Some Heuristic Rules for Job Shop Scheduling Problem

Bou Shaala, A., Shouman, M. A., and Esheem, S.

Industrial and manufacturing systems Engineering, Garyounis University, Libya

Abstract

Scheduling job shop problem is searching for good schedule or optimization with the goal of finding the best schedule. The main objective of this paper is to develop some heuristic rules for job shop scheduling problem. These heuristic rules are developed and tested in their scheduling performances with other common rules in benchmark problems. Six measures are considered for performance evaluation. Seventy instance cases are used as jobs for scheduling process. The results are compared with eighteen common heuristic rules for the considered test cases. The proposed heuristics are found to be promising in their effectiveness in scheduling job shop problem. A computer program could be designed to test the validity of the proposed heuristics.

1. Introduction

Scheduling is a well-known problem that deals with the efficient allocation of resources with respect to time in order to perform a collection of tasks. In manufacturing, tasks correspond to parts that need to be processed on a set of machines [1]. Scheduling can be considered as a searching or optimization problem, with the goal of finding the best schedule. Job shop scheduling is one of the widely studied and most complex combinatorial optimization problems. A vast amount of research has been performed in this particular area to effectively schedule jobs for various objectives [2]. In job shop scheduling problem, a set of jobs is given and a set of machines. Each machine can handle, at most, one job at a time. Each job consists of a chain of operations, each of which needs to be processed during an uninterrupted time period of a given length on a given machine. The purpose is to find a schedule, that is, an allocation of the operations to time intervals on the machines, which has a minimum duration required to complete all jobs [3]. Therefore, the exact methods such as the branch and bound method, dynamic programming and constraint logic programming need a lot of time to find an optimal solution. So, heuristic method could be better than exact ones in handling such problems. Realistically, the satisfaction is achieved by obtaining a good solution near the optimal one. Search techniques such as Genetic Algorithms (GAs), Simulated Annealing (SA) or Tabu Search (TS) are able to solve the job-shop scheduling problem [2]. Dispatching Rules (DRs) have been applied consistently to scheduling problems. They are procedures designed to provide good solutions to complex problems in real-time. Many authors claim that priority dispatching rules can be successfully used in solving large JSSPs and even other scheduling problems [4, 5]. In real applications, priority dispatching rules are actually the most widely used.

There are several dispatching rules which present a significant optimization capacity as mention in references [6, 4, and 7].To improve the scheduling performance, they proposed the usage of evolutionary algorithms to generate better, dispatching rules (DRs) which are mathematical combinations of various simple dispatching rules. Mahanim et al [8] used genetic algorithm (GA) with some modifications to deal with problem of job shop scheduling which generated an initial population randomly including the result obtain by some well-known priority rules such as shortest processing time and longest processing time. Nima et al [3] presented a method for extracting rules from the solutions of a GA, which describe its behavior by using data mining; attribute oriented induction technique [9]. Kuczapski et al presented an efficient method of enhancing genetic algorithms (GAs) for solving the Job-Shop Scheduling Problem (JSSP), by generating near optimal initial populations. Since the choice of the initial population has a high impact on the speed of the evolution and the quality of the final results, the solution scenario would be focused on generating its individuals using genetically evolved priority dispatching rules. Huiyuan et al [10] developed job shop

scheduling problem model, according to the actual factors of mass injection molding processing job shop scheduling. A heuristic active algorithm combined with priority rules is employed to give the solution. Most research on scheduling concerns with a single objective: the optimization of makespan (the completion time of all jobs). Real-life scheduling problems, however, often require the decision maker to consider a number of criteria before arriving at a decision. A solution that is optimal with respect to one given criterion might be a poor candidate for some others like mean flow-time (the average response of the schedule to the individual demands of jobs for service), tardiness (the lateness of any job measures the conformity of the schedule to that job's committed date), etc. Therefore, the trade-offs involved in considering several different criteria provide useful insights for decision makers. Surprisingly, research in this important field has been scarce in comparison with the research in single-criterion scheduling. The goal of multi-objective JSSP is to find as many different schedules as possible that are near-optimal and non-dominated with regard to different objectives [2]. The considered intention in this paper is to find out a good solution for job-shop scheduling problems based on priority rules solution procedure or scenario and based on different measuring performance criteria for the same solution.

2. Job-Shop Scheduling Structure

JSSP can be stated as a set of n jobs to be processed on a set of m machines, where each job j visits a number of machines in a predetermined order. The processing times for each job at each machine are given and no machine can process more than one job at a time. If a job is started on a machine, then it cannot be interrupted [3]. The problem is finding a schedule of the jobs on the machines. The assumptions of the present problem are:

1. Every job has a unique sequence on m machines. There are no alternate routings.
2. There is only one machine of each type in the shop.
3. Processing times for all jobs are known and constant.
4. All jobs are available for processing at time zero.
5. Machine absences are not allowed.

6. Transportation time between machines is zero.
7. Each machine can perform only one operation at a time on any job.
8. An operation of a job can be performed by only one machine.
9. Operation cannot be interrupted.
10. A job does not visit the same machine twice.
11. An operation of a job cannot be performed until its preceding operations are completed.
12. Each machine is continuously available for production.
13. There is no restriction on queue length for any machine.
14. There are no limiting resources other than machines/workstations.
15. The machines are not identical and perform different operations [3].

3. Common Priority Rules

Priority dispatching rules are actually the most widely used for solving JSSP where all the operations available to be scheduled are assigned a priority. The operation with the highest priority is chosen to be sequenced. A priority dispatching rule is a simple mathematical formula that, based on some processing parameters, specifies the priority of operations to be executed. The common usual processing parameters are shown in Table 1 where j denotes job to be processed on each machine .The commonly used heuristics in the current work and of always promising results as indicated by many researchers and of a significant optimization capacity are listed in Table 2 [9, 10, 11].

Table1. Processing parameters for priority dispatching

Symbol	Description
r_j	Job arrival time the moment when the job j arrives at the machine
w_j	Job weight the importance of the job j
p_{jm}	Processing time the time needed to process job j on machine m
n_j	Remaining operations the number of remaining operations of job j
R_j	Remaining work the time needed to complete the job's remaining operations
P_j	Total work, the time needed to execute all operations of the job j
d_j	Due date the moment when the job j should be finished

4. Proposed Priority Rules

Nine priority rules are proposed in the current work and tested on seventy instance cases

representing job shop problem. Fifty percent of these seventy cases are classical job shop problem while the other fifty percent are flexible job shop problem.

Table 2.Common priority rules.

Expression	Description
Shortest Processing Time (SPT)	The job with shortest time on machines selected. $P_i \leq P_{i+1} \leq P_{i+2} \leq \ldots\ldots\ldots \leq P_n$
Longest Processing Time (LPT)	The job with longest processing time on machine is selected. $P_i \geq P_{i+1} \geq P_{i+2} \geq \ldots\ldots\ldots \geq P_n$
Minimum Slack Time Per Operation (MINSOP)	Time remaining until the due date - processing time remaining
Minimum Due Date (MINDD)	The job with the earliest due date is processed first. $D_i \leq D_{i+1} \leq D_{i+2} \leq \ldots\ldots\ldots \leq D_n$
RANDOM (random selection)	Selects the next job to be processed at random.
Critical Ratio (CR)	Remaining due date /Remaining processing Time
Most work remaining (MWKR)	select the operation associated with the job of the most work remaining to be processed.
Least work remaining (LWKR)	select the operation associated with the job of the least work remaining to be processed
Most Operation Remaining (MOPNR)	select the operation that has largest number of successor operations.
Shortest Remaining Minimum Processing Time (SRMPT)	Min (processing time remaining – min processing time).
Longest Remaining Maximum Processing Time (SRMPT)	Max (processing time remaining – max processing time).
Fewest Number of Operation Remaining (FOPNR)	Min Ratio (operation remaining for job /sum of operations).
Greatest Number of Operation Remaining (GOPNR)	Max Ratio (operation remaining for job /sum of operations).
SPT/WKR	(smallest weight ratio of processing time to work remaining)
Shortest Weight Process Time (SWPT)	$\dfrac{P_j}{W_j} \leq \dfrac{P_{j+1}}{W_{j+1}}$
Longest Weight Process Time (LWPT)	$\dfrac{P_j}{W_j} \geq \dfrac{P_{j+1}}{W_{j+1}}$
Shortest Weight Mean Processing Time (SWMPT)	$\dfrac{P_k}{W_k} \leq \dfrac{P_{k+1}}{W_{k+1}}$
Longest Weight Mean Processing Time (LWMPT)	$\dfrac{P_k}{W_k} \geq \dfrac{P_{k+1}}{W_{k+1}}$

Also the previous eighteen priority rules are used for scheduling the instance cases under consideration. The rendered results on the instance cases under consideration by either the common priority rules (eighteen) or the new proposed priority rules are compared under six measuring performance criteria. These measuring performance criteria are makespan, lateness, number of late/tardy jobs, tardiness, earliness, and number of early jobs. Also two scenario conditions are considered for the experimental study. These scenarios are based on the considered weight of each job where this weight is based on either the significant importance or based on its relative time to other times. Based on this assumption the first experimental scenario has been done based on randomized weight values for the jobs to be scheduled while the second scenario is based on relative time of the job to other jobs times. These two scenarios are considered for the experimental work in this study and compared to see the influence of weight on the scheduling process. However the mathematical equations for the proposed rules are listed as follows:

1. Minimum Weighted Due Date (MINWDD): for this rule, the job with the earliest weighted due date is processed first.
$W_i D_{i} \leq W_{i+1} D_{i+1} \leq W_{i+2} D_{i+2} \leq \ldots\ldots \leq W_n D_n$

2. Weighted Shortest Processing Time (WSPT):for this rule, job with weighted shortest processing time on machine is selected.
$W_i P_i \leq W_{i+1} P_{i+1} \leq W_{i+2} P_{i+2} \leq \ldots\ldots \leq W_n P_n$

3. Weighted Longest Processing Time (WLPT): for this rule, the job with weighted longest processing time on machine is selected.
$W_i P_i \geq W_{i+1} P_{i+1} \geq W_{i+2} P_{i+2} \geq \ldots\ldots\ldots \geq W_n P_n$

4. Minimum Weighted Slack Time Per Operation (MINWSOP): The weighted time remaining until the due date minus weighted processing time remaining.

5. Weighted Critical Ratio (WCR):
$$WCR = \frac{\text{Remaining due date}}{\text{Remaining processing time}} \times weight$$
$W_i CR_i \leq W_{i+1} CR_{i+1} \leq W_{i+2} CR_{i+2} \leq \ldots\ldots \leq W_n CR_n$

The job with the minimum value of WCR is scheduled next .This rule calculates the weighted ratio of demand time to supply time. When the weighted ratio exceeds a value of 1, there is sufficient time a available to complete The job if the queue times are properly managed. If the weighted ratio is less than 1, the job will be late unless processing times can be compressed. The WCR rule has slightly more intuitive appeal, since the WCR itself has a precise meaning.

6. Fewest Weighted Number of Operation Remaining (FWOPNR).

7. Greatest Weighted Number of Operation Remaining (GWOPNR).

8. Most Weighted work remaining (MWWKR): the operation associated with the job having the most weighted work remaining to be processed.

9. Least Weighted work remaining (LWWKR): the operation associated with the job having the least weighted work remaining to be processed.

5. Computational Result

To evaluate the proposed heuristic priority rules, a computer program is developed using Microsoft Access and Visual Basic Application and designed to run all the priority rules for the instance cases for the considered scenarios.

Figure 1 exhibits the flow chart upon which the designed computer program is built for solving the job shop scheduling problem. In this paper heuristic and proposed rules are used to schedule jobs on machines seeking for the best result. A comparison between the results of the considered scenarios to see the influence of weight on the scheduling process is implemented. The problem instances are of size J×M, where J is the number of jobs (5) and M is the number of machines (5). The experimental results are tested on the set data set consist of 70 problems. Table 3 lists the abbreviations used in the analysis.

The following is the comparison based on makespan between common heuristic rules and proposed heuristic with randomization job weight (case1) and job weight depend on the relative importance of the job time to all jobs times (case2) as shown in tables 4 and 5 respectively.
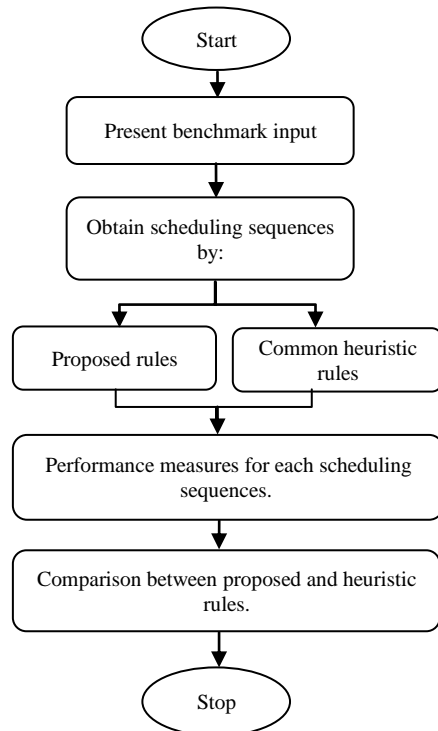

Fig 1 Flowchart for solving JSSP

Table 3 Used abbreviations in the analysis

Abbreviation	Description
C_i	completion time of job j.
A_{Ej}	Average number of Early job.
A_{Tj}	Average number tardy job.
AN_{Ej}	Average number of best solution for early job.
AN_{Tj}	Average number of best solution for tardy job.
A	Absolute average value for each evaluated parameters.

The first column shows number of problem. The minimum makespan obtained by both the heuristic and proposed rules is listed in the second column. Third and fourth columns present the percentage of achievement of minimum makespan for both the heuristic and proposed rules respectively.

Table4.The percentage value of makespan forheuristic and proposed rules for first scenario.

P	C_i	HR	PR	P	C_i	HR	PR
1	31	17	33	36	30	-	22
2	44	5	-	37	35	-	44
3	23	11	66	38	23	11	22
4	34	6	-	39	19	17	11
5	37	6	-	40	19	17	11
6	36	11	-	41	26	11	11
7	45	11	-	42	24	17	11
8	19	6	-	43	27	11	6
9	35	22	-	44	31	6	22
10	40	11	22	45	25	17	-
11	35	11	-	46	31	-	11
12	29	6	-	47	39	17	11
13	29	11	22	48	38	6	11
14	33	17	33	49	37	6	-
15	41	6	-	50	36	-	-
16	41	6	22	51	79	11	11
17	26	6	-	52	56	22	-
18	33	33	22	53	31	44	44
19	22	6	55	54	28	6	33
20	32	22	44	55	15	11	11
21	28	6	-	56	29	6	-
22	44	-	11	57	31	22	11
23	26	17	22	58	21	11	-
24	36	22	11	59	25	-	22
25	28	6	11	60	29	6	22
26	43	6	-	61	26	6	-
27	37	-	22	62	23	17	22
28	38	6	33	63	31	11	-
29	63	6	22	64	26	17	-
30	14	61	-	65	42	17	11
31	38	6	-	66	24	11	33
32	37	6	-	67	27	-	-
33	25	17	-	68	46	22	-
34	23	6	22	69	47	6	11
35	22	-	22	70	47	6	11

It is easily noted that proposed rules performs better relatively to heuristic rules, since the average of achievement percentage value of makespan for the proposed rules is 13 while it is

4

11 for heuristic rules for the first scenario while it is almost equal in the second scenario.

Table 5. The percentage value makespan betweenheuristic and proposed rules for second scenario.

P	C_i	HR	PR	P	C_i	HR	PR
1	31	17	11	36	30	11	-
2	44	6	-	37	36	6	22
3	26	6	11	38	23	6	11
4	33	-	11	39	19	22	33
5	37	6	-	40	19	6	-
6	36	11	-	41	26	17	-
7	43	6	-	42	24	17	22
8	19	6	11	43	27	11	33
9	35	28	22	44	30	6	-
10	40	22	22	45	25	22	11
11	35	6	22	46	26	6	11
12	28	6	-	47	39	17	11
13	29	11	-	48	38	6	-
14	32	6	-	49	37	6	-
15	41	6	-	50	36	6	-
16	41	11	11	51	79	6	11
17	26	6	-	52	56	17	44
18	33	33	33	53	31	55	33
19	22	6	11	54	28	11	-
20	32	28	33	55	15	11	-
21	28	6	-	56	29	6	-
22	43	-	11	57	31	11	11
23	26	6	-	58	21	17	22
24	36	28	22	59	26	17	22
25	27	6	11	60	28	6	-
26	43	6	11	61	29	6	11
27	41	50	78	62	23	22	11
28	42	6	-	63	31	6	-
29	63	6	-	64	26	6	11
30	14	72	44	65	42	6	-
31	38	6	-	66	27	11	-
32	37	6	-	67	28	6	33
33	25	6	-	68	46	22	11
34	23	6	-	69	47	6	22
35	23	6	11	70	47	6	22

Tables 6 and 7 exhibit the number of achievement the minimum makespan for the two scenarios by both the proposed and heuristic rules respectively.

Table 6.. Number of achievement minimum makespan for the two scenarios by proposedrules.

PR	1	2	3	4	5	6	7	8	9
Case 1	6	5	3	6	17	13	3	7	6
Case 2	3	4	3	3	10	22	4	4	17

It is easily noted that proposed rules performs better relatively to heuristic rules, since the average of achievement percentage value of makespan for the proposed rules is 7.9 while it is 7.8 for heuristic rules. Also this fact assured by table 7 where the average achievement value with randomization job weight is 7.2 while it is 8.5 for the other scenario.

Table 7 Number of achievement minimum makespan for the two scenarios by heuristic rules

HR	Case 1	Case 2	HR	Case 1	Case 2
1	2	5	10	3	5
2	14	14	11	9	9
3	6	9	12	15	17
4	2	4	13	6	6
5	12	10	14	3	4
6	13	16	15	24	22
7	2	3	16	6	8
8	7	8	17	3	4
9	6	1	18	6	8

Tables 8 lists the comparison based on number of early job between heuristic rules and proposed rules for first scenario. The average number of early job for proposed rules is 2.42 while it is 2.39for heuristic rules. Table 9 lists the same results for second scenario where the average number of early job is 2.47 for the proposed rules and it is 2.4 for heuristic rules.

Table 8. Difference in the average number of early job between heuristic and proposed rules (case 1).

P	$(A_{Ej})_{HR}$	$(A_{Ej})_{PR}$	P	$(A_{Ej})_{HR}$	$(A_{Ej})_{PR}$
1	3.5	3.56	36	2.67	2
2	1.44	1.44	37	1.61	1.56
3	2.39	3	38	2.67	2.22
4	2.39	1.89	39	3.11	3.56
5	2.56	3	40	3.56	3.67
6	2.89	3	41	2.67	1.89
7	1.78	1.89	42	2.22	1.78
8	1.67	2.44	43	2.89	2.33
9	2.83	2.78	44	2.39	2
10	1.78	2.56	45	1.89	2
11	2.28	2.78	46	2.44	2.33
12	3.56	3.33	47	2.33	2.44
13	2.28	2.56	48	2.06	2.22
14	2.56	2.33	49	1.61	1.89
15	1.72	1.67	50	1.61	1.44
16	2.06	2.11	51	1.44	1.33
17	3.5	3.56	52	1.22	1.33
18	1.78	1.56	53	4.44	3.89
19	4.33	4.67	54	3.67	3.89
20	2.72	2.78	55	4	3.67
21	1.94	1.89	56	2.72	2.78
22	1.89	2.44	57	2.06	1.78
23	3.11	3	58	2.94	3
24	2	2	59	2.67	2.67
25	1.89	2.56	60	2.89	2.22
26	1.5	1.44	61	2	2.44
27	1.89	2	62	2.94	2.33
28	2.28	2.33	63	2.72	2.89
29	1.17	1.33	64	2.5	2.67
30	4.94	4.89	65	1.89	2.11
31	2.06	2.22	66	2.39	2.33
32	1.61	1.89	67	2.22	2.33
33	1.56	1.56	68	1.67	1.89
34	0.83	0.56	69	1.17	1.67
35	1.5	2.33	70	3.56	3.67

Table 9. The average number of early job between heuristic and proposed rules for second scenario

P	$(A_{Ej})_{HR}$	$(A_{Ej})_{PR}$	P	$(A_{Ej})_{HR}$	$(A_{Ej})_{PR}$
1	3.56	3.78	36	2.78	2.11
2	1.61	1.11	37	1.61	1.89
3	2.22	2.11	38	2.78	2.78
4	2.5	2.56	39	3.17	3
5	2.5	2.67	40	3.44	3.44
6	2.89	3.44	41	2.94	2.56
7	1.72	2	42	2.28	2.33
8	1.67	1.78	43	2.89	3.22
9	2.94	3	44	2.39	2.56
10	1.83	1.89	45	1.89	2
11	2.22	2.56	46	2.44	2.44
12	3.56	3.78	47	2.33	2.22
13	2.33	2.2	48	2.05	2.67
14	2.5	2.56	49	1.67	2
15	1.78	2.11	50	1.67	1.67
16	2.17	2.11	51	1.44	1.33
17	3.67	3.67	52	1.17	1.11
18	1.89	1.67	53	4.44	4.33
19	4.39	4.44	54	3.61	3.56
20	2.67	2.56	55	3.83	3.78
21	1.89	1.89	56	2.61	2.67
22	1.83	1.89	57	2.22	1.89
23	3.06	3	58	2.94	2.89
24	2.11	2	59	2.56	3
25	1.89	2.22	60	3	3.22
26	1.61	1.68	61	1.83	1.67
27	1.94	2.22	62	3.22	3.44
28	2.28	2.78	63	2.78	3.11
29	1.11	1.11	64	2.5	2.56
30	4.94	4.89	65	1.83	1.89
31	2.06	2.67	66	2.5	2.56
32	1.67	2	67	2.28	2.44
33	1.56	1.56	68	1.72	1.67
34	1	0.89	69	1.06	1.11
35	1.33	1.56	70	3.44	3.44

Table 10 shows the results of number of tardy jobs. The average number of tardy job for heuristic rules is 2.61 while it is 2.58 for the proposed rules. Table 11 exhibits the same results for second scenario where the average number of tardy job for heuristic rules is 2.60 and it is 2.53 for proposed rules.

Tables 12 and 13 illustrate the average number of early and tardy jobs that achieved for proposed rules and heuristic rules. The third column presents case1 while fifth column presents case 2.

6. Conclusion

In the current paper nine heuristic rules are proposed for scheduling job shop problem and are compared with eighteen of most common and recommended heuristics that are used in job shop scheduling problem. The twenty seven rules are tested on seventy case instances of benchmark problems. The experiments are performed under two scenarios. In the first one randomization job weight is considered while in the second the job weight depends on the relative importance of the job time to all jobs times is considered.

Fifty percent of these instance cases are classical job shop problem and the other are flexible job shop problem. The results clearly show that the developed heuristic rules are very promising and achieved better results than the most common and recommended heuristics used for many different instances. The problem instances are tested through a software program that is designed and built in the current paper and of a structure of a decision support system. This system is capable of running all the twenty seven rules for any problem in the aim of achieving the minimum makespan or the best solution that can be obtained by these embedded 27 rules for the tested or considered problem.

Table 10. Number of tardy jobs for heuristic and proposed rules for first scenario.

P	$(A_{Tj})_{HR}$	$(A_{Tj})_{PR}$	P	$(A_{Tj})_{HR}$	$(A_{Tj})_{PR}$
1	1.5	1.44	36	2.33	3
2	3.56	3.56	37	3.39	3.44
3	2.61	2	38	2.33	2.78
4	2.61	3.11	39	1.89	1.44
5	2.44	2	40	1.44	1.33
6	2.11	2	41	2.33	3.11
7	3.22	3.11	42	2.78	3.22
8	3.33	2.56	43	2.11	2.67
9	2.17	2.22	44	2.61	3
10	3.22	2.44	45	3.11	3
11	2.72	2.22	46	2.56	2.67
12	1.44	1.67	47	2.67	2.56
13	2.72	2.44	48	2.94	2.78
14	2.44	2.67	49	3.39	3.11
15	3.28	3.33	50	3.39	3.56
16	2.94	2.89	51	3.56	3.67
17	1.5	1.44	52	3.78	3.67
18	3.22	3.44	53	0.56	1.11
19	0.67	0.33	54	1.33	1.11
20	2.28	2.22	55	1	1.33
21	3.06	3.11	56	2.28	2.22
22	3.11	2.56	57	2.94	3.22
23	1.89	2	58	2.06	2
24	3	3	59	2.33	2.33
25	3.11	2.44	60	2.11	2.78
26	3.5	3.56	61	3	2.56
27	3.11	3	62	2.06	2.67
28	2.72	2.67	63	2.28	2.11
29	3.83	3.67	64	2.5	2.33
30	0.06	0.11	65	3.11	2.89
31	2.94	2.78	66	2.61	2.67
32	3.39	3.11	67	2.78	2.67
33	3.44	3.44	68	3.33	3.11
34	4.17	4.44	69	3.83	3.33
35	3.5	2.67	70	1.44	1.33

Table 11. Number of tardy jobs for heuristic and
proposed rules for second scenario.

P	$(A_{Tj})_{HR}$	$(A_{Tj})_{PR}$	P	$(A_{Tj})_{HR}$	$(A_{Tj})_{PR}$
1	1.44	1.22	36	2.22	2.89
2	3.39	3.89	37	3.39	3.11
3	2.78	2.89	38	2.22	2.22
4	2.5	2.44	39	1.83	2
5	2.5	2.33	40	1.56	1.56
6	2.11	1.56	41	2.06	2.44
7	3.28	3	42	2.72	2.67
8	3.33	3.22	43	2.11	1.78
9	2.06	2	44	2.61	2.44
10	3.17	3.11	45	3.11	3
11	2.78	2.44	46	2.56	2.56
12	1.44	1.22	47	2.67	2.78
13	2.68	2.79	48	2.94	2.33
14	2.5	2.44	49	3.33	3
15	3.22	2.89	50	3.33	3.33
16	2.83	2.89	51	3.56	3.67
17	1.33	1.33	52	3.83	3.89
18	3.11	3.33	53	0.56	0.67
19	0.61	0.56	54	1.39	1.44
20	2.33	2.44	55	1.17	1.22
21	3.11	3.11	56	2.37	2.33
22	3.17	3.11	57	2.78	3.11
23	1.94	2	58	2.06	2.11
24	2.89	3	59	2.44	2
25	3.11	2.78	60	2	1.78
26	3.39	3.33	61	3.17	3.33
27	3.06	2.78	62	1.78	1.56
28	2.72	2.22	63	2.22	1.89
29	3.89	3.89	64	2.5	2.44
30	0.06	0.11	65	3.17	3.11
31	2.94	2.33	66	2.5	2.44
32	3.33	3	67	2.72	2.56
33	3.44	3.44	68	3.28	3.33
34	4	4.11	69	3.94	3.89
35	3.67	3.44	70	1.56	1.57

Table 12. Averagenumber of best solution of
proposed rules for early and tardy jobs.

P	AN_{Ej}	AN_{EWj}	AN_{Tj}	AN_{TWj}
1	2.6	2.7	2.4	2.3
2	2.4	2.5	2.6	2.5
3	2.3	2.7	2.7	2.3
4	2.6	2.7	2.4	2.3
5	2.2	2.1	2.8	2.9
6	2.2	2.1	2.9	2.9
7	2.6	2.7	2.4	2.3
8	2.6	2.5	2.4	2.5
9	2.2	2.2	2.6	2.8
A	2.42	2.47	2.56	2.5

Table 13. Average number of best solution
of heuristic rules for early and tardy jobs.

HR	AN_{Ej}	AN_{EWj}	AN_{Tj}	AN_{TWj}
1	2.7	2.7	2.3	2.3
2	2.2	2.2	2.8	2.8
3	2.3	2.3	2.7	2.7
4	2.7	2.7	2.3	2.3
5	2.5	2.5	2.5	2.5
6	2	2	3	3
7	2.6	2.6	2.4	2.4
8	2.4	2.3	2.6	2.7
9	2.2	2.4	2.8	2.6
10	2.2	2.4	2.8	2.6
11	2.4	2.3	2.6	2.7
12	2.1	2.1	2.9	2.9
13	2.7	2.7	2.3	2.3
14	2.6	2.6	2.4	2.4
15	2.1	2.1	2.9	2.9
16	2.3	2.3	2.7	2.7
17	2.5	2.5	2.5	2.5
18	2.5	2.5	2.5	2.5
A	2.39	2.40	2.61	2.60

7. References

[1] Dumitrescu , D. , Catlin , S ., Ruxandra ,S., "Genetic Chromo-Dynamics for The Job Shop Scheduling Problem " , Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, pp. 153-160, KEPT2007.

[2]Tamilarasi ,A., kumar, T., " An Enhanced Genetic Algorithm with Simulated Annealing for Job-Shop Scheduling" , International Journal of Engineering, Science and Technology Vol. 2, No. 1, pp. 144-151 , 2010.

[3] Rafinejad, S., Ramtin , F., Arabani , A., " A New Approach to Generate Rulesin Genetic Algorithm Solution to a Job Shop Schedule by Fuzzy Clustering", Proceedings of the World Congress on Engineering and Computer Science Vol. II, 2009.

[4] Jain, A., Meeran, S., "DeterministicJob-Shop Scheduling: Past, Present and Future", European Journalof Operational Research, 113 (2), pp. 390−434, 1999.

[5]Ho, A., Tay, J., "Evolving Dispatching Rules for Solving the Flexible Job- Shop Problem",IEEE Congress on Evolutionary Computation, pp. 245-276, 2005.

[6] Pinedo, M., "Planning and Scheduling in Manufacturing and Services", 1st Ed. Springer, New York, 2006.

[7]Lin, S., Goodman, E., Punch, W., "A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems", Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 481−488, 2003.

[8] Omar, M., Baharum, A., AbuHasan, Y.,"Ajob – Shop Scheduling Problem (JSSP) Using Genetic Algorithm -SHOP", Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, statistics and applications University Sains Malaysia pp.13-16,2006.

[9] Kuczapski, M., Micea1 , V., Maniu , A., Cretu , A., " Efficient Generation Of Near Optimal Initial Population to Enhance Genetic Algorithm for Job-Shop Scheduling", ISSN 1392 – 124X information technology and control, Vol.39, No.1, pp. 32- 37, 2010.

[10]Huiyuan, R.., Lili ,J., Xiaoying , X. , Muzhi, L., " Heuristic Optimization for Dual-Resource Constrained Job Shop Scheduling", International Asia Conference on Informatics in Control, Automation and Robotics, pp.485-488, 2009.

[11] Moghaddas , R., and Houshmand , M., " Job-Shop Scheduling Problem with Sequence Dependent Setup Times " , Proceedings of the International Multi Conference of Engineers and Computer Scientists 19-21 March, Vol., II, Hong Kong, 2008.