms. V .Vinod and R.Sridharan [7] provided a simulation study of dynamic job shop scheduling problems and presented some scheduling rules. Shyh-Chang Lin, erik D. Goodman and William F. Punch, III [3] compared their GA algorithm for dynamic problems with some priority rules. Xili Chen, Hao Wen Lin and Tomohiro Murata [8] proposed a dispatching rule by combining different elementary dispatching rules. Erik Pitzer et al. [4] used a solution archive of priority rules and presented a new approach by optimizing priority rules.

This paper is organized as follows. Section II defines the assumptions, job data and problem description. Section III describes dynamic scheduling, rules and objective functions. A new approach and algorithm is proposed in section IV then, the result of computational analysis is provided in section V. Finally conclusions are presented in section VI.

M. Sanatizadeh: Imam Khomeini International University, sanatizadeh_83@yahoo.com. .
M.A. Badamchizadeh : Faculty of Electrical and Computer Engineering , University of Tabriz, Iran, mbadamchi@tabrziu.ac.ir.
S. Khanmohammadi: Faculty of Electrical and Computer Engineering , University of Tabriz, Iran, khan@tabrziu.ac.ir
D. Ashtiani Haghighi: University of Tabriz, Iran, d.ashtiani.h@gmail.com

II. Assunmptions Made in This Study

The assumptions of model are as follow:

- There are n independent jobs that are indexed by i
- Each job i has Oi operations and the operations' sequence is given by Oij for $j = 1, \ldots, Oi$ where Oij is jth operation of ith job so $O_{i\ j+1}$ cannot be processed before Oij.
- There are 8 machines indexed by $k = 1, \ldots, 8$.
- Machines never breakdown and are always available.
- All operations can be processed on each machine.
- A started operation cannot be interrupted (no preemption condition)
- Each machine can process at most one operation at any time (resource constraint)
- There is no restriction on queue length at any machine.

A. Job Data

In this paper we are going to study dynamic flexible job shop scheduling problem. Most of the studies have used between four and ten machines. Hence, in the present study, a job shop system consisting of 8 machines is chosen. The machines are identical and perform all different operations. Jobs are different and each job composed of 6 operations with nondeterministic arrival and processing time. There are 8 similar machines and each operation can be performed on each machine.

B. Arrival Time

The distribution of the job arrival process closely follows the Poisson distribution; hence, the time between arrivals of jobs is exponentially distributed [1]. The mean of this exponentially distribution is determined for a specified shop utilization percentage and the processing requirements of the jobs. Thus, the mean interarrival time of jobs is obtained using the following relationship:

$$a = \frac{\mu_p \mu_g}{\rho n_m} \tag{1}$$

Where a is the mean interarrival time, μ_p the mean processing per operation, μ_g the number of operations per job, ρ the shop utilization and n_m the number of machines in the shop. In the present study $\mu_p = 30$, $\mu_g = 6$, $n_m = 8$, and $\rho = 0.90$ [5].

C. Due date

The processing time is exponentially with mean of 30. Due date of a job determined according to the following equation:

$$d_i = a_i + k\left(\sum_{j=1}^{6} p_{ij}\right) \tag{2}$$

where d_i, due date of job i, a_i arrival time of job i and p_{ij} is the jth operation's processing time of job i. k is due date factor and this study have been done for two values of k (i.e., $k = 4$ and 5).

III. Dynamic Scheduling

When a machine becomes free, it has to be decided which of the waiting jobs (if there is any in the queue of machine) is to be processed on the machine. For making this decision, a scheduling rule is used to assign a priority value to each waiting jobs. The job having the highest priority value is selected for processing next [7]. The scheduling rules which are used in this study are described as follow:

(1) FIFO: First In First Out

$$Z_{it} = a_{im} \tag{3}$$

Where, Z_{it} is priority value of job i at time t, t is the time at which the priority value are calculated, a_{im} is arrival time of job i at machine m, N_{tm} is set of jobs, waiting for processing in the queue of machine m at time t. The highest priority is given to the job i^* with:

$$Z_{i*t} = Min\,\{Z_{it} | i \in N_{tm}\} \tag{4}$$

Using FIFO rule, the jobs are processed in order they arrive at the machine.

(2) SPT: Shortest Processing Time

$$Z_{it} = p_{ij} \tag{5}$$

Where the highest priority is given to the job i^* with:

$$Z_{i*t} = Min\,\{Z_{it} | i \in N_{tm}\} \tag{6}$$

The job with the shortest processing time for the imminent operation is selected.

(3) EDD: Earliest Due Date

$$Z_{it} = d_i \tag{7}$$

Where the highest priority is given to the job i^* with:

$$Z_{i*t} = Min\,\{Z_{it} | i \in N_{tm}\} \tag{8}$$

Using this rule, the job with earliest due date is selected.

(4) MWCR: Most Work Content Remain

$$Z_{it} = \sum_{j=1}^{r} p_{ij} - \sum_{q=1}^{c} p_{iq} \tag{9}$$

Where the highest priority is given to the job i^* with:

$$Z_{i*t} = Max \{Z_{it} | i \in N_{tm}\} \quad (10)$$

Where, c is the number of operations which are completed at time t and r is the number of operation of job i.
Using this rule, the job which has the most work content remain is selected.

In dynamic JSPs, minimizing makespan (total time of finishing all jobs) is less important because the scheduling horizon is open and the makespan gives no credit for jobs that finish well before the last one finishes [3] so we choose the following objective functions:

(1) Flowtime: $F_i = C_i - a_i$
(2) Mean flowtime: $\bar{F} = [\frac{1}{n}] \sum_{i=1}^{n} F_i$
(3) Lateness: $L_i = C_i - d_i$
(4) Tardiness: $T_i = max\ (0, L_i)$
(5) Mean tardiness: $\bar{T} = [\frac{1}{n}] \sum_{i=1}^{n} T_i$

Where F_i is flowtime, T_i is tardiness, C_i is completion time and L_i is lateness of job i.

IV. Proposed Algorithm

According to the previous studies [5] and [6], genetic algorithm (GA) is the best method to solve static problems. GA is a local search algorithm that follows the evolution paradigm; the strength of GA with respect to other local search algorithms is due to the fact that in a GA framework more strategies can be adopted; then a more variable search space can be explored [6]. So an acceptable and near-optimal solution can be achieved in a reasonable time. In dynamic flexible job shop problems, arrived jobs should be decided to be processed on proper machine in order to optimize an objective function. There are a large number of priority rules for real time (online) scheduling. Jobs go to shop floor and for each operation, related machine is chosen based on algorithm. When speeds of arriving are more than processing time, waiting jobs make a long queue so, the more jobs arrive the longer queue is made. Our proposed approach consists of two parts; each new job arrives in stochastic interval time to the shop floor. If related machine is idle the arriving job will be processed immediately based on a priority rule; but most of the time, demand is more than production so the job should wait in a queue of related machine. If at $time = t_0$, a job stays more than 100 units of time in shop floor, second part of algorithm will starts and all waited jobs queues will be considered as a static event that starts at $time = t_0$; then the static event will be scheduled based on genetic algorithm.

TABLE I. Information of Jobs For Example

Jobs	*Arriving time*	*Start time*	*End time*	*Time=150*
Job1	0	0	50	Completed
Job2	12	30	85	Completed
Job3	35	73	120	Completed
Job4	50	90	Under process	-
Job5	55	91	Under process	-
Job6	73	Waited	-	-

A. *Static Part*

In dynamic problems, jobs arrive dynamically and the start time is $t = 0$. During scheduling based on GA, it is assumed that, the problem is static. GA's runtime is considered much less than processing time of operations so there is no disturbance during GA programming and the only objective is minimizing of the makespan.
For example, in a shop, the start time of arriving jobs is at $time = 0$. Jobs are processed based on a priority rule, up to $time = 150$. At $time = 150$, six jobs have arrived. First, second and third jobs are completed, the fourth and fifth jobs are under processing and the remained job and operations are waiting in buffers. Table I. shows, the information of this example.
Flowtime of each completed job and the time of staying each uncompleted job are calculated after finishing each operation during scheduling based on priority rule. If one of the flow times or staying times is more than 100 units of time, all the waiting jobs and operations will go to GA module for scheduling based on proposed GA. As shown in table I, at $time = 150$ the job4 stays more than 100 units of time then, GA based scheduling starts.
Remained jobs and operations can be processed on 8 machines at $start\ time = 150$. Machines are chosen according to the proposed genetic algorithm. A GA [6] proposed by F. Pezzella is used in our scheduling method. The structure of GA can be described as follows:

1) Coding: the genes of the chromosomes consist of three elements:
   - i is the number of job
   - j is the number of operation of job i
   - k is the machine assigned to operation

   So the gene is formed by a triple (i, j, k).
   The order in which they appear in the chromosome describes the sequence of operations. Each chromosome represents a solution for the problem.
2) Initial population: the initial chromosomes are obtained by a mix of two assignment procedures (global minimum and random permutation of jobs and machines) and a mix of three dispatching rules for sequencing:
   - Randomly select a job
   - Most work remaining
   - Most number of operation remaining

ISSN: 2409-2770

3) Fitness evaluation: the makespan is computed for each chromosome. The chromosomes with lower makespan value are better ones. .
4) Selection: at each iteration, the best chromosome are chosen for reproduction among three different methods i.e., binary tournament, n-size tournament and linear ranking.
5) Offspring generation: the new generation is obtained by changing the assignment of operations to the machines (assignment crossover, assignment mutation,) and by changing the sequencing of operations (POX crossover and PPS mutation). The Precedence Preserving Order-based crossover (POX) and Precedence Preserving Shift mutation (PPS) are proposed by Kacem [9] These rules preserve feasibility of new individuals.
6) Stop criterion: Fixed number of generation is reached. If the stop criterion is satisfied, the algorithm ends and the best chromosome is given as the best solution.

Different values are tested and the best ones are the following values:

- Population size: 20
- Number of generations: 200
- Rate of initial assignments with global minimum method: 10%
- Rate of initial assignments with random permutation method: 90%
- Rate of initial sequences with random rule: 20%
- Rate of initial sequences with MWR rule: 40%
- Rate of initial sequences with MOR rule : 40%
- POX crossover probability: 45%
- Assignment crossover probability:45%
- PPS crossover probability: 5%
- Assignment mutation probability:5%

B. Dynamic Part

In the interval times between static events, the jobs which are in queues are scheduled based on priority rules which are explained in section II. A. Each operation of a job can be processed on every machine, so the machine with the lowest processing time will be chosen. After processing of each operation the condition of static event is checked then next operation is decided to be started. The flowchart of proposed algorithm is shown in Fig 1.

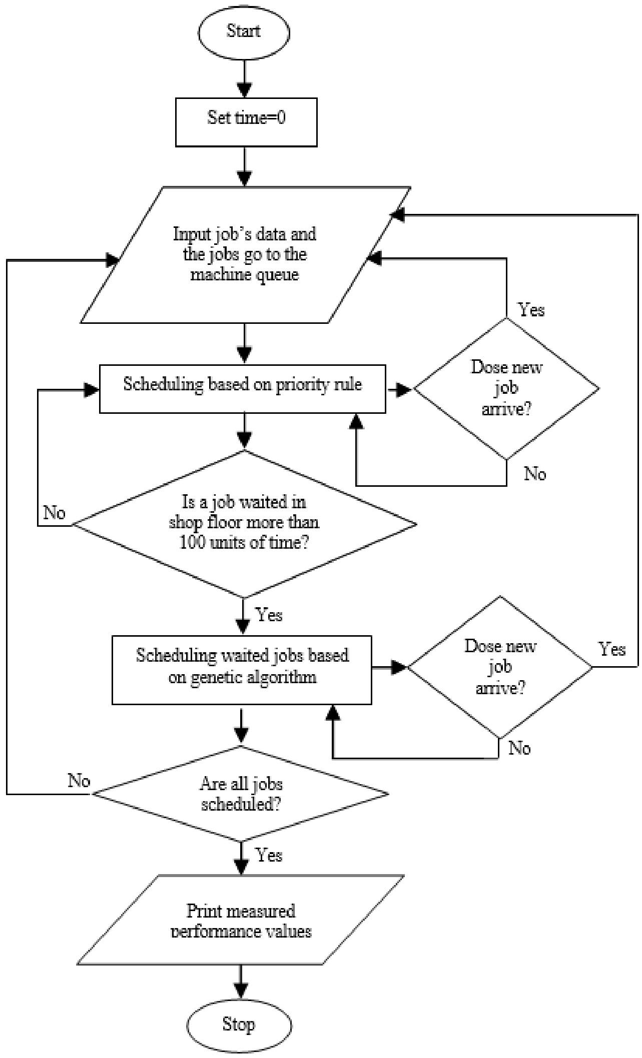


Figure 1. Flowchart of proposed algorithm

V. SIMULATIN RESULT

In this section, the performance of the proposed algorithm is compared with traditional priority rules. These rules are FIFO, SPT, MWCR and EDD. For each priority rule and according proposed algorithm, the values of mean flowtime, number of tardiness, mean of tardiness and number of using GA are computed. Number of jobs is considered 300 in this study. Table II shows the comparison results of simulations by using the introduced algorithm with considering k=5 and Table III. shows the results for k=4. As the tables show, mean of flowtime and mean of tardiness are improved considerably in all proposed algorithms for k=4 and k=5. Number of tardiness is improved in all cases except PMWCR(Proposed algorithm) for k=4. The best values for flowtime, number of tardiness and mean of tardiness are related to PFIFO (Proposed algorithm) and PEDD method. For each objective function the following ranking is obtained:

Mean flow time: PFIFO < PEDD < PMWCR < PSPT

Number of GA using: PEDD < PFIFO < PSPT < PMWCR

For k=4:

Number of tardiness: PFIFIO < PEDD < PSPT, PMWCR

Mean of tardiness: PFIFIO < PEDD < PMWCR <PSPT

ISSN: 2409-2770

For k=5:
Number of tardiness: PFIFO < PEDD < PSPT, PMWCR
Mean of tardiness: PFIFO < PEDD < PMWCR < PSPT

According to simulation results, PFIFO and PEDD methods are lead to better performance.

TABLE II. COPMPARISION OF PROPOSED ALGORITHM AND PRIORITY RULES FOR K=5

Priority rule	Mean of flowtime	Number of tardiness	Mean of tardiness	Number of using GAs
FIFO	990.70	277	704.09	-
PFIFO	79.86	16	1.31	38
SPT	507.32	82	341.72	-
PSPT	122.93	70	10.79	49
MWCR	953.33	77	790.72	-
PMWCR	114.58	70	8.33	71
EDD	871.64	271	587.59	-
PEDD	93	21	2.22	28

TABLE II COPMPARISION OF PROPOSED ALGORITHM AND PRIORITY RULES FOR K=4

Priority rule	Mean of flowtime	Number of tardiness	Mean of tardiness	Number of using GAs
FIFO	990.70	287	759.42	-
PFIFO	79.86	32	2.94	38
SPT	507.32	92	361.36	-
PSPT	122.93	106	17.86	49
MWCR	953.33	86	810.86	-
PMWCR	114.58	106	14.89	75
EDD	871.64	279	641.81	-
PEDD	93	42	5.46	28

Number of jobs are 300 and and flowtimes are calculated for each job. Figure II shows flowtimes for each job after using PFIFO (Proposed FIFO) it can be seen, the maximum value for flowtime is 178 unit of time. Figure III shows the result of using PSPT (Proposed SPT). Maximum flowtime is 355 which is much more than PFIFO. The result of applying PEDD (Proposed EDD) is shown in figure V. Maximum flowtime is 297. Finally the result of PMWRC is shown in figure IV and maximum flowtime is 275.

VI. CONCLUSION

In this study a new approach integrating GA and dynamic rules has been presented and its performance is compared with traditional rules. As the computational results show in all methods except PMWCR, our proposed algorithm performs better than original rules. The simulation results show the PFIFO is the best algorithm among all rules. In large manufacturing systems with large amount of productions, more benefit can be obtained for the system even with 1% of improvement in performance. In future, we are going to extend priority rules to more complex ones in order to achieve better performance.

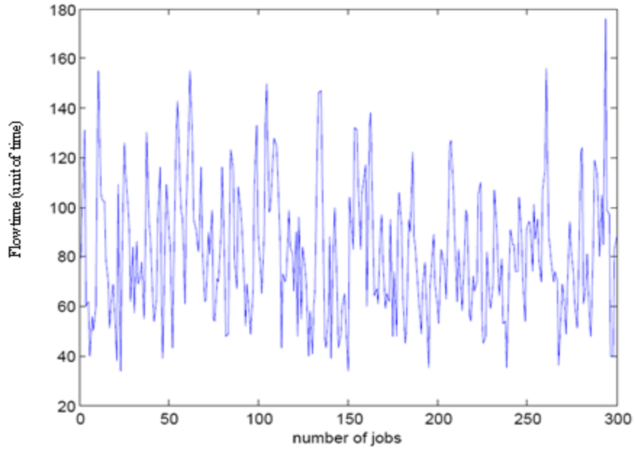


Figure II. Flowtime after using PFIFO for each job

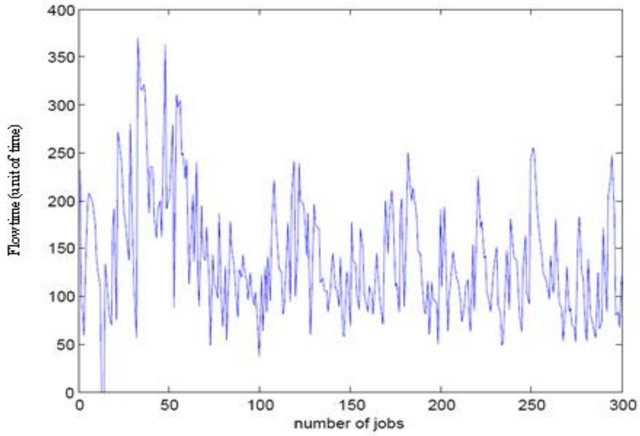


Figure III. Flowtime after using PSPT for each job

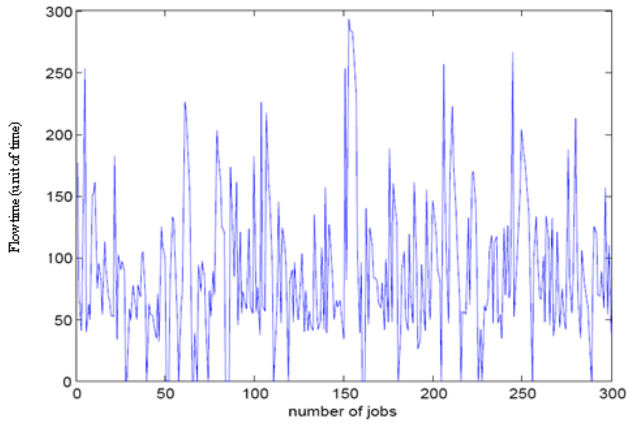


Figure V. Flowtime after using PEDD for each job

ISSN: 2409-2770

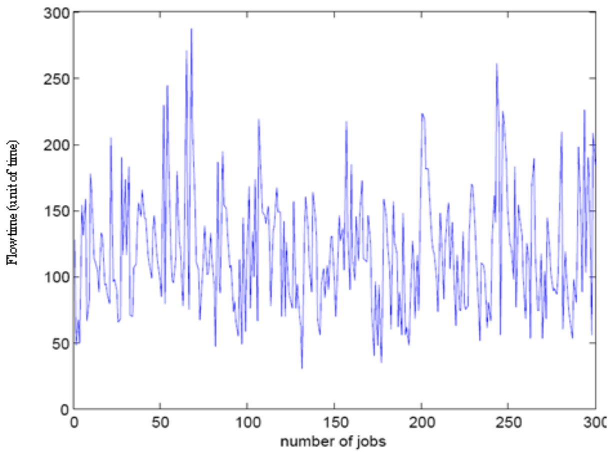


Figure IV. Flowtime after using PMWCR for each job

REFERENCES

[1] V. Vinod, R. Sridharan, "Scheduling a Dynamic Job Shop Production System with Sequence-dependent Setup: An Experimental Study" Robotic and Computer-Integrated Manufacturing 24(2008) 435-449, 2008

[2] Parviz Fattahi, Alireza Fallahi: "Dynamic Scheduling in Flexible Job Shop systems by Considering Simultaneously Efficiency and Stability". CIRP Journal of Manufacturing Science and Technology 2(2010) 114-123, 2009

[3] Shyh-Chang Lin, Erik D. Goodman, William F. Punch, III, " A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problem" .Genetic Algorithms Research and Application Group, The 17th International Workshop on 1997

[4] Erik Pitzer, Andreas Behem, Meachel Affenzeller, Helega Heiss, Markus Vorderwinkler: "Production Fine Planning Using a Solution Archive of priority Rules", LINDI 20011.3rd IEEE International Symposium on Logistics and Industrial Informatics. August 25-27, Budapest , Hungry, 2011

[5] L. De Giovanni ,F. Pezzella: "An Improved Genetic Algorithm for Distributed and Flexible Job-shop Scheduling problem", European Journal of Operational Research 200(2010) 395-408, 2009

[6] F. Pezzella, G. Morganti, G. Ciaschetti: "A Genetic Algorithm for the Flexible Job-shop Scheduling Problem", Computer and Operational Research 35 (2008) 3202-3212, 2007

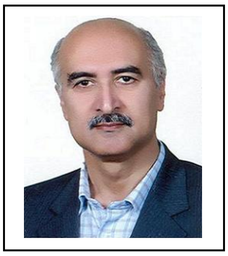
[7] V. Vinod, R. Sridharan:"Simulation Modeling and Analysis of Due-date Assignment Methods and Scheduling Decision Rules in a Dynamic Job Shop Production Systems" , Int. Production Economics 129(2011) 127-146, 2010

[8] Xili Chen, Hao Wen Lin, Tomohiro Murata:" Dispatching Rule Composition Method for Single Machine Multi Objective Scheduling", ICIEA 2009. 4th IEEE Conference on Industrial Electronic and Application ,2009

[9] I. Kacem , S. hammadi, P. Borne ."Approach by Localization and Multiobjective Evolutionary optimization for Flexible Job-shop Scheduling Problem". IEEE Transactions on Systems, Man, and Cybernetics, Part C 2002; 32(1): 1-13 ,2002

Mahsa Sanatizadeh received the B.S. in Control Enginnering from Isfahan University of thechnology, Iran and M.S. degree from University of Tabriz, Iran in 2008 and 2012, respectively. Now she is Ph.D. student in Imam Khomeini International University. Her research interests incluse job scheduling, fuzzy systems and fuzzy control. Email: sanatizadeh_83@yahoo.com

Mohammadali Badamchizadeh was born in Tabriz, Iran, in December 1975. He received the B.S. degree in Electrical Engineering from University of Tabriz in 1988 and the M.Sc. degree in Control Enginnering from University of Tabriz in 2001. He received the Ph.D. degree in Control Engineering from University of Tabriz in 2007. He is now an Associate Professor in the Faculty of Electrical and Computer Engineering at University of Tabriz. His research interests include Hybrid dynamical systems, Stability of systems and Adaptive Control. Email: mbadamchi@tabrziu.ac.ir



Sohrab Khanmohammadi received his B.S. degree in Industrial Engineering from Sharif University, Iran in 1977 and M.Sc. degree in Automatic from University Paul Sabatie, France in 1980 and his Ph.D. degree in Automatic from National University, ENSAE, France in 1983. He is now a professor of Electrical Engineering at University of Tabriz. His research interests are Fuzzy control, Artificial Intelligence applications in Control and Simulation on industrial systems and human behavior. Email: khan@tabrziu.ac.ir

Donia Ashtiani Haghighi received the B.S. in Control Enginnering from Ghilan University, Iran and M.S. degree from University of Tabriz, Iran in 2008 and 2012, respectively. Her research interests incluse job scheduling, fuzzy systems and fuzzy control. Email. d.ashtiani.h@gmail.com