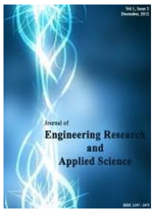
Journal of Engineering Research and Applied Science
Available at www.journaleras.com
Volume 10 (2), December 2021, pp 1906-1918
ISSN 2147-3471 © 2021



An efficient priority rule for flexible job shop scheduling problem

Y. Demir[1,a], H. Yılmaz[2]

[1]Bursa Technical University, Industrial Engineering, Bursa, Turkey.
[2]Bayburt University, Industrial Engineering Engineering, Bayburt, Turkey.

Accepted 3 November 2021

Abstract

In the classical planning approach, the production plan is made by a central planning unit and the production is expected to be made in accordance with the plan. But in real life; the existence of many dynamic factors such as machine failures, new order arrivals, order cancellations or changes cause the plans to be partially or completely incomplete. This reduces the confidence of the enterprises in the production planning function and even causes it to be perceived as an unnecessary activity. The increase in internet speed and developments in technologies for gathering information from the shop floor has provided the opportunity to closely follow the instant changes and to give the reaction in the most accurate way. These advances have led researchers studying in the field of operational research to agent-based approaches or dynamic sequencing rules. In this study, an effective composite priority rule has been developed for Cmax minimization of flexible job shop scheduling problem (FJSP). The composite rule, which is called the relativity rule, is compared with various combinations of priority rules, which are well known in the literature. The results show that the developed composite rule provides a clear dominance over other priority rules.

Keywords: Flexible job shop scheduling problem, priority rules, simulation experiments.

1. Introduction

Many problems encountered in real life are ill-defined due to lack of data. In order to solve such problems, researchers first use a number of assumptions to make the problem well defined [1]. However, this situation makes concessions from the structure of the problem and takes away it from its real structure. Today, the use of advanced automatic identification and data acquisition technologies in the production environment has enabled real-time data collection from machines. This has led to improved solutions and response quickly to the dynamic changes in the shop environment by incorporating the previous assumptions into the problem. However, classical heuristic techniques fail to respond quickly to dynamic systems due to problem-specific structures and long computational time requirements [2]. In this respect, the rules named in the literature in different ways such as priority rules [3], dispatching rules [4, 5] machine scheduling rules [6] decision rules [2], heuristic rules [7, 8], list scheduling approaches [9] in the literature[1] has become a frequently used method for solving scheduling problems due to its simple structure, low computational effort, low domain knowledge requirements [10]. Many studies have been carried out on the priority rules which of foundation dating back to the end of 60s [11] and their performances have been evaluated with simulation based experimental studies in various production environments [12-14]. There is no priority rule that outperforms others in general, since their performance varies according to the scheduling environment in which priority rules are applied and the goals that are tried to be optimized [6, 15].

Various classifications have been made on the priority rules in the literature. Priority rules are generally classified as simple and composite according to their structure. Rajendran and Holthaus [13] classified priority rules according to the information used as input (arrival time, processing time, delivery date etc.) in the basic sense. Ramasesh [15] provides comprehensive classification of priority rules. The author classified dynamic and static priority rules, according to whether the priority value of the work changes according to the flow of the work throughout the workshop. In addition, Ramasesh [15] classified the priority rules as state-dependent if the priority value changes depending on

[1] Priority rule was preferred during this study.

[a] Corresponding author; yunus.demir@btu.edu.tr

the current state of the workshop (workloads on machines, queue lengths, etc.), otherwise the state-independent. Vepsalainen [16] classified the information that can be used in a state-dependent rule according to the forecasting horizon to observed status, anticipated status, and feedback from anticipated performance. Land [17] categorized the priority rules according to their logistic performance. Baker [18] divided into two groups as local and global according to the scope of information required for priority rules. In local rules priority index is determined based on information about jobs represented in the individual machine queue. The Shortest processing time (SPT) and most operation remaining (MOR) are two examples of local rules [19]**.** By contrast, global rules are used to dispatch jobs using all information available on the shop floor [20]. Composite priority rule (CPR) developed in this study can be given as an example of global rule. Mei et. al. [10] divided the priority rules into two as non-delay and active according to the decision point. In non-delay rules, it is necessary to decide job to process immediately after the machine is became idle and machines are not allowed to remain idle. A comprehensive literature review of priority rules is given by Panwalkar and Iskander [21], Haupt [3], Blackstone et al. [22], Horng [23].

There are two ways in the literature to use priority rules in the flexible job shop scheduling (FJS) environment. In the first, priority rules are used in various operators of heuristic techniques. Brandimarte [24], Scrich et al. [25], Alvares-Valdes et al. [26], Pezzella et al. [27], Na and Park [9] used them to create the initial solution. Baykasoğlu [28] used it in encoding operator of his proposed Linguistic-based simulated algorithm.

The second part, which uses priority rules in FJS environment, is based on algorithms developed to generate automatic priority rules. Zhang et al. [29] have tried to establish a priority rule that minimizes total energy consumption, unlike the classical priority rules. Researchers have developed a gene expression algorithm for the automatic generation of this priority rule. Zhou et al. [2] developed a genetic programming-based approach that automatically generates a priority rule for multi-objective dynamic flexible job shop scheduling problem (FJSP). The researchers created a total of 320 combination test sets consist of 10 priority rules for machine assignment and 30 priority rules for job sequencing, and presented the performance of the priority rules they developed in a comparative manner.

The first of the two closest paper to this study was done by Tay and Ho [30]. Researchers have developed a CPR with genetic programming for multi-objective FJSP and tested it separately for each objective. The second study deals with a different version of FJSP where each job is an alternative operation. In related study 36 priority rule combinations of three machine selection rules and 12 job sequencing rules are tested [31].

In this study, an effective CPR, called relativity rule, is developed for C_{max} minimization in a flexible job shop-scheduling environment. In this study, a simulation experiment was developed in order to compare the various priority rules known in the literature in terms of C_{max} criteria. The comparative results obtained at the end of the experiment showed that the developed rule gives a clear supremacy over other priority rules. A second comparison was made to clarify the performance of the rule developed in the study. This time, the results obtained by the relativity rule are compared with the results obtained from the complex meta-heuristic algorithms (search-based) developed for this problem. As a result of the comparison, it was seen that the results obtained with the composite rule developed were very close (superior to some) to the results obtained from the algorithms of complex structure [32-38].

Ongoing parts of the study are organized as follows: In the second part, the problem handled (FJSP) is introduced. In the third section, the simple priority rules used in the comparison and the CPR developed are introduced. In the fourth section, the experimental conditions are explained. In Section 5, developed Arena simulation model was introduced. In the sixth chapter, the results of the simulation experiment are presented in a comparative manner. In the seventh chapter, a general evaluation is made [39-46].

2.Definition of problem

In this section, the notations used in the definition of the problem and the priority rules are explained and then FJSP is introduced [47-54].

2.1. Flexible job shop scheduling problem

FJSP, which is a specialized version of the job scheduling problem, can be described as follows:

- There are n jobs J={J_1, J_2, …, J_n} waiting to be processed on the m machine M={M_1, M_2, …, M_m}.
- Each job j consist of n_j consecutive {O_{j1}, O_{j2}, …, O_{jnj}} operations.
- Every O_{jk} operation can be processed in one of machine among the M_{jk} machine set ($M_{jk} \subseteq M$).
- FJSP is the problem of determining which machine will perform the operation in which order, in order to optimize one or more criteria. In this study, makespan (C_{max}) is tried to be minimized.

FJSP can contain various assumptions. The assumptions valid in this study are as follows:

- The setup times for the machines are neglected and the operations are processed without interruption on the machine to which they are assigned.
- Jobs are independent of each other.
- All machines are always available. It is assumed that there will be no downtime due to break down and maintenance.
- All jobs are ready to be processed time 0.

3.Priority rules

Since simulation experiments based on priority rules in the literature are generally conducted for flow shop and job shop scheduling environments, it is focused on priority rules for job sequencing. Albeit rare, there are a few studies that include the priority rules developed for machine selection [23, 7]. Since FJSP consists of sub-problems of machine assignment and job sequencing, this section is discussed under two sub-topics [55-65].

3.1. Job sequencing rules

The job sequencing rules, which can be defined as the priority function in its simplest form, determine which job will be selected among the jobs waiting in the queue of the machine when it has become idle during scheduling horizon. With this function priority value is created for each pending job and job with the best (Min / Max) priority value is selected [10]. Z_{ijk} refers to the priority index, which is defined differently in each rule [23]. All of the following rules are based the first processing of operation with the minimum Z_{ijk} value.

Service in Random Order (SIRO): One of the jobs waiting in the queue is randomly selected for processing.

First In First Out (FIFO): According to this rule, job j, which comes to machine i, is processed first. It is seen that different names such as First in queue (FIQ) [32], first come first served (FCFS) [33] and smallest release time (SRT) [20] are used in the literature for this rule, which minimizes the change in waiting periods in the queue.

$$Z_{ijk} = r_{ijk} \quad (1)$$

Shortest Processing Time (SPT): According to this rule, the job with the shortest processing time (p_{ijk}) of the jobs waiting in machine queue is processed first.

$$Z_{ijk} = p_{ijk} \quad (2)$$

Longest Processing Time (LPT): Of the jobs waiting in the queue of machine i, the job with the longest processing time (p_{ijk}) takes precedence. It is used to balance workload in parallel machine scheduling problems.

$$Z_{ijk} = -p_{ijk} \quad (3)$$

Most operation remaining (MOR): The job j with the maximum number of operations (rmn_j) remaining is selected. The aim is to maximize capacity utilization.

$$Z_{ijk} = -rmn_j \quad (4)$$

Least Operations Remaining (LOR): The job j with the minimum number of operations (rmn_j) remaining is selected.

$$Z_{ijk} = rmn_j \quad (5)$$

Most average work remaining (MAWR): The job *j* with the maximum total processing time of the remaining operations is selected (α_{jk}: Set of remaining operations).

$$Z_{ijk} = -\sum_{g\in\alpha_{jk}} \bar{p}_{ig}, \quad \bar{p} = (\sum_{i\in M_{jk}} p_{ijk})/N_{jk} \quad (6)$$

Least average work remaining (LAWKR):

The job j with the minimum total processing time of the remaining operations is selected.

$$Z_{ijk} = \sum_{g\in\alpha_{jk}} \bar{p}_{ig}, \qquad \bar{p} = \Big(\sum_{i\in M_{jk}} p_{ijk}\Big)/N_{jk} \tag{7}$$

3.2.Machine selection rules

These rules are used for routing jobs to machines. M_{jk} and N_{jk} are the set of machines and the number of alternative machines that operation O_{jk} can be performed, respectively. The machine selection rules are as follows:

Random (RND): The machines to which the operation will be assigned have equal priority and are random one is selected [23].

Number in Next Queue (NINQ): The machine with the minimum number of jobs waiting in the queue is selected.

$$Z_i' = NQ_i \tag{8}$$

Length in Next Queue (LINQ): The machine with the smallest total processing time of jobs waiting in the queue is selected (β_i: Set of operations waiting in the queue of machine i). Kaweegitbundit [7] introduced this rule as Work in next queue (WINQ).

$$Z_i' = \sum_{O_{jk}\in\beta_i} p_{ijk} \tag{9}$$

Balanced loads (BL): Jobs are assigned to the machines so that the workloads on the machines are balanced. In other words, the machine with the least work load is selected. In LINQ, the O_{jk} operation is removed from the β_i set after the O_{jk} operation is processed in the machine i and leaves the queue. However, processing times of all operations assigned to machine i are stored in BL.

$$Z_i' = \sum_{O_{jk}\in\gamma_i} p_{ijk} \tag{10}$$

3.3. Proposed composite priority rule: Relativity Rule (RR)

The FJSP consists of two sub-problems: assigning jobs to machines and sequencing them on the machines to which they are assigned. C_{max} minimization is closely related to the solution of both sub-problems (together or separately), meeting the following conditions:

- Balancing machine workloads
- Minimizing machine idle times
- Choosing the fastest alternative machine

Let's assume that the above-mentioned conditions are evaluated separately while scheduling. In the first case, if the jobs are assigned to the machines only according to the fastest machine criteria, it causes an unbalanced workload distribution as shown in Figure 1, an increase in idle time of the machines and thus an increase in C_{max} value.

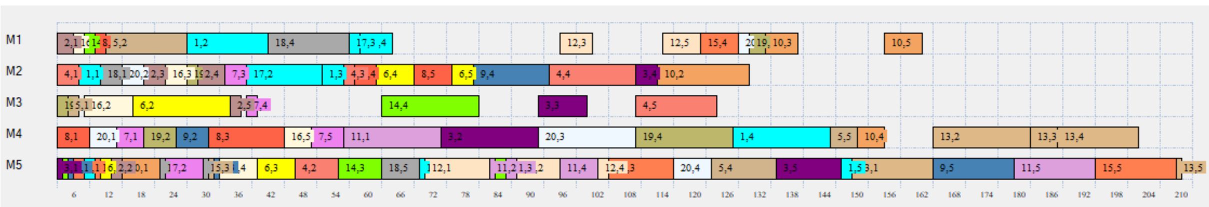


Figure 1. Selection of the fastest machine according to process times. (C_{max}: 210, MK07 test problem)

Or, if scheduling is made only by considering the balancing of machine workloads, it may cause the selection of slower machines and increase the C_{max} value, as shown in Figure 2.

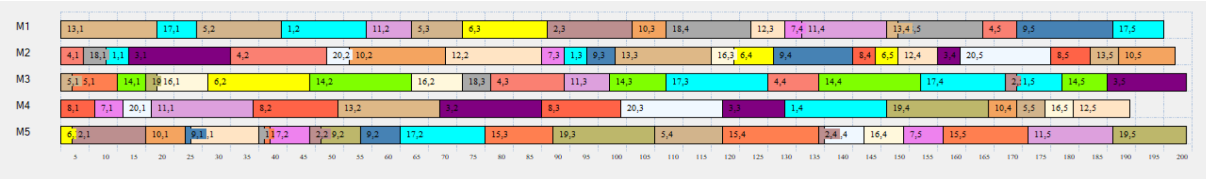


Figure 2. Machine selection according to balanced workload distribution (Cmax: 200, MK07 test problem)

Hence in FJSP, the aforementioned elements should be jointly considered in the solution method to be developed for C_{max} minimization. The composite machine selection rule proposed in this study has been developed to take into account all of the specified criteria. The rule called Relativity Rule (RR) consists of the weighted sum of the three conditions described above (Eq. 15):

1. Relative machine availability time (RMAT)
   Machine availability time is the sum of the remaining processing time (*RPT*) of the job being processed on the machine and the processing time of the jobs waiting in the queue of the respective machine. This criterion prevents the machines from being idle. A balanced workload is also provided to the machines
   $[(LINQ_i + RPT_i)/\sum_i (LINQ_i + RPT_i)]$ (11)
2. Relative machine processing speed (RMPS)
   With this criterion, faster machines are prioritized.
   $p_{ijk}/\sum_{i \in M_{jk}} p_{ijk}$ (12)
3. Relative expected machine workload (REMW)

It provides a more global perspective than the first criterion. With this rule, it is tried to identify the machines that may be bottleneck for future operations. When calculating this parameter, whose value depends on time, the expected processing time (p'_{ijk}) is calculated first. Expected processing time is found by dividing the processing time of the relevant operation by the number of alternative machines (N_{jk}) where the operation can be performed (Eq. 13). Then, expected workload for the relevant machine i is calculated by summing the expected processing times for the operations ($O_{jk} \in K'$) that can be processed on each machine i and have not started to be processed yet at time t. This value found for each machine is then proportioned to the total expected machine workload for all machines, and a parameter value in the range of 0-1 is obtained (Eq. 14).

$$p'_{ijk} = \frac{p_{ijk}}{N_{jk}} \quad (13)$$

$$\sum_j \sum_{k \in K'} p'_{ijk} / \sum_i (\sum_j \sum_{k \in K'} p'_{ijk}) \quad (14)$$

The composite priority rule developed in this study consists of the weighted sum of the three conditions mentioned above and the candidate operation is assigned to the machine with the smallest RR value.

$$RR_i = \delta * RMAT + \theta * RMPS + \lambda * REMW \quad (15)$$

The coefficients δ θ and λ guide the machine selection decision in case of conflict. For example, suppose there are two alternative machines for the O_{jk} operation. The first machine can process the relevant operation in 5 minutes, while the other can process in 8 minutes. Suppose there are operations with a total time of 80 minutes waiting to be processed in the queue of the first machine, while there are no operations waiting to be processed on the second machine. In this case, should the faster first machine or the second machine with less queue length be preferred? Another question is what is the breakeven point for these two conflicting situations?

Another example; Let's consider an operation that can be processed on four machines. Let the queue lengths in minutes on these machines: 70, 60, 30, 0 and process times 8, 6, 5 and 4, respectively. So, there is no operation waiting to be processed on the fourth machine and it is the fastest machine. Choosing the fourth machine, which is idle and faster, may be the first choice that comes to mind. However, if the fourth machine is a bottleneck machine (that is, if it is the only alternative for remaining operations or one of very few alternatives) for the remaining operations, assigning an operation with many alternatives to this machine with a higher expected workload may cause queues and increase the C_{max} value in the future. In this case, which alternative should be preferred?

Considering these questions, a single machine selection index (RR) was created by weighting the three functions described above for the most appropriate machine selection.

4.Experimental conditions

Priority rules are rather a preferred method for dynamic scheduling problems. However, in order to test the composite sorting rule (RR) developed in this study, the test problems in static structure, which are frequently used in the literature, were used.
First, simulation models were created for MFJS7 and MFJS10 test problems created by Fattahi et al. [34] and MK05 and MK07 test problems created by Brandimarte [24] to determine δ, θ, λ weight values.

For the simulation experiment, MFJS7 test problem (8 jobs, 7 machine) created by Fattahi et al. [34] and MK07 test problem (20 jobs, 5 machine) created by Brandimarte [24] was used. Parameter optimization has been done with the optquest optimization tool. Each pattern was repeated until 95% confidence level was achieved. At the time of zero, all jobs are ready to be processed. However, as the number of machines is less than the number of jobs, jobs begin to be processed according to the previously determined random order. Pseudo code of schedule generation scheme (SGS) is presented below.

```
Create random job order to get started
while there are unprocessed operations do
Wait for a machine with pending operations;
Calculate priorities of all operations with time RR formulation;
Schedule the smallest RR value operation;
Update machine and job's next operation ready time;
end while
```

The second simulation experiment was carried out to determine with which job sequencing rule the RR rule (designed for machine selection) gives effective results. Accordingly, the performances of well-known simple priority rule pairs (machine selection-job sequencing) and the proposed composite priority rule (RR) in the flexible job shop scheduling environment were evaluated. The priority rule pairs were first tested on the MK07 test problem created by Brandimarte [24]. The results obtained for a total of six performance criteria, especially the Cmax performance criteria, are presented in Table 1.

Table 1. Simulation test results for MK07 test problem

	Cmax		Entity Stat.		Queue Stat		Resrc Stst.
	AVG	**MIN**	**AWT**	**WIP**	**ANW**	**AWTQ**	**AMU**
BL-FIFO	212	199	134,000	0,876	2,537	27,100	0,969
BL-SPT	223	201	106,030	0,706	1,910	21,655	0,913
BL-LPT	224	210	110,940	0,721	1,985	22,674	0,900
BL-MOR	209	196	135,140	0,887	2,585	27,193	0,965
BL-LOR	232	207	90,170	0,604	1,556	18,542	0,859
BL-MAWR	213	197	139,520	0,898	2,625	28,248	0,967
BL-LAWR	232	212	87,700	0,599	1,520	17,767	0,874
NINQ-FIFO	214	204	125,530	0,820	2,348	25,948	0,931
NINQ-SPT	236	219	104,180	0,654	1,772	21,446	0,843
NINQ-LPT	224	200	107,220	0,700	1,922	22,382	0,879
NINQ-MOR	209	196	135,140	0,887	2,585	17,193	0,965
NINQ-LOR	232	207	90,170	0,604	1,558	18,542	0,859
NINQ-MAWR	210	192	129,350	0,848	2,462	26,948	0,929
NINQ-LAWR	232	212	87,700	0,599	1,520	17,770	0,874
LINQ-FIFO	206	196	125,370	0,848	2,431	25,989	0,961
LINQ-SPT	211	198	96,140	0,681	1,820	20,352	0,906
LINQ-LPT	225	205	104,600	13,658	1,856	22,408	0,876
LINQ-MOR	212	198	136,420	0,886	2,579	27,577	0,965
LINQ-LOR	222	204	90,921	0,647	1,674	19,082	0,914
LINQ-MAWR	211	196	138,900	0,900	2,632	28,206	0,968
LINQ-LAWR	234	202	91,020	0,605	1,559	18,700	0,860
RR-FIFO	156	146	93,950	0,838	2,410	18,210	0,943
RR -SPT	162	152	73,595	0,684	1,827	73,575	0,910

RR -LPT	178	163	88,220	0,714	1,992	18,356	0,863
RR -MOR	155	146	96,415	0,856	2,487	18,636	0,939
RR -LOR	180	161	68,137	0,606	1,560	13,395	0,865
RR -MAWR	157	153	101,665	0,882	2,587	19,671	0,940
RR -LAWR	187	169	68,410	0,575	1,473	12,765	0,827

AWT: Average waiting time, WIP: Work in process, ANW: Average number of waiting, AWTQ: average waiting time in queue, AMU: Average machine utilization

For all rule pairs, it was observed that LOR and LAWR job sequencing rules had a positive effect on performance criteria related to flow time. However, it has been observed that it has a negative effect on machine utilization rate. The LPT job sequencing rule was found to be poor in terms of both flow time and machine utilization.

In order to examine the effect of the priority rules on the C_{max} performance criterion, the results obtained in Table 1 were arranged and Table 2 and Table 3 were formed. Table 2 and Table 3 show the average C_{max} values for machine selection and job sequencing rules, respectively.

Table 2. Average C_{max} values for machine selection rules

	C_{max}	
	AVG[2]	**MIN**
BL	221	203
NINQ	222	204
LINQ	217	200
RR	168	146

Table 3. Average C_{max} values for job sequencing rules

	C_{max}	
	AVG[3]	**MIN**
FIFO	197	186
SPT	208	193
LPT	213	195
MOR	196	184
LOR	217	195
MAWR	198	185
LAWR	221	199

[2] Average values for seven job sequencing rules
[3] Average values for four machine selection rules

When both tables were evaluated together, it was seen that job selection rules were more effective on C_{max} value than machine selection rules (except for developed composite priority rule-RR) in terms of simple priority rules. Job sequencing rules FIFO and MOR and MAWR have been observed to have a significant contribution on C_{max} value (Table 3). Table 2 shows that the composite rule developed for machine selection has a considerable effect on Cmax. When the machine selection and job sequencing rules were evaluated together, the RR-FIFO and RR-MOR rule pairs reached 146 C_{max} value. This result is very close to the average result (143) obtained in the literature so far from the studies conducted for Cmax minimization in FJS environment.

5.Development of proposed simulation model

This section discusses the ARENA simulation model established to see the effect of different priority rules on the Cmax performance criterion in a FJS environment. The sample simulation model presented in this section was formed for the MFJS7 test problem with 8 jobs and 7 machines created by Fattahi et al. [344]. Proposed simulation model consists of three parts:

Part-I: This part consists of two sub-components as shown in Figure 3. In the first subcomponent, jobs are created and their properties are assigned. Also a random priority coefficient is assigned to each job in this subcomponent (Att J.). After all the jobs are created, they are kept waiting for the selection of first operation. In the second subcomponent job with the highest priority among jobs held selected and sent to the department where the machine assignment is made for the first operation. In this way, a variety of solutions is provided for the simulation model with deterministic input parameter.

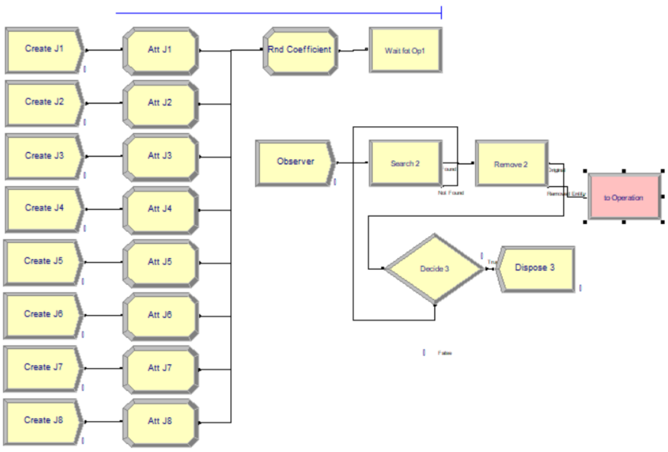


Figure 3. Arena Model Part I

Part-II: This is the part where machine selection is made. Incoming works are directed to the machines according to the rule defined in the “Decide” module. The “Record” module keeps a record of the machine with which the operation was performed. Then, with the “Assign” module, the processing time of the operation is assigned and directed to the relevant machine.

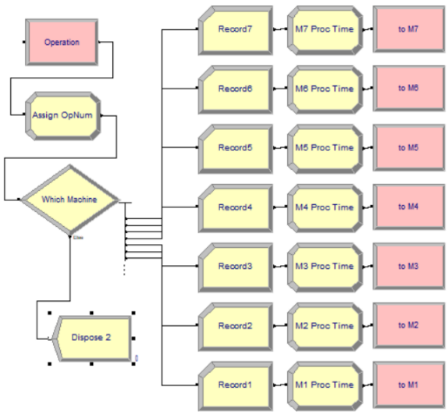


Figure 4. Arena Model Part II

Part-III: In this part according to the job sequencing rule selected, jobs are taken for processing to the machines to which they are assigned. With "Assign" modules, the information to be used in the priority rules are updated and assigned to the jobs during the simulation. The on-going jobs are redirected to the machine selection stage introduced in part II, and the finished jobs are removed from the system. "Record" modules in this part keep the completion times of operations.

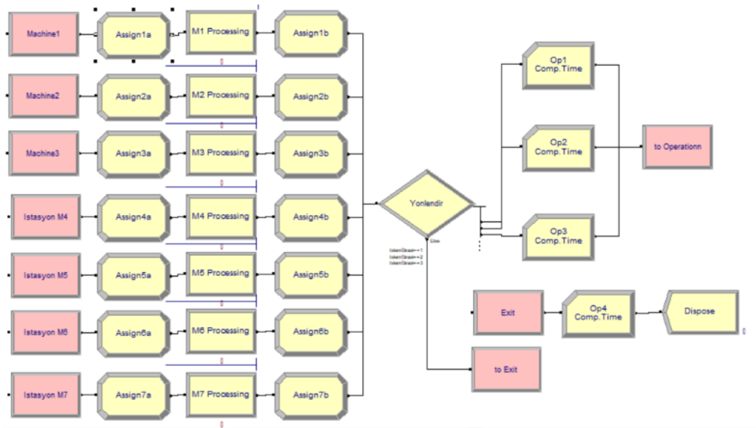


Figure 5. Arena Model Part III

6.Performance of RR

In the experimental conditions section, the performance of the RR was compared with some of the most well-known ranking rules in the literature. Again, in the relevant section, the job sequencing rule that RR works effectively has been determined. In this section the performance of the RR has been compared with various search heuristics suggested in the literature. Medium and large sized test problems created by Fattahi et al. [34] and Brandimarte [24] were used for comparison. Comparative results are presented in the Table 4.

Table 4. Comparative results

Fattahi et al. (2007)	FJSSP Instances	Size	RR-FIFO	Bagheri et al. 2010	Fattahi et al. 2007-1	Fattahi et al. 2007-2	Zandieh et al. 2008	Brandimarte (1993)	FJSSP Instances	Size	RR-FIFO	Xing et al. 2010	Ho et al. 2007	Teekeng and Thammano	Teekeng and Thammano	Habib et al 2012
	MFJS6	8x7	717	625	717	816	634		MK 02	10x6	28	29	29	27	30	28

MFJS7	8x7	1000	879	1020	1048	881	MK04	15x8	65	65	67	64	68	64
MFJS8	9x7	1108	884	1030	1220	891	MK05	15x4	178	173	176	175	181	173
MFJS9	11x8	1112	1088	1105	1124	1094	MK07	20x5	154	144	147	144	159	144
MFJS10	12x8	1382	1267	1384	1546	1286	MK10	20x15	242	229	296	234	272	230

To the left of the table presented in two parts the results obtained with Fattahi et al. [34] test problems and on the right, the results obtained with the test problems of Brandimarte [24] are presented. The code of the problem is presented in the first column and in the second column size of problem is presented. In the third column, the results obtained with the combination of RR (recommended for machine selection) and FIFO (preferred for job sequencing) are presented. As can be seen in the Table 4, acceptable solutions have been reached in terms of solution quality.

Another advantage of the developed RR priority rule is its change-sensitive structure. According to the instant information such as queue length, remaining processing time, or failure status collected from the shop floor, the production can be directed to minimize the Cmax value (increase the machine utilization rate) by changing the RR value. For example, in the case of failure of machine i, the p_{ijk} value is determined as a very large number for all operation O_{jk} (eq.12). In this way, it is prevented from assigning a job to the malfunctioning machine among the alternatives and is dynamically directed to the most suitable alternatives.

7.Conclusions

In this study, an effective composite priority rule called the relativity rule for C_{max} minimization in the flexible job shop scheduling environment is developed. In the study, a simulation experiment was created in which the developed rule and the most known priority rules in the literature were compared in terms of C_{max} criteria. For this purpose, six different machine selection rules and 8 different job sequencing rules were used. The comparative results obtained at the end of the experiment showed that the developed rule provides a clear advantage over other priority rules. The RR rule has also been compared with complex search heuristics. In terms of solution quality, RR yielded acceptable results even if it did not pass all of these algorithms. With advances in internet speed and advancing data collection technology, classical central planning leaves its place to adaptive on-site planning. In this context, the dynamic structure RR, which changes according to the instant data, has the opportunity to apply in real life. For example, by being embedded in MES software, it can provide a decentralized production optimization that is compatible with changing conditions. And furthermore, optimized route information can be projected on screens on forklifts performing material handling between workstations / machines.

As future work, higher quality solutions can be obtained by estimating the parameters (δ, θ, λ) expressing the trade-off against contradictory situations with machine learning techniques according to the problem characteristics (number of jobs, number of machines, level of flexibility ...). In other words, instead of fixed values, changing values can be used for each problem. In addition, the RR value was found by the weighted scalar sum of the specified contradictory conditions. By analyzing the RR function mathematically, functions that will lead to better quality solutions can be defined.

Notations

i: Machine index
j: Job index
k: operation index
O_{jk}: k. operation of j. job
M_{jk}: Machine set for processing O_{jk}.
N_{jk}: Number of alternative machines for processing O_{jk}.
r_{ijk}: Arrival time of O_{jk} to machine i.
p_{ijk}: Processing time of operation O_{jk} on machine i
rmn_j: Number of remaining operations of job j after operation O_{jk} is completed.

Z_{ijk}: Priority value of O_{jk} operation in machine i queue
β_i: Set of operations waiting in queue for machine i.
γ_i: Operation set assigned to machine i

References

[1] Fortus, D. The Importance of Learning to Make Assumptions. Wiley Periodicals, Inc. Sci Ed 2009; 93(1): 86–108.
[2] Zhou, Y., Yang, J-J, Zheng, L-Y., 2018. Hyper-Heuristic Coevolution of Machine Assignment and Job Sequencing Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling, IEEE Access, vol. 7, pp. 68-88, 2019. doi: 10.1109/ACCESS.2018.2883802
[3] Haupt, R., 1989, "A Survey of Priority Rule-Based Scheduling," OR Spektrum, 11, 3-16.
[4] Branke J., Hildebrandt, T., Scholz-Reiter, B. Hyper-heuristic Evolution of Dispatching Rules: A Comparison of Rule Representations. Evolutionary Computation, 2015; 23(2), 249-277.
[5] Xiong, H., Fan, H., Jiang G., Li, G. A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. European Journal of Operational Research. 2017; 257: 13-24.
[6] Sabuncuoglu, I. A study of scheduling rules of flexible manufacturing systems: A simulation approach. Int. Journal of Production Research 1998; 36(2): 527–546.
[7] Kaweegitbundit, P., Eguchi, T. Job Shop Scheduling with Alternative Machines Using a Genetic Algorithm Incorporating Heuristic Rules-Effectiveness of Due-date Related Information. IFIP Advances in Information and Communication Technology, AICT-459 (Part I), 2016; 439-446.
[8] Dooley KJ., Mahmoodi, F. Identification of Robust Scheduling Heuristics: Application of Taguchi Methods in Simulation Studuies, 1992.
[9] Na, H., Park, J. Multi-level job scheduling in a flexible job shop environment. Int. Journal of Production Research 2014; 52(13): 3877-3887.
[10] Mei, Y., Zhang, M., Nyugen, S. Feature Selection in Evolving Job Shop Dispatching Rules with Genetic Programming. GECCO'16, July 20-24, Denver, Colorado, USA, 2016.
[11] Conway, RW., Maxwell, WL., Miller, LW. Theory of scheduling. Addison-Wesley Publishing Company, Reading, Massachusetts, 1967.
[12] Kaban, AK., Othman, Z., Rohmah, DS. Comparison of dispatching rules in job-shop scheduling problem using simulation: a case study. Int. Journal of Simulation Modeling 2012; 11(3): 129-140.
[13] Rajendran, C., Holthaus, O. A comparative study of dispatching rules in dynamic flow shops and job shops, European Journal of Operational Research, 1999; 116: 156-170.
[14] Jaya, NB. Evaluation of different dispatching rules in computer integrated manufacturing using design of experiment techniques. Ms.Thesis, Faculty of Mechanical and Manufacturing Engineering, University Tun Hussein Onn, Malaysia, 2015.
[15] Mouelhi-Chibani, W., Pierreval, H. Training a neural network to select dispatching rules in real time. Computers & Industrial Engineering, 2010; 58(2): 249–256.
[16] Vepsalainen, A.P.J., 1984. State Dependent Priority Rules for Scheduling. Graduate School of Industrial Administration. The Robotics Institute Carnegie-Mellon University Pittsburgh, Pennsylvania 15213.
[17] Land, MJ. Workload control in job shops, grasping the tap. s.n, 2004.
[18] Baker, K.R. Introduction to sequencing and scheduling. Wiley, New York, 1974.
[19] Trietsch, D., Baker, KR. Principles of Sequencing and Scheduling. Wiley & Sons, Hoboken, New Jersey, 2009.
[20] Alharkan I.M. Algorithms for sequencing and scheduling, Industrial Engineering Department, College of Engineering, King Saud University., Riyadh, Saudi Arabia, 2005.
[21] Panwalkar, SS., Iskander, W. A Survey of Scheduling Rules. Operation Research 1977; 25(1): 45-61.
[22] Blackstone, JH., Phillips, DT., Hogg, GL. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. Int. J. of Production Research, 1982; 20(1): 27-45.
[23] Horng, HC. Comparing Steady-state Performance of Dispatching Rule-pairs in Open Shops. Int. Journal of Applied Science and Engineering 2006; 4(3): 259-273.
[24] Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research 1993; 41: 157-183.
[25] Scrich, CR., Armentano AV., Laguna, M. Tardiness minimization in a flexible job shop A

Tabu Seaarch Approach. Journal of intelligent manufacturing. 2004; 15: 103-115.
[26] Alvarez-Valdes, R., Fuertes, A., Tamarit, J.M., Gimenez, G., Ramos, R. A heuristic to schedule flexible job-shop in a glass factory. European Journal of Operational Research, 2005; 165: 525–534.
[27] Pezzella, F., Morganti, G., Ciaschetti, G.A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & Operations Research 2008; 35: 3202 - 3212.
[28] Baykasoğlu A. Linguistic-based meta-heuristic optimization model for flexible job shop scheduling. International Journal of Production Research, 2002; 40(17): 4523-4543.
[29] Zhang, L., Tang, Q., Wu, Z., Wang, F. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. Energy 2017; 138: 210-227.
[30] Tay, J.C., Ho, N.B. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Computers & Industrial Engineering, 2008; 54: 453–473.
[31] Doh, HH., Yu, JM., Kim, JS., Lee, DH., Nam, SH. A priority scheduling approach for flexible job shops with multiple process plans. International Journal of Production Research, 2013; 51(12), 3748-3764.
[32] Barman, S. Simple priority rule combinations: An approach to improve both flow time and tardiness. International Journal of Production Research, 1997; 35(10): 2857-2870.
[33] Vepsalainen, APJ., Morton, TE. Priority Rules for Job Shops with Weighted Tardiness Costs. Management Science 1987; 33(8): 1035-1047.
[34] Fattahi, P., Mehrebad, MS., Jolai, F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems, J. Intell. Manuf. 2007; 18: 331–342.
[35] Amiri, M., Zandieh, M., Yadani, M., Bagheri, A. A variable neighbourhood search algorithm for the flexible job-shop scheduling problem. International Journal of Production Research 2010; 48(19): 5671-5689.
[36] Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M. An artificial immune algorithm for the flexible job-shop scheduling problem. Future Generation Computer Systems, 2010; 26 (4): 533-541.
[37] Ennigrou, M., Ghédira, K., 2008. New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach. Autonomous Agents and Multi-Agent Systems, 17(2), 270–287.
[38] Gao, L., Peng, C., Chi Zhou, and Li, P. 2006. Solving Flexible Job-shop Scheduling Problem Using General Particle Swarm Optimization. Proceedings of the 36th CIE Conference on Computers & Industrial Engineering.
[39] Gutiérrez, C., García-Magariño, I., 2011. Modular design of a hybrid genetic algorithm for a flexible job–shop scheduling problem. Knowledge-Based Systems, 24, 1, 102-112.
[40] Himida, A.B., Haouari, M., Huguet, M-J., Lopez, P., 2010. Discrepancy search for the flexible job shop scheduling problem. Computers & Operations Research, 37, (12), 2192-2201.
[41] Ho, N.B., and Tay, J.C., 2004. "GENACE: an efficient cultural algorithm for solving the flexible job-shop problem," Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, pp. 1759-1766 Vol.2.
[42] Ho, N.B., Tay, J.C., Lai, E, M.-K. 2007. An effective architecture for learning and evolving flexible job-shop schedules. European Journal of Operational Research, 179(2), 316-333.
[43] Holthaus, O. and Rajendran, C. 1997. Efficient dispatching rules for scheduling in a job shop, International Journal of Production Economics, 48: 87-105.
[44] Ida, K. and Oka, K. 2011. Flexible Job-Shop Scheduling Problem by Genetic Algorithm. Electrical Engineering in Japan,177,(3), 505-511.
[45] Karimi, H., Rahmati, S.H.A., Zandieh, M., 2012. An efficient knowledge-based algorithm for the flexible job shop scheduling problem. Knowledge-Based Systems, 36, 236-244.
[46] Koruca, H.İ., Aydemir, E., 2014. A Priority Rule Based Production Scheduling Module on Faborg-Sim Simulation Tool. Gazi University Journal of Science, 27(4), 1143-1155.
[47] Li, J-Q., Pan, Q-K., Suganthan, P.H., Chua, T.,J., 2011. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. The International Journal of Advanced Manufacturing Technology 52, 5–8, 683–697.
[48] Najid, N.N., Dauzere-Peres, S. and Zaidat, A. 2002. "A modified simulated annealing method for flexible job shop scheduling problem," IEEE International Conference on Systems, Man and Cybernetics, Yasmine Hammamet, Tunisia, pp. 6 pp. vol.5
[49] Ong, Z.X., Tay, J.C., Kwoh, C.K., 2005. Applying the Clonal Selection Principle to Find

Flexible Job-Shop Schedules. International Conference on Artificial Immune Systems ICARIS 2005: Artificial Immune Systems, 442-455.
[50] Raghu, T. S. and Rajendran, C. 1993. An efficient dynamic dispatching rule for scheduling in a job shop, International Journal of Production Economics, 32: 301-313.
[51] Rahmati, S.H.A., Zandieh, M., 2012. A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem. The International Journal of Advanced Manufacturing Technology, 58, (9–12), 1115–1129.
[52] Ramasesh, R., 1990. Dynamic Job Shop Scheduling: A Survey of Simulation Research. OMEGA, 18, 43-57
[53] Sargent, R.G., 2010. Verification and Validation of Simulation Models, Proceedings of the 2010 IEEE Winter Simulation Conference, 166-183.
[54] Tang, J., Zhang, G., Lin, B., Zhang, B., 2011. A Hybrid Algorithm for Flexible Job-Shop Scheduling Problem. Procedia Engineering, 5, 3678-3683.
[55] Teekeng, W., Thammano, A., 2011. A Combination of Shuffled Frog Leaping and Fuzzy Logic for Flexible Job-Shop Scheduling Problems., Procedia Computer Science, 6, 69-75.
[56] Teekeng, W., Thammano, A. 2012. Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. Procedia Computer Science, 12, 122-128.
[57] Teymourifar, A., Bahadir, O., Ozturk, G., 2018. Dynamic Priority Rule Selection for Solving Multi-objective Job Shop Scheduling Problems. Universal Journal of Industrial and Business Management,6 (1), 11-22.
[58] Xing, L-N., Chen, Y-W. Wang, P., Zhao, O-S., Xiong, J., 2010. A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems. Applied Soft Computing, 10, (3), 888-896.
[59] Wang (a), L., Zhou, G., Xu, Y., Wang, S., Liu, M., 2012. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. The International Journal of Advanced Manufacturing Technology, 60, (1–4), 303–315.
[60] Wang (b), L., Wang, S., Xu, Y., Zhou, G., Liu, M., 2012. A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. Computers & Industrial Engineering, 62, (4), 917-926.
[61] Yazdani, M., Amiri, M., Zandieh, M., 2010. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Systems with Applications, 37, (1), 678-687.
[62] Yuan, Y., Xu, H., 2012. HHS/LNS: An integrated search method for flexible job shop scheduling," IEEE Congress on Evolutionary Computation, Brisbane, QLD, 1-8.
[63] Yuan, Y., Xu, H., 2013. Flexible job shop scheduling using hybrid differential evolution algorithms., Computers & Industrial Engineering, 65, (2), 246-260.
[64] Zandieh, M., Mahdavi, I., Bagheri, A. 2008. Solving the Flexible Job-Shop Scheduling Problem by a Genetic Algorithm. Journal of Applied Sciences, 8 (24), 4650-4655.
[65] Zhang, G., Gao, L., Shi, Y., 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications, 38, 4, 3563-3573.
..