

Using dispatching rules for job shop scheduling with due date-based objectives

T. C. CHIANG†* and L. C. FU‡

† Affiliation: Dept. Computer Science and Information Engineering, National Taiwan University

Post Addr: No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106

Email: tcchiang@icsee.org

Phone: +886-918891498

Fax: +886-2-23657887

‡ Affiliation: Dept. Computer Science and Information Engineering, National Taiwan University

Post Addr: No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106

Email: lichen@ntu.edu.tw

Phone: +886-2-23622209

Fax: +886-2-23657887

* To whom correspondence should be addressed.

Abstract

This paper addresses the job shop scheduling problem with the due date-based objectives including the tardy rate, the mean tardiness, and the maximum tardiness. The focused approach is the dispatching rules. Eighteen dispatching rules are selected from the literature, and their features and design concepts are discussed. Then a dispatching rule is proposed with the goal as achieving good and balanced performance when more than one objective is concerned at the same time. First, it realizes three good design principles recognized from the existing rules. Second, it introduces a due date extension procedure to solve a problem of negative allowance time. Third, a job candidate reduction mechanism is developed to make the rule computationally efficient. At last, a comprehensive simulation study is conducted with the eighteen existing rules as the benchmarks. The experimental results verify the superiority of the proposed rule, especially on the tardy rate and the mean tardiness.

1. Introduction

Job shop scheduling has attracted researchers from the academia and the industry for several decades due to its high problem complexity and practicality in the real world. In the job shop scheduling problem, there are many machines and the task is to schedule jobs to be processed on the machines following predefined routes so as to optimize or to satisfy the concerned performance criteria.

The common assumptions of the job shop scheduling problem include:

- 1) All jobs are ready at the beginning.
- 2) Processing times are known and fixed. Setup times are included in the processing times.
- 3) One machine can process only one job at a time, and one job can be processed on only one machine at a time.
- 4) Processing of a job on a machine is called an operation, and there is a precedence constraint between operations of a job. It means that the succeeding operation can not start until the preceding operation is finished.
- 5) No preemption is allowed, which means that once an operation starts, it must continue until the processing is finished.
- 6) Machines are continuously available. No machine breakdown or maintenance takes place.

The job shop scheduling problem is often treated as a sequencing problem – to determine the processing order of operations on the machines. In the literature, the approaches to solve job shop scheduling problems include exact algorithms like mathematical programming and branch and bound, search based meta-heuristics like local search and genetic algorithms, and dispatching rules. Dispatching rules are widely accepted in the industry (Appleton-Day *et al.* 1997, Giegandt *et al.* 1998) because of ease of implementation, satisfactory performance, low computation requirement, and flexibility to incorporate domain knowledge and expertise. Thus, many researchers in the academia devoted themselves on designing the dispatching rules and studying their performance on job shop

scheduling. For example, Vepsalainen *et al.* (1987) proposed a parameterized rule, well known as the Apparent Tardiness Cost (ATC) rule to minimize weighted tardiness in the job shop. It assigned priorities to jobs according to the expected delay cost per imminent machine processing time. Anderson *et al.* (1990) developed two rules using dynamic operation due dates based on the remaining allowance times to minimize due date-based objectives in the job shop. The shop floor utilization level was taken to adjust the weights to the processing time and due date information in the rule proposed by Raghu *et al.* (1993). For other reports and surveys of priority rules, see Moser *et al.* (1992), Kim *et al.* (1994), Chang *et al.* (1996), Jeong *et al.* (1998), Rajendran *et al.* (1999), and Jayamonhan *et al.* (2000).

In this paper, we place the focus on solving job shop scheduling problems with due date-based objectives by dispatching rules. In Section 2, we summarize existing rules from the literature, and describe their design principles. Then, Section 3 gives our newly developed dispatching rule, which aims to provide good and balanced performance with respect to multiple objectives simultaneously. The performance of the proposed rule and the existing rules is examined in Section 4, and conclusions are made in Section 5.

2. A survey on existing dispatching rules

Notations for the dispatching rules

p	processing time of the imminent operation
r	remaining processing time of the job (including p)
P	total processing time of the job
q	queueing time of the imminent operation
t	system time, the time at which the dispatching decision is to be made
d	due date of the job
s	slack of the job, $s = d - t - r$
s'	slack of the imminent operation, $s' = d - t - p - c \cdot (r - p)$, where c is a parameter
l	average processing time of the operations at the current machine
l'	total processing time of the operations at the next machine
k_a/k_b	parameter of the rule ATC/COVERT

- Q queue containing waiting jobs
- e how many times the due date of a job has been extended
- u parameter of the proposed rule, the switch of the expediting function
- k parameter of the proposed rule, the multiplier of the remaining processing time to extend the due date
- a allowance time

Objectives

- C_i completion time of a job i
- n number of jobs in the shop
- U_i $U_i = 1$ if $C_i > d_i$, otherwise, $U_i = 0$

tardy rate = $\sum_i U_i / n$; mean tardiness = $\sum_i (C_i - d_i)^+ / n$; maximum tardiness = $\max_i \{(C_i - d_i)^+\}$

There were already many dispatching rules proposed in the literature. Some of them were originally developed for flow time-based objectives but were shown to be effective for due date-based objectives. In table 1, we summarized eighteen rules that were shown to perform well or were commonly used as benchmarks in the literature which focused on due date-based objectives. For each rule, we provide its mathematical equation for calculating priority values, the relationship between the priority value and the involved factors, and some references. With a check in the third column, it means that the job with a larger priority value will be processed earlier; otherwise, the job with a smaller priority value will be processed earlier. A downward/upward arrow in the 4th to 12th columns means that the rule prefers the job with a small/large value of the corresponding factor. The notations used in table 1 are described at the beginning of this section.

<< Insert table 1 about here >>

In table 1, the first four rules used only processing time-related information. The shortest processing time (SPT) rule is undoubtedly one of the most common benchmark rules in the literature, and was shown to provide good performance on minimizing the tardy rate in the works of Chang *et al.* (1996)

and Rajendran *et al.* (1999), especially when the shop utilization level is high or when the due dates of jobs are tight. Rules including shortest remaining processing time (SRPT), least total workload (LTWK), and shortest processing time over total workload (SPT/TWK) extend the SPT concept to multiple operations and were also shown to perform well on minimizing the tardy rate (Chang *et al.* 1996).

Following those mentioned above in the same table are five rules that added one more factor, “due date,” when calculating the priority values. By simply using the due dates, the earliest due date (EDD) rule was shown to be the best for minimizing the maximum tardiness in the study of Barman (1998), and was also shown to perform well on minimizing the tardy rate (Chang *et al.* 1996) and the mean tardiness (Jeong *et al.* 1998). The modified due date (MDD) rule, which is a combination of EDD and SRPT, was shown to be good at minimizing the mean tardiness (Jeong *et al.* 1998 and Kim *et al.* 2003). On the contrary, the operation due date (ODD) rule achieves good performance on minimizing the maximum tardiness (Raghu *et al.* 1993 and Jayamonhan *et al.* 2000) by combining EDD and “the longest remaining processing time first (LRPT).” The modified operation due date (MOD) rule, which combines ODD and SPT, was verified to be good for minimizing the mean tardiness in the works by Jeong *et al.* (1998) and Kim *et al.* (2003).

More complicated combinations of processing time-related and due date-related information lead to the rules such as cost over time (COVERT), apparent tardiness cost (ATC), critical ratio (CR), CR+SPT and S/RPT+SPT. The COVERT rule is a popular benchmark rule when the mean tardiness is considered, and was shown to perform well in the study of Rajendran *et al.* (1999) and Kim *et al.* (2003). In the work of Rajendran *et al.* (1999), COVERT also provided good results with respect to the maximum tardiness. The CR rule uses the ratio of the allowance time (time left to the due date) to the remaining processing time to determine jobs’ priority. In the study of Shafaei *et al.* (1999b), CR was the second best rule for a cost-based performance considering tardiness and holding costs.

Abu-Suleiman *et al.* (2005) developed a parameterized CR rule, which was shown to outperform EDD and SPT on several due date-based objectives. The CR rule was also utilized for scheduling in the semiconductor wafer fabrication facility, which can be seen as a complex job shop, in Rose (2002). Rules CR+SPT and S/RPT+SPT were proposed by Anderson *et al.* (1990), and are the dynamic version of the MOD rule. Instead of the fixed allocation of the allowance time to operations, these two rules allocate the allowance time to operations according to the critical ratio and the ratio of the slack to the remaining processing time, respectively. CR+SPT was shown to provide good performance on minimizing the mean tardiness (Anderson *et al.* 1990), and S/RPT+SPT was verified to be good for reducing the mean tardiness and the tardy rate (Raghu *et al.* 1993).

The last four rules in table 1 are relatively new rules, and were discussed in recent studies by Rajendran *et al.* (1999) and Jayamonhan *et al.* (2000). In their studies, PT+PW+ODD can achieve good performance on minimizing both the mean tardiness and the tardy rate. As for PT+WING+SLACK, it was shown to be good on minimizing the maximum tardiness.

In summary, we can find the common characteristics for the rules that are good at each performance criterion. To minimize the tardy rate, rules that follow the concept of “shorter processing time earlier” usually provide good results, in which the processing time may refer to the time required to finish the imminent, remaining, or all operations. Considering minimizing the maximum tardiness, rules that adopt the slack (either of the job itself or of the imminent operation) as the major factor often give satisfactory performance. The spirit “less slack earlier” is actually a combination of “earlier due date earlier” and “longer remaining processing time earlier.” As for rules that perform well with respect to the mean tardiness, they often combine several principles. For example, COVERT and CR+SPT realize “shorter processing time earlier,” “longer remaining processing time earlier,” and “earlier due date earlier” in their calculations. It implies that more factors should be taken into consideration for minimizing the mean tardiness than other performance measures.

3. The proposed dispatching rule

Recently the authors (Chiang *et al.* 2004) proposed a dispatching rule named enhanced critical ratio (ECR), whose feature is to use “group information” to prioritize jobs. The priority value of a job depends on the influence upon all competing jobs caused by first processing of it. In that way, this rule combines the concepts of “shorter processing time earlier,” “earlier due date earlier,” and “longer remaining processing time earlier,” and was shown to perform well on minimizing the tardy rate. However, this combined principle is only applicable to non-tardy jobs in ECR. For tardy jobs, ECR used a separate mechanism to calculate their priority values and lost the merits of the combined principle. In this paper, we extend the ECR rule, and make the combined principle applicable to all jobs by introducing a due date extension procedure. Two new factors are involved in the procedure – e and d^e . The factor e refers to how many times the due date of a job has been extended, and the factor d^e is the due date set at the e^{th} extension. The initial value of e_j for each job j is zero, and the initial value of d_j^0 is its real due date (d_j) given by the problem. These two factors are used only inside the rule, and the extended due date will not affect the real due date. Besides, there are two parameters, k and u , in the rule. The parameter k is related to how long the due date will extend, and the parameter u controls whether to expedite the tardy jobs. In the following, we will show the detailed steps of the proposed rule first, and then give the explanations for the design spirit and the usage of parameters k and u .

3.1 Detailed algorithm and design concept

Detailed steps

Step1. *Due date extension*

For each job j waiting in the queue Q , if $t + r_j > d_j^{e_j}$, then $e_j = e_j + 1$, $d_j^{e_j} = d_j^{e_j-1} + k \cdot r_j$

Step2. *Influence estimation*

For each job j , estimate the influence upon all waiting jobs by first processing of it.

$$V_j = \sum_{i \in Q, i \neq j} \text{urg}(r_i, d_i^{e_i} - p_j - t, e_i) + \text{urg}(r_j - p_j, d_j^{e_j} - p_j - t, e_j) \quad \text{where}$$

$$\begin{aligned} \text{urg}(r, a, e) &= (e + 1)^u - 1 && \text{if } r = 0 \\ &= (e + 1)^u \cdot (r/a)^2 && \text{if } a \geq r > 0 \\ &= (e + 1)^u && \text{if } a < r. \end{aligned}$$

Select the job j with the smallest V_j to be the next processing target.

■

To explain the rule, let us start from Step2. In Step2, the influence caused by first processing of the job j is estimated by the changes to the degree of urgency of all competing jobs (including itself). In most cases, the degree of urgency of a job is assessed by the square of the ratio of the remaining processing time to the allowance time. A larger ratio implies higher urgency, and the square is used so as to make the degree of urgency raise faster when the allowance time is getting closer to the remaining processing time.

For job j , the remaining processing time will decrease by the processing time of its imminent operation, but for all other jobs, the remaining processing times keep unchanged. As for the allowance time, it is the time to the corresponding due date of each job ($d_i^{e_i} - t$) less the processing time of the imminent operation of the job j (p_j). The degree of urgency will decrease for job j and will increase for all other jobs. The longer the remaining processing time of j is, the more its degree of urgency decreases; the shorter the processing time of the imminent operation of j is, the less the degree of urgency of other competing jobs increases. Thus, by giving the highest priority to the job with the smallest V_j , which means that the first processing of this job will maintain the lowest total degree of urgency of all jobs, the proposed rule realizes the concepts of “shorter processing time earlier,” “earlier due date earlier,” and “longer remaining processing time earlier.”

For tardy jobs, the ratio of the remaining processing time to the allowance time becomes negative due to negative allowance time. That causes a conflict with the spirit of the rule since now the degree of urgency of a tardy job gets lower and lower as time goes by. In the proposed rule, we solve this problem by introducing the due date extension procedure in Step1. When a job becomes tardy, we extend its due date by k times its remaining processing time where k is a rule parameter (note that the extended due date is only for the rule). By extending the due date, the ratio in Step2 will be kept positive, and the aforementioned combined principle of the rule is now also applicable to tardy jobs. The value of parameter k decides the starting degree of urgency of the tardy job after due date extension, and its impact on the performance will be discussed in Section 4 later.

At last, we explain the use of the factor e and the second parameter u of the proposed rule in Step2. The value of e_j starts from zero, and increases by one each time the job j receives a due date extension. In order to expedite tardy jobs, we make the degree of urgency in Step2 in proportion to $(e+1)$. In other words, the urgency of a job will increase or decrease faster when it receives more due date extensions. The urgency of a job j is confined between $(e_j+1)^u - 1$ and $(e_j+1)^u$. If a job with only one remaining operation is selected to be the next processing target, its urgency is set as $(e_j+1)^u - 1$. If a job cannot meet its extended due date because another job is selected as the next processing target, its urgency is set as $(e_j+1)^u$. As mentioned, in most cases, the urgency of a job is in proportion to the square of the ratio of the remaining processing time to the allowance time. Introducing the factor e in the urgency function could effectively reduce the maximum tardiness; however, expediting tardy jobs could also increase the tardy rate. If minimizing the tardy rate has higher importance than the maximum tardiness, this expediting effect can be switched off by changing u from one to zero. That is the function of u .

3.2 Example

In the following, we use an example to demonstrate how the proposed rule works. There are four jobs and four machines in the example. The routes of jobs, processing times of operations, and due

dates of jobs are given in table 2. Assume the partial schedule is as illustrated in figure 1. The system time t is 30, and now machine M3 finishes the third operation of Job1. The values of rule parameters k and u are set as two and one, respectively. The proposed rule is invoked, and the calculation is as follows:

<< Insert table 2 and figure 1 about here >>

Step0 *Initialization*

$$Q = \{2, 3, 4\},$$

$$r_2 = 10+8+7 = 25, r_3 = 20+4+21 = 45, r_4 = 15+9+16 = 40$$

Step1. *Due date extension*

$$j = 2, 30 + 25 \leq 70 \dots \text{no extension of due date}$$

$$j = 3, 30 + 45 > 70, e_3 = 1, d_3^l = 30 + 2 \times 45 = 120$$

$$j = 4, 30 + 40 \leq 90 \dots \text{no extension of due date}$$

Step2-3. *Influence estimation*

$$V_2 = \left(\frac{25-10}{70-30-10}\right)^2 \cdot 1 + \left(\frac{45}{120-30-10}\right)^2 \cdot 2 + \left(\frac{40}{90-30-10}\right)^2 \cdot 1 \cong 1.523$$

$$V_3 = 1^\psi + \left(\frac{45-20}{120-30-20}\right)^2 \cdot 2 + \left(\frac{40}{90-30-20}\right)^2 \cdot 1 \cong 2.255 \quad (\psi: 70-30-20 < 25)$$

$$V_4 = \left(\frac{25}{70-30-15}\right)^2 \cdot 1 + \left(\frac{45}{120-30-15}\right)^2 \cdot 2 + \left(\frac{40-15}{90-30-15}\right)^2 \cdot 1 \cong 2.028$$

Step4. Finally, Job2 is selected as the next processing target for machine M3. ■

3.3 Job candidate reduction mechanism

In the proposed rule, evaluating the influence upon all competing jobs is helpful (as will be shown in Section 4) to select an appropriate next processing job. However, this evaluation could also take long computation time due to the $O(n^2)$ time complexity with n being the number of job candidates. In this subsection, we will show how we make the proposed rule computationally efficient by a job candidate

reduction mechanism. The reduction mechanism is based on the following theorem.

Theorem 1. *Dominance relationship between job candidates*

Given two jobs i and h , if the following two conditions are satisfied, then $V_i < V_h$ in Step2 of the proposed rule. In this condition, we say job i *dominates* job h , job i is dominant and job h is non-dominant.

$$(1) p_i < p_h,$$

$$(2) \text{urg}(r_h, d_h^{e_h} - p_i - t, e_h) + \text{urg}(r_i - p_i, d_i^{e_i} - p_i - t, e_i) < \text{urg}(r_i, d_i^{e_i} - p_h - t, e_i) + \text{urg}(r_h - p_h, d_h^{e_h} - p_h - t, e_h).$$

Proof:

$$\begin{aligned} V_i &= \sum_{j \in Q, j \neq i} \text{urg}(r_j, d_j^{e_j} - p_i - t, e_j) + \text{urg}(r_i - p_i, d_i^{e_i} - p_i - t, e_i) \\ &= \sum_{j \in Q, j \neq i, j \neq h} \text{urg}(r_j, d_j^{e_j} - p_i - t, e_j) + \text{urg}(r_h, d_h^{e_h} - p_i - t, e_h) + \text{urg}(r_i - p_i, d_i^{e_i} - p_i - t, e_i) \\ V_h &= \sum_{j \in Q, j \neq h} \text{urg}(r_j, d_j^{e_j} - p_h - t, e_j) + \text{urg}(r_h - p_h, d_h^{e_h} - p_h - t, e_h) \\ &= \sum_{j \in Q, j \neq i, j \neq h} \text{urg}(r_j, d_j^{e_j} - p_h - t, e_j) + \text{urg}(r_i, d_i^{e_i} - p_h - t, e_i) + \text{urg}(r_h - p_h, d_h^{e_h} - p_h - t, e_h) \end{aligned}$$

$$\text{Let } Y = \sum_{j \in Q, j \neq i, j \neq h} \text{urg}(r_j, d_j^{e_j} - p_i - t, e_j) - \sum_{j \in Q, j \neq i, j \neq h} \text{urg}(r_j, d_j^{e_j} - p_h - t, e_j)$$

$$\begin{aligned} V_i - V_h &= (\text{urg}(r_h, d_h^{e_h} - p_i - t, e_h) + \text{urg}(r_i - p_i, d_i^{e_i} - p_i - t, e_i)) - \\ &\quad (\text{urg}(r_i, d_i^{e_i} - p_h - t, e_i) + \text{urg}(r_h - p_h, d_h^{e_h} - p_h - t, e_h)) + Y. \end{aligned}$$

Because $p_i < p_h$ and $\text{urg}(\cdot)$ is nondecreasing as a decreases, we have $Y \leq 0$.

According to the assumption, we have $\text{urg}(r_h, d_h^{e_h} - p_i - t, e_h) + \text{urg}(r_i - p_i, d_i^{e_i} - p_i - t, e_i) < \text{urg}(r_i, d_i^{e_i} - p_h - t, e_i) + \text{urg}(r_h - p_h, d_h^{e_h} - p_h - t, e_h)$. Therefore, $V_i - V_h < 0$, namely, $V_i < V_h$.

■

Corollary 1. *Job candidate set reduction*

Based on Theorem 1, the job candidates necessary to be evaluated in the proposed rule can be reduced to the set of dominant jobs.

Proof:

Assume that a job i is not dominant, namely, it is dominated by one job, job j . By Theorem 1, $V_i > V_j$. Thus, the job i can not be the one with the smallest V value, and can not be the next processing target. In other words, it can be removed from the set of job candidates being evaluated by the proposed rule. ■

After the derivation of the aforementioned theorem and corollary, what we need is an efficient procedure to find out the dominant jobs. Here we resort to the fast dominant sorting approach proposed by K. Deb *et al.* (2002). This approach was originally used to sort individuals based on the dominance relationships in a multi-objective genetic algorithm, and here we take this procedure to be a tool to efficiently reduce the size of job candidate set for our proposed rule. Its time complexity is $O(n)$ empirically, and the algorithm is shown in table 3 for completeness.

<< Insert table 3 about here >>

4. Experiments and results**4.1 Benchmark problems**

To examine the performance of the proposed rule and existing rules, we used the benchmark problems from Demirkol *et al.* (1998) since they are publicly accessible and due dates are involved. In the benchmarks, all jobs are available at time zero. Other related information is given in table 4. For each parameter combination, there are five problem instances. Thus, there are 160 ($4 \times 2 \times 2 \times 2 \times 5$) problem instances in total.

<< Insert table 4 about here >>

From the experimental results, we found that the tardy rates of the instances from Demirkol *et al.* are very high. To examine these nineteen rules in the job shops with lower tardy rates, we also generated a data set of 50 instances by ourselves. In our own data set, there are 200 jobs, 20 machines, and the processing time is generated from Uniform[1, 100]. Due dates of jobs are set according to the TWK (Total Workload) method with tightness factor generated from 6 to 10 uniformly.

4.2 Performance of the proposed rule and eighteen existing rules

Performance of the proposed rule and eighteen existing rules in table 1 are examined with respect to three due date-based objectives – the tardy rate, the mean tardiness, and the maximum tardiness. For the existing rules with parameters, including ODD, MOD, COVERT, ATC, and PT+PW+ODD, ten values from 1 to 10 were tested. As for the proposed rule, eight combinations of two parameters (k and u) were tested. The results obtained by taking the mean over 160 public problem instances and 50 self-generated problem instances are summarized in table 5 and 6, respectively. Due to the limitation of space, we only list performance results of five parameters for each of the existing rules with parameters (note that such omission does not affect the relative ranks between the rules). In table 5 and 6, the proposed rule is denoted by ECR-II.

<< Insert table 5 about here >>

<< Insert table 6 about here >>

4.2.1 Performance results with the public instances

To minimize the tardy rate for the public instances, the best five rules are ECR-II, SRPT, LTWK, PT+PW, and SPT. The proposed rule performs well on minimizing the tardy rate with a wide range of parameter values. The common feature of the best five rules is that they all adopt the “shorter processing time earlier” principle. Note that SRPT, LTWK, and SPT use this principle only. The

average tardy rates are very high in these public instances by all rules. It was reported that under the condition in which due dates are tight for jobs, rules following the “shorter processing time earlier” concept usually perform well. Our observation is consistent with the finding.

To minimize the mean tardiness, the best five rules are MDD, ATC, CR+SPT, COVERT, and MOD. These rules were reported to be good at reducing the mean tardiness, and they all consider both processing time and due date information simultaneously. Our proposed rule ranks the sixth, and the mean tardiness is greater than MDD by less than 1%.

The best five rules for minimizing the maximum tardiness are PT+WINQ+SLACK, ODD, SLACK, PT+PW+ODD, and EDD. Apparently, they are similar in the use of slack-based information. Our proposed rule is the seventh best rule among the nineteen ones. Although the maximum tardiness achieved by the proposed rule is 30% higher than that of the best rule, it significantly outperforms those best sixth rules with respect to both the tardy rate and the mean tardiness.

As can be seen in table 5, no rule can dominate all the others in terms of all three performance objectives. We provide figure 2 in order to examine the performance of the nineteen rules on three concerned objectives easily. In this figure, performance measures of three objectives are normalized, with the minimum/maximum value being normalized to zero/one. We also show the data points projected to the plane of normalized tardy rate and normalized mean tardiness so that the normalized maximum tardiness is easier to check. Observing the projected data points, we can find that the proposed rule can dominate many rules when the tardy rate and the mean tardiness are concerned. No rule can dominate the proposed rule for these two objectives. On the ground plane, there are two data points of other rules close to the origin, which represent the SRPT and LTWK rules. Although they provide satisfactory performance on the tardy rate and the mean tardiness, their performance on the maximum tardiness is quite bad, as can be seen in the figure. Considering the maximum tardiness, we can classify the existing rules into two groups, one for low mean tardiness and the other for low

maximum tardiness. Note also that, most rules in the group with lower maximum tardiness have very high tardy rate. The representative rule in the first group is ATC, with mean tardiness around 2000 and the maximum tardiness around 4000. The representative rules in the second group are EDD and PT+WING+SLACK, with mean tardiness from 2100 to 2200 and the maximum tardiness from 3500 to 3000. These three rules do not dominate one another. The proposed rule provides performance close to ATC when mean tardiness and the maximum tardiness are considered at the same time. In summary, the proposed rule can lower the tardy rate, keep the least mean tardiness, and give acceptable maximum tardiness when comparing with the existing rules. The difficulty to generate solutions that are good on all three objectives could be due to the nature of problem instances in the data set. Further research will continue in the future works.

<< Insert figure 2 about here >>

4.2.2 Performance results with the self-generated instances

The second data set includes 50 self-generated problem instances with much lower tardy rate than the first data set. It is used to examine the performance of rules in different shop conditions. In this data set, the best five rules to minimize the tardy rate are ECR-II, S/RPT+SPT, ATC, MDD, and PT+PW+ODD. The tardy rate of the second best rule is almost twice as high as the tardy rate of the proposed rule. Note that the proposed rule is ranked as the best rule in both data sets for this criterion. Except the proposed rule, no rule is ranked in the best five in both data sets. This observation shows that the advantage of the proposed rule on minimizing the tardy rate is robust in different shop conditions.

Concerning the mean tardiness, ECR-II, CR+SPT, ATC, MOD, and S/RPT+SPT are the best five rules. Although CR+SPT, ATC, and MOD consistently perform well on this criterion in both data sets, the proposed rule can provide better results when the tardy rate and the mean tardiness are taken into consideration at the same time.

For the last criterion, the maximum tardiness, the best five rules include ODD, SLACK, CR, CR+SPT, and ATC. The proposed rule ranks as the eighth. ODD and SLACK are common in the best five rules in both data sets, and again shows the strength of slack on reducing the maximum tardiness.

The normalized performance of rules on three concerned objectives is illustrated in figure 3. The advantage of the proposed rule on the tardy rate and mean tardiness is clear in the figure. It can dominate all rules when only these two objectives are considered. Taking the maximum tardiness into consideration, the proposed rule can dominate two-third of the existing rules. As mentioned in Section 4.1, this data set was generated for the environments with lower tardy rates than the previous (public) data set. Hence, it is more likely to obtain solutions that are good on all three objectives. In this situation, the proposed rule is able to construct schedules with the lowest tardy rate, the least mean tardiness, and quite low maximum tardiness. It is also interesting to observe the performance of the existing rules in two data sets. In the previous data set, where objectives conflict with one another, the existing rules can usually find their advantages on one objective. But in this data set, where solutions that are good on one objective are also good on the others, many existing rules could perform badly on all objectives.

<< Insert figure 3 about here >>

4.3 Effects of the parameters of the proposed rule

There are two parameters in our proposed rule, u for controlling the expediting function and k for the starting degree of urgency of tardy jobs. First, we observe the effect of the parameter u on the tardy rate and the maximum tardiness. With u being zero, which means we switch off the expediting function, we can obtain lower tardy rate. By setting u as one to switch on the expediting function, the maximum tardiness can be reduced at the expense of larger tardy rate. As for the parameter k , the smaller the value of k is, the higher the starting degree of urgency is assigned to the tardy job. When we decrease

the value of k , schedules with lower mean tardiness can be generated but also with a loss of meet-due-date jobs. With an adequate setting of these two parameters, the managers can generate schedules with desired compromise between multiple performance criteria.

4.4 Computation efficiency of the proposed rule

In Section 3.3, we mentioned about the $O(n^2)$ time complexity of the proposed rule and also developed a job candidate reduction mechanism to improve its computation efficiency. To examine the effect of the job candidate reduction mechanism, we generated fifty more problem instances only for this experiment on computation efficiency. These instances fall into five groups with the number of jobs being 100, 200, 300, 400, and 500, respectively. The number of machines is always ten. Average computation times over ten instances in each category are shown in figure 4. The SPT rule is taken as a reference since its calculation is very simple. From figure 4, it is apparent that the proposed job candidate set reduction mechanism is very effective on shortening the computation time of our rule. With the reduction mechanism, our rule can run almost as fast as the SPT rule.

<< Insert figure 4 about here >>

5. Conclusion

In this paper, we discuss the use of dispatching rules to solve the job shop scheduling problems concerning the due date-based criteria. Eighteen existing rules that were popularly adopted for due date-based objectives in the literature are first examined. We analyze the advantages of these rules and the core design concepts inside them. Then, we propose a rule which combines three key design concepts including “shorter processing time earlier,” “earlier due date earlier,” and “longer remaining processing time earlier” in order to generate schedules with good and balanced performance on multiple due date-based objectives. Besides, we also develop a job candidate reduction mechanism,

which aims to improve the computational efficiency of the proposed rule. Finally, a comprehensive simulation study is conducted to examine the performance of the proposed rule and the existing rules. The experimental results confirm the advantage of the proposed rule on multiple criteria in different shop conditions, especially when the tardy rate and the mean tardiness are the major concerns. It indicates that the proposed rule is a promising alternative for the job shops in which due date-based objectives are concerned. The job candidate reduction mechanism is also shown beneficial by observing that the proposed rule can run almost as fast as a very simple rule. Hence, the proposed rule is not only effective but also efficient to be used in the environment where real-time dispatching is required.

In the experimental results, performance of the proposed rule on minimizing the maximum tardiness is satisfactory but not as good as that on the tardy rate and the mean tardiness. In our future works, there are three planned research directions:

- (1) We will enhance the performance of the proposed rule on the maximum tardiness while keeping its superiority on the other two criteria.
- (2) In the shops where there is an essential conflict between the tardy rate and the maximum tardiness, which means that we can not reduce both of them at the same time, we want to make the proposed rule able to generate a number of compromising solutions through tuning the parameters of the rule.
- (3) We will try to integrate the proposed rule and multi-objective meta-heuristics such as genetic algorithms so that tuning the parameters of the rule is done automatically and intelligently by the meta-heuristics.

References

- Abu-Suleiman, A., Pratt, D.B., and Boardman, B., 2005, The modified critical ratio: towards sequencing with a continuous decision domain, *International Journal of Production Research*, **43** (15), 3287 – 3296.
- Anderson, E.J. and Nyirenda, J.C., 1990, Two new rules to minimize tardiness in a job shop, *International Journal of Production Research*, **28** (3), 2277 – 2292.
- Appleton-Day, K. and Shao, L., 1997, Real-time dispatch gets real-time results in AMD's Fab25, *Proc. of IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 444 – 447.
- Barman, S., 1997, Simple priority rule combinations: an approach to improve both flow time and tardiness, *International Journal of Production Research*, **35** (10), 2857 – 2870.
- Barman, S., 1998, The impact of priority rule combinations on lateness and tardiness, *IIE Transactions*, **30**, 495 – 504.
- Chang, Y.-L., Sueyoshi, T. and Sullivan, R. S., 1996, Ranking dispatching rules by data envelopment analysis in a job shop environment, *IIE Transactions*, **28**, 631 – 642.
- Chiang, T.-C. and Fu, L.-C., 2004, Solving the FMS scheduling problem by critical ratio-based heuristics and the genetic algorithm, *Proc. of IEEE Conference on Robotics and Automation*, **3**, 3131 – 3136.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, **6** (2), 181 – 197.
- Demirkol, E., Mehta, S. and Uzsoy, R., 1998, Benchmarks for shop scheduling problems, *European Journal of Operational Research*, **109**, 137 – 141.
- Giegandt, A. and Nicholson, G., 1998, Better dispatch application – a success story, *Proc. of IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 396 – 399.
- Jayamonhan, M.S. and Rajendran, C., 2000, New dispatching rules for shop scheduling: a step forward, *International Journal of Production Research*, **38** (3), 563 – 586.

- Jeong, K.-C. and Kim, Y.-D., 1998, A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules, *International Journal of Production Research*, **36** (9), 2609 – 2626.
- Kim, M. H. and Kim, Y.-D., 1994, Simulation-based real-time scheduling in a flexible manufacturing system, *Journal of Manufacturing Systems*, **13** (2), 85 – 93.
- Kim, Y.-D., Shim, S.-O., Choi, B. and Hwang, H., 2003, Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility, *IEEE Transactions on Semiconductor Manufacturing*, **16** (2), 290 – 298.
- Lodree Jr., E., Jang, W. and Klein, C.M., 2004, A new rule for minimizing the number of tardy jobs in dynamic flow shops, *European Journal of Operational Research*, **159**, 258 – 263.
- Moser, M. and Engell, S., 1992, A survey of priority rules for FMS scheduling and their performance for the benchmark problem, *Proc. of the 31st Conference on Decision and Control*, 392 – 397.
- Pierreval, H. and Mebarki, N., 1997, Dynamic selection of dispatching rules for manufacturing system scheduling, *International Journal of Production Research*, **35**, 1575 – 1591.
- 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.
- Rajendran, C. and Holthaus, O., 1999, A comparative study of dispatching rules in dynamic flowshops and job shops, *European Journal of Operational Research*, **116**, 156 – 170.
- Rose, O., 2002, Some issues of the critical ratio dispatch rule in semiconductor manufacturing, *Proc. of the 2002 Winter Simulation Conference*, 1401 – 1405.
- Russell, R.S., Dar-El, E.M. and Taylor III, B.W., 1987, A comparative analysis of the COVERT job sequencing rule using various shop performance measures, *International Journal of Production Research*, **25** (10), 1523 – 1540.

- Shafaei, R. and Brunn, P., 1999, Workshop scheduling using practical (inaccurate) data – Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach, *International Journal of Production Research*, **37** (17), 3913 – 3925.
- Shafaei, R. and Brunn P., 1999, Workshop scheduling using practical (inaccurate) data – Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment, *International Journal of Production Research*, **37** (18), 4105 – 4117.
- Vepsalainen, A. P. J. and Morton, T. E., 1987, Priority rules for job shops with weighted tardiness costs, *Management Science*, **33** (8), 1035 – 1047.

Table 1

Collection of the best existing rules for due date-based objectives

	$Z =$	◆	p	r	P	q	d	s	s'	l	l'	Tested in ^ψ
SPT	p			↓								M92, R93, K94, C96, P97, B97, J98, B98, R99, S99ab, J00, L04
SRPT	r				↓							C96, J98
LTWK	P					↓						K94, C96, J98
SPT/TWK	p/P			↓	↑							K94, C96, J98
EDD	d							↓				R87, M92, R93, C96, B97, J98, B98, R99, S99ab, J00
MDD	$\max\{d, t+r\}$				↓			↓				R87, K94, J98, K03
ODD	$s'+p$								↓			M92, C96, J00
MOD	$\max\{s'+p, p\}$			↓					↓			R87, M92, R93, K94, J98, S99ab, J00, K03
SLACK	s							↓				R87, K94, C96, J98, B98
COVERT	$(1/p) \cdot (1 - s^+ / (k_b \cdot (r - p)))^+$	√	↓	↑				↓				R87, M92, K94, P97, J98, R99, S99ab, K03
ATC	$(1/p) \cdot \exp(-(s' / (k_a \cdot l))^+)$	√	↓					↓		ϕ		R93, K94, J98, S99ab, J00
CR	$(d - t)/r$				↑			↓				M92, P97, S99ab, R02, A05
CR+SPT	$\max\{(d - t)/r \cdot p, p\}$			↓	↑			↓				A90, M92, R93, P97, S99ab
S/RPT+SPT	$\max\{(s/r) \cdot p, p\}$			↓	↑			↓				A90, R93
PT+PW	$p+q$			↓			↓					J00
PT+PW+ODD	$p+q+s'$			↓			↓		↓			J00
WINQ	l'										↓	M92, C96
PT+WINQ+SLACK	$p + l' + s^-$			↓				↓			↓	R99, J00

◆ The larger, the higher priority

ψ The letter stands for the first letter of the first author, and the number stands for the year of publication.

ϕ If it increases, the influence of s decreasesTable 2
Data of the example problem

	Op1	Op2	Op3	Op4	Due date
Job1	M1 / 10	M4 / 5	M3 / 15	M2 / 9	80
Job2	M2 / 20	M3 / 10	M1 / 8	M4 / 7	70
Job3	M2 / 6	M3 / 20	M2 / 4	M1 / 21	70
Job4	M1 / 7	M3 / 15	M4 / 9	M2 / 16	90

Table 3
Fast dominant sorting approach

P : the set of all competing jobs in the queue
 P' : the reduced set of job candidates

$P' = \text{find-dominant-jobs}(P)$

$P' = \{\}$

For each $i \in P$ and $i \notin P'$

$P' = P' \cup \{i\}$

For each $j \in P$ and $j \neq i$

If i dominates j , then $P' = P' \setminus \{j\}$

Else if j dominates i , $P' = P' \setminus \{i\}$

Table 4
Data of the benchmark problem set one

Problem parameter	Values considered	Total values
Processing time (p)	Uniform [1, 200]	1
Number of jobs (n)	20, 30, 40, 50	4
Number of machines	15, 20	2
Job due date	$d_j = \text{Uniform}[\mu - (\mu R)/2, \mu + (\mu R)/2]$	
	$\mu = (1.0 - T) \cdot n \cdot E[p]$	
	$T = \text{percentage tardy jobs} = 0.3, 0.6$	2
	$R = \text{due date range} = 0.5, 2.5$	2

Table 5
Performance and ranks of the dispatching rules with 160 public problem instances

	Tardy rate (%)		Mean tardiness		Maximum tardiness		
SPT	91.59	11 *	2057.0	22	4257.9	38	
SRPT	87.92	4 *	2045.6	19	5071.4	46	
LTWK	88.51	5 *	2066.0	23	4925.6	45	
SPT/TWK	92.60	15	2131.7	29	4201.2	32	
SLACK	97.95	45	2287.7	37	3107.6	3 *	
EDD	96.74	33	2102.2	27	3537.0	9 *	
MDD	95.83	25	1988.7	1 *	4744.0	44	
CR+SPT	96.65	30	1997.6	3 *	4065.4	20	
S/RPT+SPT	92.81	17	2011.9	12	4250.5	35	
	$c=2$	98.06	46	2302.8	38	3045.2	2 *
	$c=4$	97.65	41	2390.2	40	3312.5	5 *
ODD	$c=6$	97.00	37	2438.8	43	3512.3	7
	$c=8$	96.95	36	2469.6	45	3618.4	10
	$c=10$	96.65	31	2488.3	46	3716.7	13
	$c=2$	94.95	23	2003.2	7 *	4151.7	26
	$c=4$	92.73	16	2034.4	17	4255.7	36
MOD	$c=6$	92.50	14	2043.9	18	4270.8	40
	$c=8$	92.41	12	2049.2	20	4256.9	37
	$c=10$	92.42	13	2050.5	21	4262.7	39
	$k_b=2$	96.51	28	1999.9	5 *	4101.4	24
	$k_b=4$	95.26	24	2003.8	9	4181.4	28
COVERT	$k_b=6$	94.13	22	2009.2	11	4207.8	33
	$k_b=8$	93.43	19	2011.9	13	4183.3	29
	$k_b=10$	93.00	18	2017.4	15	4190.3	31
	$k_a=2$	97.43	40	2014.8	14	4018.3	16
	$k_a=4$	96.92	35	2003.7	8	4034.8	19
ATC	$k_a=6$	96.81	34	2001.5	6	4068.1	21
	$k_a=8$	96.32	27	1998.0	4 *	4070.9	22
	$k_a=10$	96.00	26	1989.5	2 *	4032.2	18
CR		97.94	44	2105.3	28	3687.4	12
PT+PW		91.03	9 *	2087.0	25	4533.5	43
	$c=2$	97.86	43	2222.2	33	3281.2	4 *
	$c=4$	97.79	42	2325.9	39	3405.4	6
PT+PW+	$c=6$	97.22	39	2396.9	41	3527.2	8
ODD	$c=8$	96.71	32	2434.6	42	3648.2	11
	$c=10$	96.62	29	2463.7	44	3719.4	14
WINQ		93.73	21	2232.1	34	4273.3	41
PT+WINQ+SLACK		97.08	38	2183.8	32	3027.5	1 *
ECR-II $u = 0, k = 1.5$		90.45	8 *	2007.6	10	4293.3	42
ECR-II $u = 0, k = 2.0$		88.68	6 *	2084.1	24	4185.6	30
ECR-II $u = 0, k = 2.5$		86.84	2 *	2172.2	31	4170.1	27
ECR-II $u = 0, k = 3.0$		85.73	1 *	2244.7	36	4242.5	34
ECR-II $u = 1, k = 1.5$		93.50	20	2020.2	16	4070.9	22
ECR-II $u = 1, k = 2.0$		91.32	10	2088.9	26	3912.9	15
ECR-II $u = 1, k = 2.5$		88.88	7 *	2170.7	30	4026.3	17
ECR-II $u = 1, k = 3.0$		87.18	3 *	2243.6	35	4151.4	25

* refers to the best five rules

Table 6

Performance and ranks of the dispatching rules with 50 self-generated problem instances

	Tardy rate (%)		Mean tardiness		Maximum tardiness		
SPT	41.80	28	357.15	39	2527.93	43	
SRPT	28.36	18	258.83	34	2464.02	42	
LTWK	28.01	17	215.86	32	2000.48	36	
SPT/TWK	48.11	33	561.77	44	3063.36	46	
SLACK	26.17	16	61.09	18	465.91	2 *	
EDD	24.51	13	66.03	20	639.07	6	
MDD	22.70	11 *	65.00	19	955.95	19	
CR+SPT	29.01	21	34.43	3 *	622.56	4 *	
S/RPT+SPT	19.79	9 *	48.82	13 *	1030.81	22	
	$c=2$	31.02	23	57.53	15	364.20	1 *
	$c=4$	62.83	37	176.68	29	640.08	7
ODD	$c=6$	89.57	44	383.07	41	899.94	14
	$c=8$	92.68	46	554.84	43	1107.16	24
	$c=10$	92.21	45	659.25	46	1323.20	29
	$c=2$	25.41	15	47.41	11 *	777.07	9
	$c=4$	45.43	31	132.28	26	1450.95	31
MOD	$c=6$	63.32	38	247.76	33	2016.81	37
	$c=8$	59.09	36	278.69	36	2372.72	40
	$c=10$	57.15	35	298.31	38	2456.43	41
	$k_b=2$	74.36	42	159.19	27	1452.70	32
	$k_b=4$	44.80	29	91.17	23	1249.94	27
COVERT	$k_b=6$	31.34	24	57.46	14	1110.95	25
	$k_b=8$	28.72	20	57.93	16	1162.64	26
	$k_b=10$	28.49	19	66.71	21	1261.67	28
	$k_a=2$	69.10	40	169.18	28	1335.64	30
	$k_a=4$	44.80	29	98.77	24	1052.15	23
ATC	$k_a=6$	32.31	25	67.46	22	876.35	13
	$k_a=8$	24.61	14	48.18	12	720.21	8
	$k_a=10$	20.35	10 *	38.18	7 *	638.92	5 *
CR		48.14	34	59.82	17	483.51	3 *
PT+PW		37.22	27	374.09	40	2763.87	44
	$c=2$	22.91	12 *	123.41	25	1625.34	33
	$c=4$	30.15	22	188.02	30	1936.88	34
PT+PW+	$c=6$	34.69	26	215.68	31	1983.37	35
ODD	$c=8$	47.95	32	263.97	35	2017.00	38
	$c=10$	80.27	43	452.60	42	2158.52	39
WINQ		63.45	39	606.97	45	2797.03	45
PT+WINQ+SLACK		70.64	41	285.90	37	848.16	12
ECR-II $u = 0, k = 1.5$		10.96	6 *	34.35	2 *	929.51	17
ECR-II $u = 0, k = 2.0$		10.25	5 *	36.44	5 *	918.17	16
ECR-II $u = 0, k = 2.5$		10.09	2 *	39.66	9 *	950.40	18
ECR-II $u = 0, k = 3.0$		9.92	1 *	41.08	10 *	986.16	21
ECR-II $u = 1, k = 1.5$		11.80	8 *	33.10	1 *	792.77	10
ECR-II $u = 1, k = 2.0$		11.15	7 *	35.94	4 *	834.60	11
ECR-II $u = 1, k = 2.5$		10.22	4 *	37.42	6 *	914.58	15
ECR-II $u = 1, k = 3.0$		10.10	3 *	39.55	8 *	956.96	20

* refers to the best five rules

Figures

Figure 1. The partial schedule of the example problem

Figure 2. Normalized performance of the proposed rule and other existing rules in data set 1.

Figure 3. Normalized performance of the proposed rule and other existing rules in data set 2.

Figure 4. The effect of the job candidate reduction mechanism

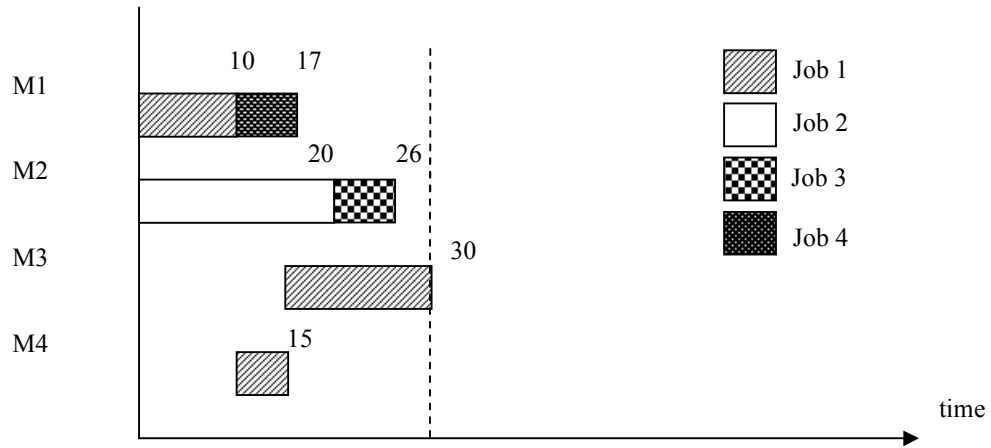
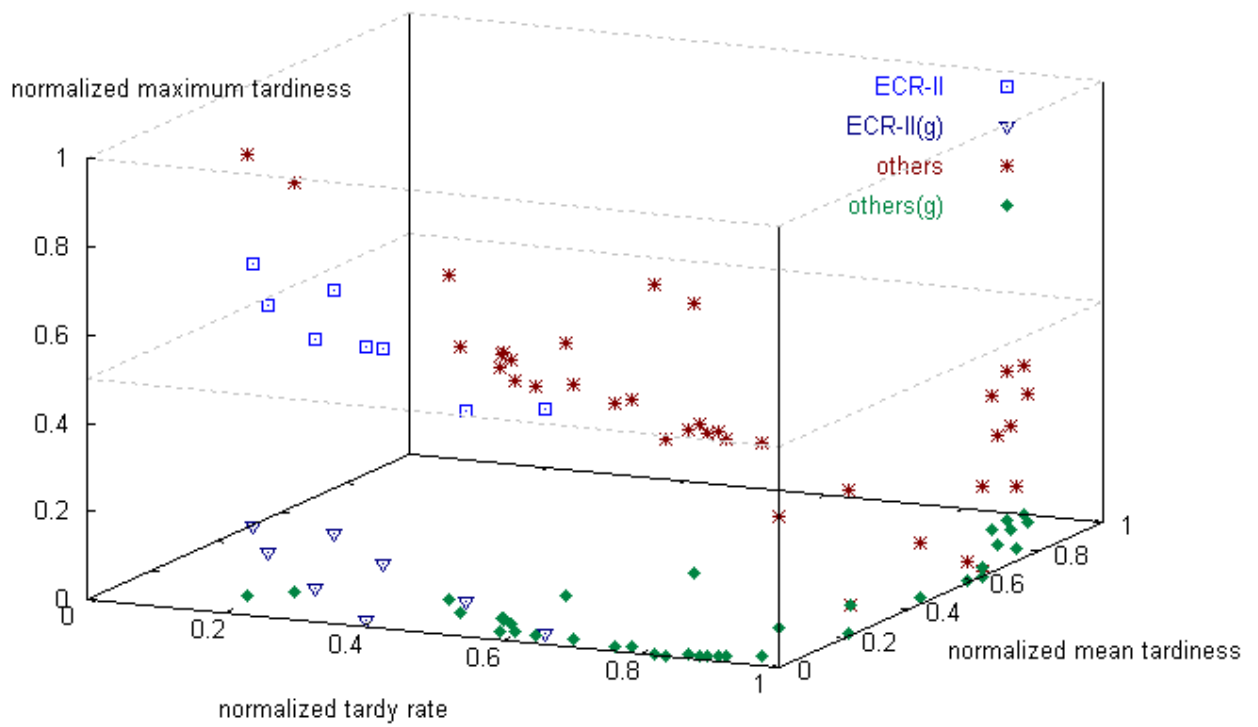
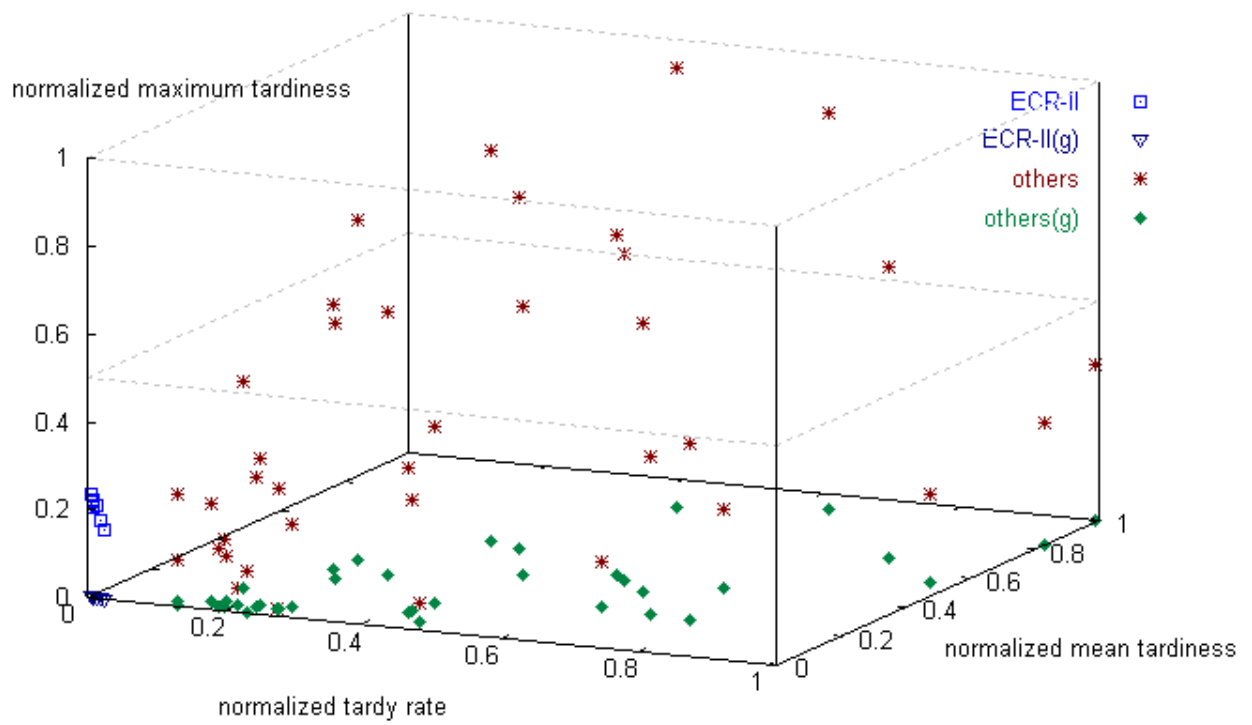


Figure. 1 The partial schedule of the example problem



(ECR-II(g)/others(g) are the data points of ECR-II/others projected to the ground plane, with normalized maximum tardiness as zero.)

Figure. 2 Normalized performance of the proposed rule and other existing rules in data set 1



(ECR-II(g)/others(g) are the data points of ECR-II/others projected to the ground plane, with normalized maximum tardiness as zero.)

Figure. 3 Normalized performance of the proposed rule and other existing rules in data set 2

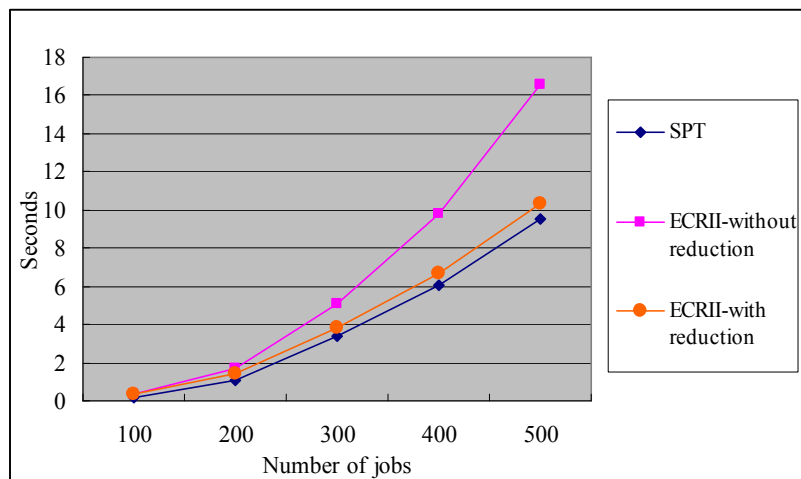


Figure. 4 The effect of the job candidate reduction mechanism