www.elsevier.com/locate/caor

Rule-based scheduling in wafer fabrication with due date-based objectives

Tsung-Che Chiang, Li-Chen Fu

PII: S0305-0548(12)00040-8
DOI: doi:10.1016/j.cor.2012.02.014
Reference: CAOR3031

To appear in: *Computers & Operations Research*

Received date: 18 April 2011
Revised date: 14 February 2012
Accepted date: 14 February 2012

Rule-based scheduling in wafer fabrication with due date-based objectives

Tsung-Che Chiang[a], Li-Chen Fu[b,c*]

[a]*Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiwan, R.O.C.*

[b]*Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, R.O.C.*

[c]*Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C.*

Abstract

Wafer fabrication is a capital-intensive and highly complex manufacturing process. In the wafer fabrication facility (fab), wafers are grouped as a lot to go through repeated sequences of operations to build circuitry. Lot scheduling is an important task for manufacturers to improve production efficiency and meet customers' requirements of on-time delivery. In this research we propose a dispatching rule for lot scheduling in wafer fabs, focusing on three due date-based objectives: on-time delivery rate, mean tardiness, and maximum tardiness. Although many dispatching rules have been proposed in the literature, they usually perform well in some objectives and bad in others. Our rule implements good principles in existing rules by means of (1) an urgency function for a single lot, (2) a priority index function considering total urgency of multiple waiting lots, (3) a due date extension procedure for dealing with tardy lots, and (4) a lot filtering procedure for selecting urgent lots. Simulation experiments are conducted using nine data sets of fabs. Six scenarios formed by two levels of load and three levels of due date tightness are tested for each fab. Performance verification of the proposed rule is achieved by comparing with nine benchmark rules. The experimental results show that the proposed rule outperforms the benchmark rules in terms of all concerned objective functions.

Keywords: Semiconductor manufacturing; Wafer fabrication; Scheduling; Dispatching rules; On-time delivery; Tardiness

* Corresponding author, full postal address: No. 1 Roosevelt Road, Sec. 4, Taipei, Taiwan 106, R.O.C.
Email addresses: tcchiang@ieee.org (Tsung-Che Chiang), lichen@ntu.edu.tw (Li-Chen Fu)

Nomenclature

A_i	time at which lot i is released into the fab
d_i	due date of lot i
e_i	number of due date extension of lot i
p_i	processing time of the imminent operation of lot i
P_i	sum of processing time of all operations of lot i
Q	queue of the station that becomes available
R_i	sum of processing time of the unfinished operations of lot i
r_i	time at which lot i arrives at the current station
s_{ij}	sequence dependent setup time required for processing lot j right after lot i
t	system time, the time at which the dispatching decision is to be made
Z_i	index value of lot i

1 Introduction

In recent years, the number of applications and demand for integrated circuits has increased dramatically. Microprocessors, memory chips, and other semiconductor devices are now a part of our daily lives, appearing everywhere in computer, communication, and consumer products. The semiconductor manufacturing process consists of four phases: wafer fabrication, wafer probe, packaging, and final testing. Among them, wafer fabrication is the most complex and costly one. In a wafer fabrication facility (fab), wafers are grouped and put into a container, usually called a *lot*. Each lot goes through repeated sequences of operations including diffusion, photolithography, etching, ion implanting, etc. to build up layers of circuitry on wafers. These operations are complicated and need high technology, and thus the equipment is usually very expensive. Gupta *et al*. [1] and Pfund *et al*. [2] mentioned that a large portion of capital cost in the wafer fab is due to the cost of manufacturing equipment. The high cost of equipment prohibits manufacturers from buying equipment and increasing manufacturing capacity unlimitedly, and it also forces production managers to utilize the equipment effectively in order to achieve high production efficiency (e.g. short mean cycle time) and meet customers' requirements (e.g. high on-time delivery rate). Hence, scheduling, which refers to the allocation of equipment over time to lots to optimize the concerned performance measures, becomes an important task in wafer fabs.

A wafer fab is usually viewed as a job shop with the following extensions:

(1) reentrant process flow: lots may visit the same station more than once;

(2) dynamic job arrival: customer orders may arrive at different times;

(3) parallel machines: more than one station is able to process one operation;

(4) batch processes: in some stages (e.g. diffusion) more than one lot can be processed simultaneously;

(5) sequence dependent setup: in some stages (e.g. ion implanting) a setup time is required between two operations with different recipes;

(6) machine failure: a station might not be available for an uncertain duration.

Since scheduling in a classical job shop is already known to be NP-hard for many performance measures like mean tardiness [3], scheduling in a wafer fab is much more difficult. The challenging complexity and practical value of fab scheduling attracted researchers in the academia and practitioners in the industry, and several kinds of scheduling approaches have been developed in the last decades [4]. Dispatching rules are one of the most popular approaches in the industry due to its ease of implementation, computational efficiency, convenience to deal with dynamic environments, and flexibility to incorporate domain knowledge and expertise. It has been applied for fab scheduling successfully by many real-world companies, including Siemens [5], Motorola [6], Samsung [7], IBM [8], and Agere Systems (now in LSI corporation) [9].

With Application Specific Integrated Circuit (ASIC) and specialty processors gaining more and more market share, the capability of meeting due dates is becoming a critical factor in the low-volume, high-variety, and make-to-order wafer fabs. Several studies indicated that today's wafer fabs have been forced to become increasingly conscious of their due date delivery performance [1][2]. In this study, we aim at developing a dispatching rule for the scheduling of wafer fabs with respect to due date-based measures. The proposed rule is distinguished from the existing ones in the use of group information of competing lots and a due date extension procedure. Its performance is verified by simulation experiments using nine fab models and nine benchmark rules in the literature. The concerned performance measures include on-time delivery (OTD) rate, mean tardiness, and maximum tardiness. They are defined by

$$\text{OTD rate} = \frac{1}{|N|}\sum_{i\in N} U(i), \quad U(i) = \begin{cases} 0, & if\ C_i > d_i \\ 1, & if\ C_i \le d_i \end{cases}, \tag{1}$$

$$\text{mean tardiness} = \frac{1}{|N|}\sum_{i\in N} \max\{C_i - d_i, 0\}, \tag{2}$$

$$\text{maximum tardiness} = \max_{i\in N} \max\{C_i - d_i, 0\}, \tag{3}$$

where N denotes the set of finished lots, and C_i and d_i denote the completion time and due date of a job i, respectively. The rest of this paper is organized as follows. Section 2 gives a review of related work, and Section 3 details the proposed rule. The simulation model and experimental setting are presented in Section 4. Experimental results and discussion are provided in Section 5. Finally, Section 6 gives the conclusion and future research directions.

2 Literature review

A dispatching rule is usually a simple mathematical equation or a short algorithm for calculating priority indices for jobs. It is often invoked when a station finishes a job and becomes available to process the next job. The rule assigns priority indices for waiting jobs, and the job with the highest priority (sometimes the highest index value and sometimes the lowest index value, depending on the rule) is taken as the next processing target. Some reviews of dispatching rules can be found in Panwalker and Iskander [10], Blackstone *et al*. [11], Rajendran and Holthaus [12], Jayamohan and Rajendran [13], and Sarin *et al*.

[14]. The key points in designing a dispatching rule is what attributes are included and how they are combined if there are more than one attribute.

Early research studies on fab scheduling were usually concerned about cycle time-based performance measures. The first way to design rules for fab scheduling is to borrow ideas from rules for classic flow shop or job shop scheduling. Inspired by the classic least slack (SLACK) rule, Lu *et al*. [15] proposed the FSVCT and FSMCT rules. The FSVCT rule is like the SLACK rule but replaces the term of due date with the lot arrival time. In this way, the cycle time is equal to the lateness, and hence the FSVCT rule can reduce the variation of cycle time just like the SLACK rule can reduce the variation of lateness. The FSMCT rule is a little more complex. It aims at reducing the burstiness of arrivals of lots to each buffer and consequently reducing the mean cycle time. Based on the idea of the operation due date (ODD) rule, Yoon and Lee [16] developed a rule by allocating the desired cycle time to operations according to the utilization rate of the corresponding stations and then assigning due dates to operations. The lot with the earliest due date of the imminent operation is the next one to be processed. Their rule outperformed the rule proposed by Lu *et al*. in terms of standard deviation of cycle times. Bahaji and Kuhl [17] proposed four rules based on two rules proposed by Rajendran and Holthaus [12]. They introduced the X factor, the ratio of the accumulated flow time to the sum of the processing time of completed operations, in the new rules. Considering mean cycle time and OTD rate, their recommended rule is the one which combines shortest-processing-time rule (SPT), lowest-work-in-next-queue rule (WINQ), and the X factor.

With Little’s Law in queueing theory [18], several researchers noticed the relationship between cycle time and inventory (or work-in-process, WIP) and devised rules based on WIP information. Li *et al*. [19] proposed the MIVS rule by introducing correlation between inter-arrivals and services to reduce variability. The rule separates lots into four classes according to the deviation of current inventory from the average inventory. The basic principle is to expedite the lots with excessive inventory at the current stage and to postpone those with excessive inventory at the downstream stage. Lee *et al*. [20] addressed lot release and lot scheduling by two push-type and two pull-type rules. The push-type rules calculate the deviation of current WIP level from the planned WIP level for each layer of each device and process the lots of layers with more excess WIP level at higher priority. The pull-type rules calculate a due date for each layer of each device by averaging the sum of due dates of lots at the layer over the current WIP level. Then, this layer due date is used in the SLACK rule and ATC rule [21]. By comparing the two pull-type rules with two push-type rules, the pull-type rules showed better performance on several performance criteria including cycle time and machine utilization. Duwayri *et al*. [9] proposed a rule for balancing workload levels of lots at different layers. It calculates the workload index of a layer l by the ratio of current workload of the bottleneck stage of l to total processing time of operations in l. Then, the lots of the layer with the maximum workload index have the highest priority to be processed. This rule outperformed the first-come-first-serve (FCFS) rule and the earliest-due-date (EDD) rule with respect to mean cycle time and WIP level.

In addition to adapting classic rules to be used in fabs, another way to improve the performance of rules is to do estimation of time attributes in rules more accurately. The remaining processing time is a common and critical attribute in many rules and has been investigated by several researchers. Hung and Chen [22] devised a dynamic look-ahead rule, which predicts the remaining flow time of each lot by simulation. The lot with the smallest ratio of the predicted remaining flow time to the number of remaining operations is taken as the next processing target. Kim *et al*. [23] estimated the waiting times of lots at the photolithography stations by assuming that lots are processed in the EDD order. Kim *et al*. [24] computed the estimated waiting time of a lot i at a bottleneck station by the product of the average WIP level on the station, the average processing time on the station, and the number of times the lot i needs to visit the station. Chen [25] improved the FSVCT and FSMCT rules by Lu *et al*. [15] through making them nonlinear versions. He estimated remaining processing time of lots by fuzzy c-means method and the fuzzy back propagation network. The improved rules showed much better performance than five existing rules in terms of mean cycle time and standard deviation of cycle times.

As the importance of on-time delivery performance was realized by semiconductor manufacturers, researchers also started to study dispatching rules for due date-based measures like OTD rate and mean tardiness. Kim *et al*. [23] proposed several dispatching rules in order to minimize mean tardiness. These rules use much lot information including the number of remaining layers, estimated waiting time, and total processing time of unfinished photolithography operations. Later, they extended their research by considering batch scheduling [24]. Rose [26] showed that determining appropriate due date is critical to the critical ratio (CR) rule when OTD rate is concerned. He also compared several due date-based dispatching rules and found that the ODD and CR rules perform well with respect to OTD rate when the target flow factor is close to the average flow factor under the FCFS rule [27]. Li *et al*. [28] proposed a rule for improving OTD rate. In default, the rule works as the ODD rule does. The rule adjusts the priorities of lots when there is low WIP on bottleneck stations or high WIP on non-bottleneck stations. Wu *et al*. [29] developed a modified ODD rule. They combined the ODD rule with the SPT rule, where the flow time of the imminent operation is estimated by the processing time plus a multiple of the standard deviation of the flow time of this operation. The proposed rule showed better performance than several classic rules including CR and EDD in terms of OTD rate and total tardiness.

Setup is an important issue in wafer fabs. Chern and Liu [30] examined the "family-based" concept to deal with setup time on steppers (stations in the photolithography stage). This concept intends to save the long setup time caused by changing masks and keeps processing the lots belonging to the same product family until there is not such lot in the queue. They tested five family-based dispatching rules and found that the family-based concept was beneficial to reduce cycle time and increase throughput. Lee and Pinedo [31] improved the ATC rule [21] to be the ATCS rule by incorporating setup information. Kang *et al*. [32] modified the ATCS rule to be the RATCS rule by considering the incoming lots from upstream stations. The RATCS rule showed lower total weighted tardiness than the SLACK and EDD rules did. Pfund *et al*. [33] proposed the ATCSR rule, which was also based on the ATCS rule. The difference between the RATCS and ATCSR rules is in the way they penalize the machine idle time for waiting the incoming lots.

Some researchers studied the combination of existing rules. One way to combine rules is to select different rules for different states and/or stations. Chen *et al*. [34] developed a state-dependent rule, which selects among three existing rules according to machine utilization and queue length. The dynamic selection of rules achieved better performance than each individual rule regarding cycle time and WIP. Miragliotta and Perona [35] divided the stations into six groups based on machine utilization, operation type (serial or batch), and requirement of setup. Each group was assigned an appropriate rule. Wu *et al*. [36] classified stations into dedicated steppers, non-dedicated steppers, and others. Rules for steppers adopt the family-based concept in Chern and Liu [30]. For dedicated steppers, their rule selects the lot family based on the line-balancing principle; for other stations, their rule selects the lot family based on the starvation-avoidance principle. After a lot family is selected, a lot is then selected by the CR rule. Zhang *et al*. [37] classified lots into four groups based on whether they are hot lots, whether their next visiting station is a bottleneck, and how long the length of queue in the next visiting machine is. A distinct combination of rules is designated to each group of lots. Another way to combine rules is through weighted summation of priority indices calculated by multiple rules. The main difference between the relevant studies is in the approach to set the weights of rules. For example, Dabbas *et al*. [6] used the response surface method; Min and Yih [38] used the neural network; Sivakumar and Gupta [39] set weights by human experts.

When more computational budget is available, performance of dispatching rules can be further improved by sophisticated approaches. Metaheuristics such as genetic algorithms (GAs) are a popular approach to production scheduling. Due to the large scale of wafer fabs, it is difficult to build the detailed scheduling by only metaheuristics. One promising way is to optimize the use of dispatching rules by metaheuristics. For example, Sha and Liu [40] relied on the simulated annealing algorithm to search for the optimal combination of order release, dispatching, and rework rules. Liu and Wu [41] sought for the proper combination of rules in different time intervals by the GA. Authors of this study adopted the GA [42] to optimize weights of dispatching rules for calculating the aggregated priority indices of lots. Shifting bottleneck (SB) procedure [43] is another sophisticated approach to classical job shop scheduling.

It decomposes the multi-stage scheduling problem into multiple single-stage single/parallel machine scheduling sub-problems and solves the sub-problems one by one. Upasani *et al*. [44] used the SB to minimize maximum lateness in the wafer fab. Heavily-loaded stations were scheduled by a branch-and-bound algorithm, and lightly-loaded stations were scheduled by a dispatching rule. This work was extended by Sourirajan and Uzsoy [45], where parallel machines and batch machines were included. Pfund *et al*. [46] also investigated how the SB can schedule the wafer fab. They used their own dispatching rule, ATCSR [33], as the sub-problem solution procedure. Mönch *et al*. [47] presented an approach combining the dispatching rule, GA, and SB. The main flow of their approach was based on the SB. The dispatching rule and the GA [48] were used to schedule the non-critical and critical stations, respectively.

As we can see from the literature review, dispatching rule is a popular tool for fab scheduling and developing rules is an important research topic. Although some studies have shown the potential of automatic combination [40]–[42] and construction of rules [49][50], there are still some limitations. For example, the computational requirement of GA and genetic programming (GP) is large, and the interpretation of the rules evolved by GP is not straightforward. Thus, we think that researches on these different directions should be conducted in parallel and complement one another. In our previous study [51] we found that the performance of the original rules has a large impact on the performance of the combined rule. If we can design better rules based on domain knowledge, the performance of the automatic rule combination approach will also get improved. Besides, the (sub-)expression in the rule developed by domain knowledge can serve as effective components in the GP-based approaches to construct new rules. In the literature, dispatching rules usually prioritize lots based on individual information such as processing time and due date, and tardy lots are often prioritized simply by the SPT rule. These two traditional thoughts could decrease the rule performance. In this study, we propose a rule that considers the impact of processing of a lot on other lots and deals with tardy lots with better logic. The rationale and details of the proposed rule is given in the following section.

3 The proposed ECR3 dispatching rule

In this paper, we propose a rule named Enhanced Critical Ratio 3 (ECR3), whose name indicates that it experiences two times of refinement. Its first version was proposed in [52], in which this rule focused only on maximizing OTD rate and showed its superiority over five benchmark rules. Then, the second version ECRII was presented with several improvements in [53], where it demonstrated better performance for OTD rate and mean tardiness than eighteen existing rules in the classical job shop environment. In this study the ECRII rule is further enhanced to be ECR3, whose goal is to provide better performance than existing rules for OTD rate, mean tardiness, and maximum tardiness in complex job shops such as wafer fabs.

3.1 Basic form

To deal with due date-based objectives, dispatching rules in the literature usually assign index values to the waiting jobs based on their degrees of urgency estimated by the remaining processing time (R_i), the allowance time ($d_i - t$), the slack time ($d_i - R_i - t$), or some combinations of them. The main idea through ECR to ECR3 is to select the next processing target so that the sum of degrees of urgency of all waiting jobs is kept minimal after the selected job is processed. Different from most existing rules, which assign the index value to a lot based only on its individual information, all versions of ECR assign the index value to a lot considering both its own information and its influence on other competing lots. This is the most important feature that distinguishes the ECR rules from others.

In our opinion, the degree of urgency should gradually decrease as operations of a lot are finished and should gradually increase as its allowance time is consumed. In addition, the increasing rate should become higher and higher as the due date is approaching. Accordingly, we use the square of ratio of the remaining processing time to the allowance time as the measure of degree of urgency. The basic equation of ECR3 is given as follows. Among all waiting lots, the lot with the smallest Z value defined below will be selected first, where the lot k denotes the last lot being processed on the station:

$$Z_i = urg\,(R_i - p_i, d_i - s_{ki} - p_i - t) + \sum_{j \in Q, i \neq j} urg\,(R_j, d_j - s_{ki} - s_{ij} - p_i - t) \tag{4}$$

with

$$urg(R,a) = \begin{cases} 0 & , R = 0 \wedge a \geq 0 \\ (R/a)^2 , & a \geq R > 0 \\ 1 & , \quad R > a \end{cases} . \tag{5}$$

There are two terms in (4). The first term evaluates the degree of urgency of the selected lot, whereas the second term evaluates the sum of degrees of urgency of competing lots, both after the selected lot is processed. In the urgency function *urg*(*R*, *a*), *R* refers to the remaining processing time and *a* means the allowance time. By choosing the lot with the smallest *Z* value, the ECR3 rule picks the lot with higher degree of urgency earlier so that its urgency will not keep increasing; meanwhile, the lot with shorter processing time is also favored since its processing will not cause much increment on the degrees of urgency of other lots. Since the sequence dependent setup (SDS) time is not uncommon in the wafer fabrication processes, the ECR3 rule also takes the SDS time into account and reflects this factor in calculating the allowance time. In this way, the ECR3 rule will prefer the lot that needs short setup time. Here, an example is given to show how the ECR3 rule works. Assume there are three lots in the queue. The relevant information is summarized in Table 1. The ECR3 rule assigns the index values to these lots as follows. The system time (*t*) is assumed to be 1.

<< Insert Table 1 about here >>

$Z_1 = urg(10 – 5, 30 – 0 – 5 – 1) + urg(20, 30 – 0 – 2 – 5 – 1) + urg(40, 50 – 0 – 2 – 5 – 1) \cong 1.7769$

$Z_2 = urg(20 – 4, 30 – 2 – 4 – 1) + urg(10, 30 – 2 – 2 – 4 – 1) + urg(40, 50 – 2 – 2 – 4 – 1) \cong 1.6625$

$Z_3 = urg(40 – 2, 50 – 4 – 2 – 1) + urg(10, 30 – 4 – 4 – 2 – 1) + urg(20, 30 – 4 – 4 – 2 – 1) \cong 2.0580$

The original degrees of urgency of lots 1, 2, and 3 are $(10/(30–1))^2$, $(20/(30–1))^2$, and $(40/(50–1))^2$, respectively. Lots 2 and 3 have higher degrees of urgency than lot 1 does. Although lot 2 has longer processing time than lot 3 ($p_2 = 4 > p_3 = 2$), the shorter setup time ($s_{21} = s_{23} = 2 < s_{31} = s_{32} = 4$) makes lot 2 a better choice. (The total degree of urgency of all three lots after processing lot 3 is 2.058, but that of all three lots after processing lot 2 is only 1.6625.) Therefore, the ECR3 rule selects lot 2 in this example.

3.2 Due date extension

According to the urgency defined in (5), the degrees of urgency of tardy lots, including those that are expected to be tardy ($t + R_i > d_i$) and those that are already tardy ($t > d_i$), are fixed as one. That makes it difficult to evaluate the variation of degrees of urgency of these lots and thus makes them indistinguishable under the ECR rule. In ECRII, a due date extension procedure was proposed so that the variation of degree of urgency of the tardy lot can be evaluated. The idea is to internally extend the due date of a tardy lot before calculating its index value. In this procedure, two attributes e_i and $d_i^{e_i}$ are introduced for each lot i. The attribute e_i refers to the times of due date extension, and $d_i^{e_i}$ refers to the extended due date by the e_i^{th} extension. Note that $d_i^{e_i}$ is only used inside the dispatching rule. The original due date d_i is retained and used when performance measures such as OTD rate are calculated. For each lot i, the initial value of e_i is set to zero and d_i^0 is set to d_i. After introducing the due date extension procedure, the formula of ECR3 rule is modified to involve e_i and $d_i^{e_i}$ as follows:

$$Z_i = urg\,(R_i - p_i, d_i^{e_i} - s_{ki} - p_i - t, e_i) + \sum_{j \in Q, i \neq j} urg\,(R_j, d_j^{e_j} - s_{ki} - s_{ij} - p_i - t, e_j) \tag{6}$$

with

$$urg(R,a,e) = \begin{cases} 0 & , R = 0 \wedge a \geq 0 \\ (e+1)\cdot(R/a)^2 , & a \geq R > 0 \\ (e+1) & , \quad R > a \end{cases} . \tag{7}$$

Comparing the urgency functions in (5) and (7), the difference is that the degree of urgency in (5) is amplified by (e+1) times in (7) so as to raise the degrees of urgency of the lots experiencing due date extension.

One remaining issue in the due date extension procedure is how long the due date is to be extended. In

ECRII, the due date was extended by a multiplier of the remaining processing time of the lot, namely

$$d_i^{e_j} = d_i^{e_j - 1} + \alpha \cdot R_i \quad (8)$$

This method makes it easy to use α to control the degree of urgency right after due date extension. However, a problem rises when we focus on minimizing maximum tardiness – the times of due date extension (e_i) is not directly related to the amount of tardiness ($t + R_i - d_i$). In ECRII, the due date of a tardy lot with little remaining workload is extended by a little amount in each extension. It implies that this kind of lot may experience another due date extension in a short period, and its degree of urgency could increase quickly due to the fast increasing of e_i. On the contrary, a tardy lot with large remaining workload receives due date extension infrequently, and its degree of urgency increases relatively slower. Therefore, the ECRII rule could select a tardy lot with little remaining workload (and large e_i) instead of a tardy lot with large remaining workload (and small e_i) even though the latter lot has experienced much longer tardiness than the former one has.

To deal with this problem in ECRII, the equation for due date extension is modified to be

$$d_i^{e_j} = \min\{ d_i^{e_j - 1} + Y_1, t + (1+Y_2) \cdot R_i \} \quad (9)$$

in ECR3. The first term indicates that the amount of due date extension is fixed as a constant Y_1. In this way, the times of due date extension e_i of a lot i can closely reflect its amount of tardiness (since the tardiness is about $Y_1 \cdot e_i$). Consequently, preferring the lots with larger e_i becomes a reasonable strategy in ECR3 when maximum tardiness is to be minimized. Sometimes, the remaining processing time of a tardy lot could be much smaller than Y_1. In this condition, the degree of urgency could become small after due date extension, making a tardy lot look less urgent. Hence, a lower bound is given to fix this potential problem. This is the purpose of introducing the second term in (9). When the due date is extended by the second term, the degree of urgency (the square of the ratio of the remaining processing time to the allowance time, defined in Section 3.1) becomes $(R_i/((1+Y_2) \cdot R_i))^2 = 1/(1+Y_2)^2$. We can control the lower bound of the degree of urgency by the value of parameter Y_2. Values of parameters Y_1 and Y_2 are given in

Section 4.

In addition to the equation for due date extension, we make another modification in the urgency function to improve the performance on minimizing maximum tardiness. To calculate the Z_i value of a lot i with only one unfinished operation by ECRII, the degree of urgency of lot i itself is defined by e_i if lot i can be finished within the current (extended) due date and is defined by (e_i+1) if lot i cannot be finished in time. In ECR3, we change the degree of urgency of lot i in the first case from e_i to zero. Setting the degree of urgency as zero makes completion of the whole fabrication process of a tardy lot an attractive option for ECR3. This strategy aims to stop the rising of tardiness caused by the tardy lot, particularly useful when the lot is the one that causes maximum tardiness in the fab.

3.3 Two viewpoints for calculation of total degree of urgency

As mentioned, the main idea through ECR to ECR3 is to select the next processing target so that the total degree of urgency is kept minimal after the selected lot is processed. When evaluating the "total degree of urgency", there are two viewpoints – to consider or not to consider the degree of urgency of the selected lot. The former viewpoint intends to select a lot such that processing of its imminent operation can effectively reduce its own degree of urgency and does not raise the degrees of urgency of other competing lots too much. This viewpoint was followed by ECRII. On the other hand, the latter viewpoint focuses on reducing the sum of degrees of urgency accumulated on the station. Following this thought, the degree of urgency of the selected lot after its imminent operation is finished is considered as zero in the priority index function. In preliminary tests we found that both viewpoints result in good performance in some cases. Therefore, we introduce a parameter Y_3 to make both points of view realizable in the ECR3 rule. The value of Y_3 can be zero or one. The following is the final form of ECR3.

$$Z_i = Y_3 \cdot urg\,(R_i - p_i, d_i^{e_i} - s_{ki} - p_i - t, e_i) + \sum_{j \in Q, i \neq j} urg\,(R_j, d_j^{e_j} - s_{ki} - s_{ij} - p_i - t, e_j) \tag{10}$$

with

$$urg(R,a,e)=\begin{cases} 0 & ,R=0 \wedge a \geq 0 \\ (e+1)\cdot(R/a)^2, & a \geq R > 0 \\ (e+1) & , \quad R > a \end{cases} . \tag{11}$$

3.4 Lot filtering

When there are several lots waiting in the queue, the ECR3 rule may select a relatively less urgent lot instead of a very urgent one if the processing time of imminent operation of the former one is much shorter than that of the latter one. (In that condition, processing of the former lot increases the degrees of urgency of other competing lots by a much smaller amount than the latter one does.) To solve this problem, when we detect the condition in which there is a large difference of degrees of urgency among the waiting lots, a lot filtering procedure is activated. Only the lots passed the filtering procedure are assigned the index values by (10), and the one with the lowest index value is the next processing target. The algorithm of the filtering procedure is shown in Algorithm 1. The main idea is to filter out the lots that are not tardy and whose degrees of urgency are lower than the average degree of urgency over all waiting lots. The variable Y_4 in this procedure is also a parameter of ECR3.

<< Insert Algorithm 1 about here >>

4 Simulation model, experimental setting, and benchmark rules

4.1 Fab model

There are N_P products, and each product is associated with one of N_R processing routes. Each processing route is defined as a sequence of operations, and each operation is designated to be processed on a certain group of stations. There are N_E groups of stations, each consisting of at least one station.

In general, there are three types of operations, by-wafer, by-lot, and by-batch operations. Each operation has a step ID, and the by-batch operation could also have a batch ID. Only lots whose imminent operations have the same step ID or batch ID can be batched together. For each by-batch operation, the minimum and maximum batch sizes, B_m and B_M, are predefined. The by-batch operation can start only if the number of wafers of waiting lots in at least one batch is not less than B_m, and at most B_M wafers can be processed at a time. Processing time of a by-wafer operation is proportional to the lot size, and processing time of a by-batch operation depends not only on the lot size but also the maximum batch size.

In addition to the step ID, an operation may also have a setup group ID. When a station starts to process a new operation whose step ID or group ID is different from that of the previous operation, a specification setup or group setup is required. In our current model, the processing time and setup time are deterministic, which was indicated as a reasonable assumption by Sourirajan and Uzsoy [45].

Lots are released into the fab with a constant time interval. Each product has a distinct inter-arrival interval and a lot size (number of wafers). The time between machine breakdown and time to repair are assumed to follow the exponential distribution. Each group of stations has its own mean time between failure (MTBF) and mean time to repair (MTTR).

The dispatching rule is invoked each time when a serial-type station finishes an operation. As for scheduling on batch-processing stations, which is not the focus in this study, the batch containing the largest number of wafers is selected as the next processing target. Ties are broken by the EDD rule. Transportation, human operators, and rework are not considered.

4.2 Experimental setting

Nine data sets of fabs were taken in the simulation experiments, including seven MIMAC data sets from Fowler and Robinson [54], one SEMATECH data set from Campbell and Ammenheuser [55], and one data set from Sourirajan and Uzsoy [45]. Their scales are summarized in Table 2.

The lot release rate was controlled to make the utilization of bottleneck stations around 90%. We defined these scenarios as “heavy” load scenarios. In order to examine the performance of rules under different load levels, a duplicate set of experiments was conducted with the release rates set to those in heavy load scenarios times 90%. These scenarios are defined as “moderate” load scenarios. To set due dates, we first calculated the flow factor, which is defined as the ratio of average cycle time to the raw total processing time, of each product in the tested fab under the FCFS dispatching rule. Then, the average flow factor (*FF*) over all products was calculated. Finally, the TWK (Total WorK content) method was used to set due dates. Here we created three types of scenarios, standing for "tight," "moderate," and "loose" due dates, respectively. The due date of each lot i is set to $P_i \cdot U[1, 2 \cdot FF-1]$ in tight due date scenarios, $P_i \cdot U[(1+FF)/2, (3 \cdot FF-1)/2]$ in moderate due date scenarios, and $P_i \cdot U[FF, 2 \cdot FF-1]$ in loose due date scenarios. $U[a, b]$ is a function which generates a real number uniformly distributed in the interval $[a, b]$. Figure 1 shows the ranges of due dates in the three types of scenarios, and the values of *FF* for all nine fabs are given in Table 2.

The warm-up period was set to 180 days based on the observation of the curves of average cycle time and WIP level. We used the batch means method [56] to collect the simulation output data. Twenty batches were collected, with each contained data of 180 days. The Common Random Numbers (CRN) technique was used as a variance reduction technique.

<< Insert Figure 1 and Table 2 about here >>

4.3 Benchmark rules

Many dispatching rules have been proposed in the literature. In the experiments we selected nine rules to be compared with our proposed rule. When doing dispatching, the COVERT, ATCSR, and RACTS rules select the lot with the largest Z value as the next processing target while the other rules select the lot with the smallest Z value. In case of a tie, the FCFS rule is used to determine the next target.

FCFS: The FCFS rule is a common reference rule when evaluating the performance of dispatching rules [30][35][57]. It assigns the priority index value by

$$Z_i = r_i. \quad (12)$$

EDD: The EDD rule is one of the earliest rules focused on due date-based objectives. In the literature, its major advantage is shown on minimizing maximum tardiness [13][58][59][60]. It assigns the priority index value by

$$Z_i = d_i. \tag{13}$$

SLACK: The SLACK rule prefers the lots with earlier due dates and longer remaining processing time. This rule was reported as a good one for minimizing maximum tardiness in our previous studies [53][60]. It assigns the priority index value by

$$Z_i = d_i - t - R_i. \tag{14}$$

CR: The CR rule is a simple ratio-based dispatching rule. It favors the lots with shorter allowance time and longer remaining processing time when the allowance time is positive. After the allowance time becomes negative, it prefers the lots with shorter remaining processing time. It is commonly used in the semiconductor manufacturing industry [2][6][38]. By this rule, the priority index value is given by

$$Z_i = (d_i - t)/R_i. \tag{15}$$

COVERT: The Cost OVER Time (COVERT) rule [62] is one of the most widely used dispatching rule focused on due date-based objective functions. It favors the lots with earlier due dates, longer remaining processing time, and shorter processing time of the imminent operations. By combining these principles, it was often reported to perform well for due date-based objectives, especially for mean tardiness [24][58][59][60][61]. It assigns the priority index value by

$$Z_i = (1/p_i)\cdot[1 - (d_i - t - R_i)^+/(k\cdot R_i)]^+ \tag{16}$$

where k is its parameter and $(v)^+$ means $\max\{v, 0\}$.

OPDD: Wu *et al*. [29] proposed to use an operation due date-based dispatching rule for scheduling make-to-order (MTO) lots in the hybrid make-to-stock (MTS)/MTO fab. It assigns a due date for each operation e of lot i by

$$OPDD_{i,e} = A_i + (d_i - A_i)\cdot\left(\sum_{k=1}^{e} p_{i,k} / P_i\right), \tag{17}$$

where $p_{i,k}$ denotes the processing time of operation k of lot i. Then, the priority index of a lot i whose imminent operation is e is calculated by

$$Z_i = \mathrm{OPDD}_{i,e} - p_i - \beta \cdot \sigma_{i,e}, \tag{18}$$

where $\sigma_{i,e}$ denotes the standard deviation of the flow time of operation e of lot i and β is a parameter of this rule. In our experiments, $\sigma_{i,e}$ was collected and updated every 180 days.

ATCSR: Based on the ATC [21] and ATCS [31] rules, Pfund *et al*. [33] proposed the ATCSR rule. Like the COVERT rule, it favors lots with earlier due dates, longer remaining processing time, and shorter processing time of the imminent operations. Besides, it considers setup time and incoming lots. Its priority index function is

$$Z_i = \frac{1}{p_i}\exp\left(-\frac{(d_i - R_i - \max(r_i, t))^+}{k_1 \bar{p}}\right)\exp\left(-\frac{s_{li}}{k_2 \bar{s}}\right)\exp\left(-\frac{(r_i - t)^+}{k_3 \bar{p}}\right) \tag{19}$$

where r_i is the time at which the lot arrives at the station (r_i is greater than t for incoming lots from upstream stations), l is the last lot processed on the station, and $\bar{p}$ and $\bar{s}$ are average processing time and average setup time, respectively. ATCSR uses three parameters k_1, k_2, and k_3 to adjust the relative importance between lot urgency, setup overhead, and incoming lots.

RATCS: Kang *et al*. [32] developed the RATCS rule, which is also based on the ATCS rule and is similar to the ATCSR rule. The difference between ATCSR and RATCS is in that RATCS includes the time waiting for incoming lots in the setup time. Its priority index function is

$$Z_i = \frac{1}{p_i}\exp\left(-\frac{(d_i - R_i - t)^+}{k_1 \bar{p}}\right)\exp\left(-\frac{s_{li} + (r_i - t)^+}{k_2 \bar{s}}\right) \tag{20}$$

WPWX: Bahaji and Kuhl [17] proposed four rules and recommended the Wt(PT+WINQ)/XF rule, hereafter abbreviated as the WPWX rule. It favors the lots with shorter processing time, shorter queue at downstream stations, and larger flow factor (X factor, XF). The XF_i of a lot i is calculated by

$$XF_i = (t - A_i)/(P_i - R_i). \tag{21}$$

The priority index function is defined by

$$Z_i = \exp(-XF_i)\cdot((p_i + w_i)/XF_i) + \exp(XF_i)\cdot(1/XF_i) \tag{22}$$

where w_i denotes the sum of processing time of lots at the downstream stations of lot i.

4.4 Parameter setting

Among the ten tested rules, we have to determine values of parameters for the COVERT, OPDD, ATCSR, RATCS, and ECR3 rules. For COVERT, we tested ten variants with the parameter k setting to values from 1 to 5.5 in increment of 0.5, based on the values used in the literature (e.g. 0.25~2 in [58], 0.5 and 1 in [62], 1 in [24], and 4 in [59]) . In the original paper of OPDD, the authors set the parameter β to 0.5. In our experiments, we tested ten variants with the parameter β setting to {0, 0.125, 0.25, 0.5, 1, 2, 3, 4, 5, 10}, covering a wide range of values. In the original paper of ATCSR, the authors tested 3146 combinations of parameter values. Here we did not test such a large number of combinations since it could take too much computation time and may not be practical. We tested 64 ($4\cdot4\cdot4 = 64$) variants of ATCSR with k_1 setting to {0.01, 0.1, 1, 10} and k_2 and k_3 to {0.00001, 0.0001, 0.001, 0.01}, trying to include a wide range of possible values. Since RACTS is similar to ATCSR and the roles of k_1 and k_2 are the same in both rules, we tested the same number of combinations by selecting candidate parameter values from roughly the same range. We also tested 64 ($8\cdot8 = 64$) variants of RACTS with k_1 set to {0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10} and k_2 to {0.000005, 0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01}. For ECR3, we tested 32 ($4\cdot2\cdot2\cdot2 = 32$) variants by setting Y_1 to {5, 10, 20, 40}, Y_2 to {0.3, 0.4}, Y_3 to {0, 1}, and Y_4 to {0.3, 0.4}. More discussion on the selection and effect of parameter values of ECR3 will be given in Section 5.5.

The advantage of the group setup policy was discussed by Benjaafar and Sheikhzadeh [63] and by Chern and Liu [30]. The group setup policy allows setup actions only when there is no waiting lot requiring the current setup setting. Although much setup time can be saved by following the group setup policy, it is not always beneficial for the due date-based objectives [64]. Hence, we tested two variants (with and without the group setup policy) of each rule with each distinct parameter setting. In other words, we have 20 (10 parameter settings × 2 setup policies) variants for OPDD and COVERT, 128 variants for

ATCSR and RACTS, 64 variants of ECR3, and 2 variants for remaining five rules. We have 370 rule variants in total.

5 Experimental results

In the experiments, three due date-based objective functions including OTD rate, mean tardiness, and maximum tardiness were considered. They are defined in equations (1)–(3). Given two levels of load and three levels of due date tightness, we have six scenarios for each fab. In each scenario, we identified the best group of rules. The results in terms of the three objective functions are presented in the following three subsections, respectively. The last two subsections will give discussions on the design principles and parameter values of ECR3.

To identify the best group of rules, first we calculated for each rule variant the average objective values over twenty batches. (We used batch means method for data collection, as mentioned in Section 4.2). Regarding each objective function, the variant with the best performance among 370 rule variants (generated from ten main rules) was identified. Then, the paired *t*-test [56] was conducted to see if each of the other 369 rule variants is statistically different from the best one, with 95% confidence level. If any rule variant is not statistically different from the best rule variant, we put its corresponding main rule in the best group of rules. We counted the number of fabs in which a main rule is recognized in the best group for each scenario. The results are summarized in Table 3 and Figure 2-4. For each objective function, the best three main rules are marked by gray color in Table 3. We also provide the average objective values of the best variant of each main rule in terms of OTD rate, mean tardiness, and maximum tardiness in Table 4–6 for reference. The cell of a main rule is marked by gray color if the rule is in the best group of rule.

<< Insert Table 3 and Figure 2-4 about here >>

5.1 OTD rate

Given nine fabs and six scenarios, the maximum number of times of being recognized in the best group is 54. In Table 3, the proposed ECR3 rule is recognized in the best group for 32 times and outperforms all nine benchmark rules. The second and third best rules are OPDD and COVERT, which are recognized for 29 and 18 times, respectively. In the literature, rules realizing the "shortest-processing-time-first" principle usually perform well on maximizing OTD rate [13]. The ECR3, OPDD, and COVERT rules all implement this principle.

The performance difference among the tested rules is more significant when due dates are tighter. When due dates are loose, eight rules belong in the best group at least once; when due dates are tight, only four rules are in the best group. The effect of load level is relatively smaller than that of due date tightness. In some scenarios (e.g. moderate due dates and moderate load level in fab 3 and fab 4) the OPDD or WPWX rule is the only rule in the best group and outperforms the second best rule greatly. However, this good performance of OTD rate is usually obtained at the cost of bad performance of mean tardiness and maximum tardiness, as can be seen in Table 5 and 6 in the following subsections.

<< Insert Table 4 about here >>

5.2 Mean tardiness

Concerning mean tardiness, the proposed ECR3 rule is again the best one. It is in the best group in 47 of 54 scenarios. The next two rules are COVERT and CR rules, which are recognized in the best group for 35 and 30 times, respectively. These three rules have a similar term in their priority index functions. The term is a ratio based on the allowance time (or slack time) and remaining processing time. This observation could be a hint on designing rules for minimizing mean tardiness.

The OPDD rule, which is the second best rule for the OTD rate, is recognized in the best group of rules with respect to mean tardiness only in loose due date scenarios. The reason is that OTD rate is close to 100% in loose due date scenarios, and increasing OTD rate simultaneously decreases mean tardiness. In moderate and tight due date scenarios, however, expediting some lots to meet due dates may delay other lots and increase mean tardiness. The OPDD rule focuses on the operation due date but does not consider the remaining processing time. It may keep processing lots with short remaining processing time but leave lots with long remaining processing time waiting. This behavior could finish a certain group of lots early and increase OTD rate, but meanwhile it could delay another group of lots and increase mean tardiness.

<< Insert Table 5 about here >>

5.3 Maximum tardiness

The best three rules to reduce maximum tardiness are ECR3 (54 times in the best group), CR (39 times), and SLACK (26 times). Another three rules, COVERT, ATCSR, and RATCS, have close performance (21, 21, and 17 times, respectively). Among three objective functions, performance difference between rules with respect to maximum tardiness is the smallest.

When the load level is heavy and due dates are tight, performance of ATCSR, COVERT, and RATCS is much worse than ECR3. One reason could be that these rules behave like the SPT rule when dealing with tardy lots. As the load level gets heavy and due dates get tight, there are more tardy lots. Preferring tardy lots with shorter processing time regardless of their actual tardiness is not a suitable strategy to reduce maximum tardiness. For example, imagine the situation where the lot responsible for maximum tardiness is lying in the queue and keeps increasing maximum tardiness just because the rule does not like its long processing time. By contrast, we use the due date extension procedure and record the number of times of extension in ECR3. Expediting lots with more times of extension (implying larger tardiness) in ECR3 is helpful for reducing maximum tardiness.

<< Insert Table 6 about here >>

5.4 Design principles

According to the experimental results, we observed that realizing the shortest-processing-time-first (SPT) principle is good at raising OTD rate. Rules which do not incorporate this principle, such as CR and SLACK, are seldom recognized in the best group of rules with respect to OTD rate. Note that OTD rate only counts the number of lots finished within due dates. In the extreme case, OTD rate can be maximized by expediting some certain lots to meet their due dates and disregarding the remaining lots, even though these lots are delayed for a long time (since the tardiness causes no decrement on OTD rate). This explains why the OPDD rule, which does not consider tardiness of lots, and the WPWX rule, which does not even consider due dates of lots, can provide high OTD rates in some fabs, especially in scenarios of moderate and tight due dates and heavy load. As we mentioned, however, ignorance of due dates and tardiness causes weak performance of OPDD and WPWX in terms of mean tardiness and maximum tardiness. They are usually in the worst two or three rules regarding these two objective functions.

To raise OTD rate and reduce mean tardiness simultaneously, the dispatching rule should not only follow the SPT principle but also consider due date-based urgency carefully. The proposed ECR3 rule and the existing COVERT rule have a good balance between favoring lots with short processing time and lots with high urgency. Thus, they achieve good performance for OTD rate and mean tardiness at the same time. The CR rule gives good performance for mean tardiness but not OTD rate because it does not realize the SPT principle. The COVERT rule measures due date urgency based on the ratio of the slack time to the remaining processing time. Then, the SPT principle is implemented by dividing due date urgency by the processing time of the imminent operation. The calculation is simple but not easy to explain the interaction between short processing time and high urgency. In ECR3, this interaction is measured by total degree of urgency of all competing lots after the selected lot is processed. Selecting a lot with short processing time prevents total degree of urgency from increasing, and selecting a lot with high urgency decreases more total degree of urgency. The design is easier to understand and achieves better performance than the COVERT rule does.

Another problem in the COVERT rule is that the rule degenerates to be the SPT rule when dealing with tardy lots. (The problem also occurs in the ATCSR and RATCS rules.) The COVERT rule prioritizes tardy lots over non-tardy lots (assuming equal processing time). This is able to reduce maximum tardiness in a certain degree, but processing tardy lots in the SPT order is not good enough. The CR and SLACK rule consider the actual amount of tardiness ($d_i - t$) and achieve lower maximum tardiness than the COVERT rule does. In the proposed ECR3 rule, the due date extension procedure makes the balance between short processing time and high urgency still feasible for tardy lots. In addition, the urgency function reflects the amount of tardiness (see equation (7) and (9)), and therefore ECR3 can expedite lots with longer tardiness. These designs make the ECR3 rule the best one in terms of maximum tardiness.

5.5 Parameter values of ECR3

The ECR3 rule has four parameters. Parameter Y_1 is the extra duration in the due date extension; Y_2 controls the lower bound of degree of urgency after due date extension; Y_3 determines whether the degree of urgency of the processed lot is considered; Y_4 is the threshold of difference in degree of urgency in the lot filtering procedure. In the experiments, we do not intend to tune the parameter values deliberately. We want to keep the number of variants acceptable and set the parameter values with simple reasoning. Normally, the degree of urgency is between 0 and 1 ($urg(R, a)$ in equation (5)). We tested two values, 0.3 and 0.4, for Y_2 to make the lower bound of degree of urgency ($1/(1+Y_2))^2$) of the lot around 0.5~0.6, which is not too high and not too low. We tested two values, 0 and 1, for Y_3 since they are the only two possible values. For the threshold of difference in degree of urgency, Y_4, it should not be too small, which may filter out too many lots; it should not be too large, which reduces the effect of filtering. Since the degree of urgency is between 0 and 1 in the normal case, we thought that 0.3 and 0.4 might be two reasonable choices for Y_4. By observing that the maximum tardiness may range from 5 to 200 in the preliminary simulation results, we tested a little bit more values (four values: 5, 10, 20, and 40) for parameter Y_1. In total, the number of combinations of different parameter values is 32 (4×2×2×2), which is not too large.

To examine the effect of parameters on the three objective functions in different scenarios, we did a similar analysis to what we did in Table 3. Regarding each objective function, we identified the ECR3 variant with the best average performance. Then, we conducted paired *t*-test to find the best group of ECR3 variants, i.e. the variants whose average performance is not statistically different from the best variant. For each parameter value, we counted the number of fabs where at least one rule variant in the best group used that value. Table 7 summarizes the results.

<< Insert Table 7 about here >>

In Table 7, the first observation is that the setting of Y_4 is proper and applicable to all tested cases. In the best group of ECR3 variants, there is always at least one variant using either 0.3 or 0.4 as the value of Y_4. There is a little difference between performance of the two tested values of Y_2, and setting Y_2 to 0.4 is applicable to almost all cases. The effect of values of Y_1 is higher than that of Y_2 and Y_4, particularly when OTD rate is concerned and due dates are tight. A general observation is that larger Y_1 is beneficial to increase OTD rate but smaller Y_1 is good at reducing mean tardiness and maximum tardiness. The result is understandable. Smaller Y_1 gives shorter extra duration in due date extension and results in higher degree of urgency and more times of due date extension for tardy lots, both expediting tardy lots in the logic of ECR3 and helping to reduce mean tardiness and maximum tardiness. On the other hand, larger Y_1 assigns lower degree of urgency to tardy lots and invokes fewer due date extension. It makes ECR3 focus on the non-tardy lots since processing of tardy lots has no benefit in increasing OTD rate. The last observation is that consideration of the degree of urgency of the processed lot, i.e. setting Y_3 to 1, provides better performance. By looking at the detailed experimental results, we found that setting Y_3 to 0 is beneficial only in fab 1 and 5, especially when OTD rate is concerned. We leave further investigation on this phenomenon in our future work.

Based on the analysis, we suggest setting values of parameters Y_2, Y_3, and Y_4 to 0.4, 1, and 0.4, respectively. This setting showed robust performance for three due date-based objective functions in nine fabs under six scenarios with different load levels and due date tightness. As for Y_1, larger values are preferred for OTD rate and smaller values are preferred for mean tardiness and maximum tardiness.

6 Conclusions

Wafer fabrication is a complex manufacturing process, and scheduling is a critical function to make fabs run efficiently to satisfy the concerned performance objectives. The increasing importance of on-time delivery in wafer fabrication highlights the requirement of scheduling methods for due date-based objectives. In this research we developed a dispatching rule that improves the design of existing rules by the index function based on total degree of urgency and the due date extension procedure. Total degree of urgency considers the impact of processing of a lot on the competing lots and extends the scope of utilized information. The due date extension procedure helps to dispatching the tardy lots with better logic instead of merely SPT in many existing rules. We tested the rules by 54 scenarios of fabs, made by nine data sets of fabs, two levels of fab load, and three levels of lot due dates. The results showed that our rule is superior to nine benchmark rules in terms of OTD rate, mean tardiness, and maximum tardiness. In addition to numerical experiments, we had discussions about pros and cons of the tested rules. In the future, we plan to improve the proposed rule by incorporating more ideas from existing rules, for example, "prediction of waiting times" in [25], "consideration of incoming lots" in [33], and "workload balancing" in [36]. Another research direction is to extend the proposed rule for batch scheduling.

Acknowledgments

The authors are grateful to the editor and the reviewers for their valuable suggestions and comments. This research was partially supported by the National Science Council of the Republic of China (Taiwan) under Grant No. 98-2221-E-003-012 and 98-2218-E-002-014.

References

[1] Gupta JND, Ruiz R, Fowler JW, Mason SJ. Operational planning and control of semiconductor wafer production. Production Planning & Control 2006; 17:639–647.

[2] Pfund ME, Mason SJ, Fowler JW. Semiconductor manufacturing scheduling and dispatching. In: Handbook of Production Scheduling, Springer International Series, 2006, p. 213–241.

[3] Pinedo M. Scheduling Theory, Algorithms and Systems, 2nd ed.. Prentice Hall; 2002.

[4] Gupta AK, Sivakumar AI. Job shop scheduling techniques in semiconductor manufacturing. International Journal of Advanced Manufacturing Technology 2006; 27:1163–1169.

[5] Giegandt A, Nicholson G. Better dispatch application – a success story. In: Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 1998, p. 396–399.

[6] Dabbas RM, Fowler JW, Dwayne AR, Mccarville D. Multiple response optimization using mixture-designed experiments and desirability functions in semiconductor scheduling. International Journal of Production Research 2003; 41(5):939–961.

[7] Ham M, Dillard F. Dynamic photo stepper dispatching/ scheduling in wafer fabrication. In: Proceedings of International Symposium on Semiconductor Manufacturing, 2005, p. 75–79.

[8] Morrison JR, Campbell B, Elizabeth D, LaFreniere J. Implementation of a fluctuation smoothing production control policy in IBM's 200mm wafer fab. In: Proceedings of the 44th IEEE Conference on decision and Control, 2005, p. 7732−7737.

[9] Duwayri Z, Mollaghasemi M, Nazzal D, Rabadi G. Scheduling setup changes at bottleneck workstations in semiconductor manufacturing. Production Planning & Control 2006; 17(7):717–727.

[10] Panwalker SS, Iskander W. A survey of scheduling rules. Operations Research 1977; 25:45−61.

[11] Blackstone JH, Phillips DT, Hogg GL. A state-of-the-art survey of dispatching rules for job shop operations. International Journal of Production Research 1982; 20:27−45.

[12] Rajendran C, Holthaus O. A comparative study of dispatching rules in dynamic flowshops and jobshops. European Journal of Operational Research 1999; 116:156−170.

[13] Jayamohan MS, Rajendran C. New dispatching rules for shop scheduling: a step forward. International Journal of Production Research 2000; 38(3):563−586.

[14] Sarin SC, Varadarajan A, Wang L. A survey of dispatching rules for operational control in wafer fabrication. Production Planning & Control 2011; 22(1):4–24.

[15] Lu SCH, Ramaswamy D, Kumar PR. Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. IEEE Transactions on Semiconductor Manufacturing 1994; 7(3):374–388.

[16] Yoon HJ, Lee DY. A control method to reduce the standard deviation of flow time in wafer fabrication. IEEE Transactions on Semiconductor Manufacturing 2000; 13(3):389–392.

[17] Bahaji N., Kuhl ME. A simulation study of new multi-objective composite dispatching rules, CONWIP, and push lot release in semiconductor fabrication. International Journal of Production Research 2008; 46(14):3801–3824.

[18] Little JDC. A proof of the queueing formula: $L = \lambda W$. Operations Research 1961; 9:383−387.

[19] Li S, Tang T, Collins DW. Minimum inventory variability schedule with applications in semiconductor fabrication. IEEE Transactions on Semiconductor Manufacturing 1996; 9(1):145−149.

[20] Lee YH, Park JW, Kim SY. Experimental study on input and bottleneck scheduling for a semiconductor fabrication line. IIE Transactions 2002; 34:179–190.

[21] Vepsalainen APJ, Morton TE. Priority rules or job shops with weighted tardiness costs. Management Science 1987; 33(8):1035–1047.

[22] Hung YF, Chen IR. A simulation study of dispatch rules for reducing flow times in semiconductor wafer fabrication. Production Planning & Control 1998; 9(7):714–722.

[23] Kim YD, Kim JU, Lim SK, Jun HB. Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. IEEE Transactions on Semiconductor Manufacturing 1998; 11(1):155–164.

[24] Kim YD, Kim JG, Choi B, Kim HU. Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. IEEE Transactions on Robotics and Automation 2001; 17(5):589–598.

[25] Chen T. An optimized tailored nonlinear smoothing rule for scheduling a semiconductor manufacturing factory. Computers & Industrial Engineering 2010; 58:317–325.

[26] Rose O. Some issues of the critical ratio dispatch rule in semiconductor manufacturing. In: Proceedings of the 2002 Winter Simulation Conference, 2002, p. 1401–1405.

[27] Rose O. Comparison of due-date oriented dispatching rules in semiconductor manufacturing. In: Proceedings of the Industrial Engineering Research Conference, 2003.

[28] Li L, Qiao F, Jiang H, Wu Q. The research on dispatching rule for improving on-time delivery for semiconductor wafer fab. In: Proceedings of 8th International Conference on Control, Automation, Robotics, and Vision, 2004, p. 494–498.

[29] Wu MC, Jiang JH, Chang WJ. Scheduling a hybrid MTO/MTS semiconductor fab with machine-dedication features. International Journal of Production Economics 2008; 112:416–426.

[30] Chern CC, Liu YL. Family-based scheduling rules of a sequence-dependent wafer fabrication system. IEEE Transactions on Semiconductor Manufacturing 2003; 16(1):15–25.

[31] Lee YH, Pinedo M. Scheduling jobs on parallel machines with sequence-dependent setup times. European Journal of Operational Research 1997; 100:464–474.

[32] Kang YH, Kim SS, Shin HJ. A scheduling algorithm for the reentrant shop: an application in semiconductor manufacture. International Journal of Advanced Manufacturing Technology 2007; 35:566–574.

[33] Pfund M, Fowler JW, Gadkari A, Chen Y. Scheduling jobs on parallel machines with setup times and ready times. Computers & Industrial Engineering 2008, 54:764–782.

[34] Chen JC, Chen CW, Tai CY, Tyan JC. Dynamic state-dependent dispatching for wafer fabrication. International Journal of Production Research 2004; 42(21):4547–4562.

[35] Miragliotta G, Perona M. Decentralized, multi-objective driven scheduling for reentrant shops: A conceptual development and a test case. European Journal of Operational Research 2005; 167:644–662.

[36] Wu MC, Chiou SJ, Chen CF. Dispatching for make-to-order wafer fabs with machine-dedication and mask set-up characteristics. International Journal of Production Research 2008; 46(14):3993–4009.

[37] Zhang H, Jiang Z, Guo C. An optimized dynamic bottleneck dispatching policy for semiconductor wafer fabrication. International Journal of Production Research 2009; 47(12):3333–3343.

[38] Min HS, Yih Y. Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. International Journal of Production Research 2003; 41(16):3921–3941.

[39] Sivakumar AI, Gupta AK. Online multiobjective Pareto optimal dynamic scheduling of semiconductor back-end using conjunctive simulated scheduling. IEEE Transactions on Electronics, Packaging, and Manufacturing 2006; 29(2):99–109.

[40] Sha DY, Liu CY. A simulated annealing algorithm for integration of shop floor control strategies in semiconductor wafer fabrication. International Journal of Advanced Manufacturing Technology 2003; 22:75–88.

[41] Liu M, Wu C. Genetic algorithm using sequence rule chain for multi-objective optimizaiton in re-entrant micro-electronic produciton line. Robotics and Computer-Integrated Manufacturing 20(3):225–236.

[42] Chiang TC, Shen YS, Fu LC. A new paradigm for rule-based scheduling in the wafer probe center, International Journal of Production Research 2008; 46(15):4111–4133.

[43] Adams J, Balas E, Zawack D. The shifting bottleneck procedure for job shop scheduling. Management Science 1988; 34(3):391–401.

[44] Upasani AA, Uzsoy R, Sourirajan K. A problem reduction approach for scheduling semiconductor wafer fabrication facilities. IEEE Transactions on Semiconductor Manufacturing 2006; 19(2):216–225.

[45] Sourirajan K, Uzsoy R. Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication. Journal of Scheduling 2007; 10(1):41–65.

[46] Pfund M, Balasubramanian H, Fowler JW, Mason SJ, Rose O. A multi-criteria approach for scheduling semiconductor wafer fabrication facilities. Journal of Scheduling 2008; 11(1):29–47.

[47] Mönch L, Schabacker R, Pabst D, Fowler JW. Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. European Journal of Operational Research 2007; 177(3):2100–2118.

[48] Mönch L, Balasubramanina H, Fowler JW, Pfund ME. Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. Computers & Operations Research 2005; 32(11):2731–2750.

[49] Geiger CD, Uzsoy R, Aytug H. Rapid modeling and discovery of priority dispatching rules: an autonomous learning approach. Journal of Scheduling 2006; 9(1):7-34.

[50] Tay JC, Ho NB. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Computers & Industrial Engineering 2008; 54(3):453-473.

[51] Chiang TC, Fu LC. Multiobjective job shop scheduling using rule-coded genetic local search. In: Proceedings of International Conference on Computers and Industrial Engineering, 2006, p. 1764–1775.

[52] Chiang TC, Fu LC. Solving the FMS scheduling problem by critical ratio-based heuristics and the genetic algorithm. In: Proceedings of IEEE International Conference on Robotics and Automation, 2004, p. 3131–3136.

[53] Chiang TC, Fu, LC. Using dispatching rules for job shop scheduling with due date-based objectives. International Journal of Production Research 2007; 45(14):3245–3262.

[54] Fowler JW, Robinson J. Measurement and improvement of manufacturing capacities (MIMAC): Final report. Technical Report 95062861A-TR, SEMATECH, Austin, TX; 1995.

[55] Campbell E, Ammenheuser J. 300 mm factory layout and material handling modeling: Phase II report. International SEMATECH; 2000.

[56] Law AM. Simulation Modeling and Analysis, 4th ed., McGraw-Hill; 2007.

[57] Mittler M, Schoemig AK. Comparison of dispatching rules for semiconductor manufacturing using large facility models. In: Proceedings of the Winter Simulation Conference, 1999, p. 709–719.

[58] Uzsoy R, Church LK, Ovacik IM, Hinchman J. Performance evaluation of dispatching rules for semiconductor testing operations. Journal of Electronics 1993; 3:95–105.

[59] Kutanoglu E, Sabuncuoglu I. An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. International Journal of Production Research 1999; 37(1):165–187.

[60] Chiang TC, Fu LC. A simulation study on dispatching rules in semiconductor wafer fabrication facilities with due date-based objectives. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 2006, p. 4660–4665.

[61] Kim YD, Shim SO, Choi B, Hwang H. Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. IEEE Transactions on Semiconductor Manufacturing 2003; 16(2):290–298.

[62] Russell RS, Dar-El EM, Taylor III BW. A comparative analysis of the COVERT job sequencing rule using various shop performance measures. International Journal of Production Research 1987; 25(10):1523–1540.

[63] Benjaafar S, Sheikhzadeh M. Scheduling policies, batch sizes, and manufacturing lead times. IIE Transactions 1997; 29:159–166.

[64] Chen Q, Xi L, Wang Y. The impact of release times, lot size, and scheduling policy in an A&T facility. International Journal of Advanced Manufacturing Technology 2005; 29(5):577–583.

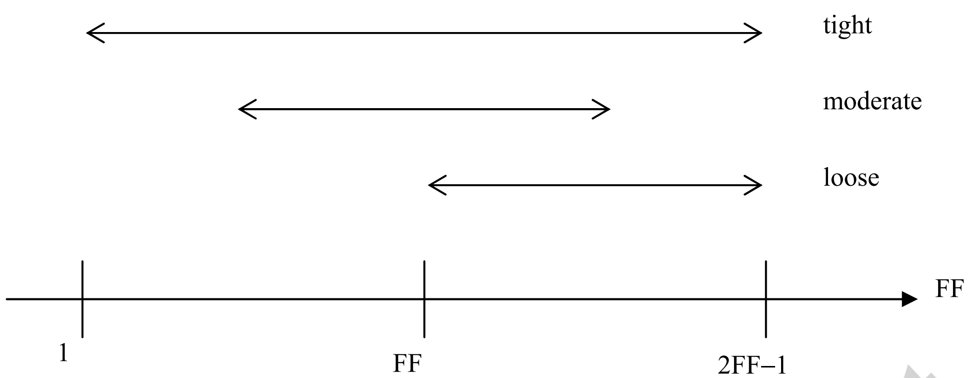

Fig. 1 Ranges of due dates in three scenarios

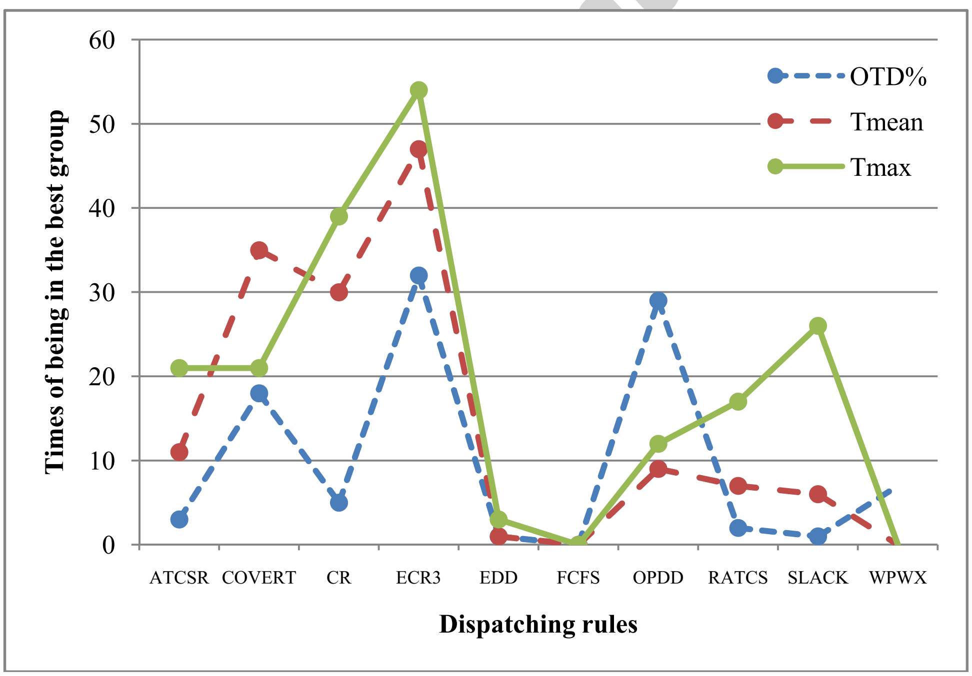

Fig. 2 Times of being in the best group of dispatching rules with respect to different objective functions

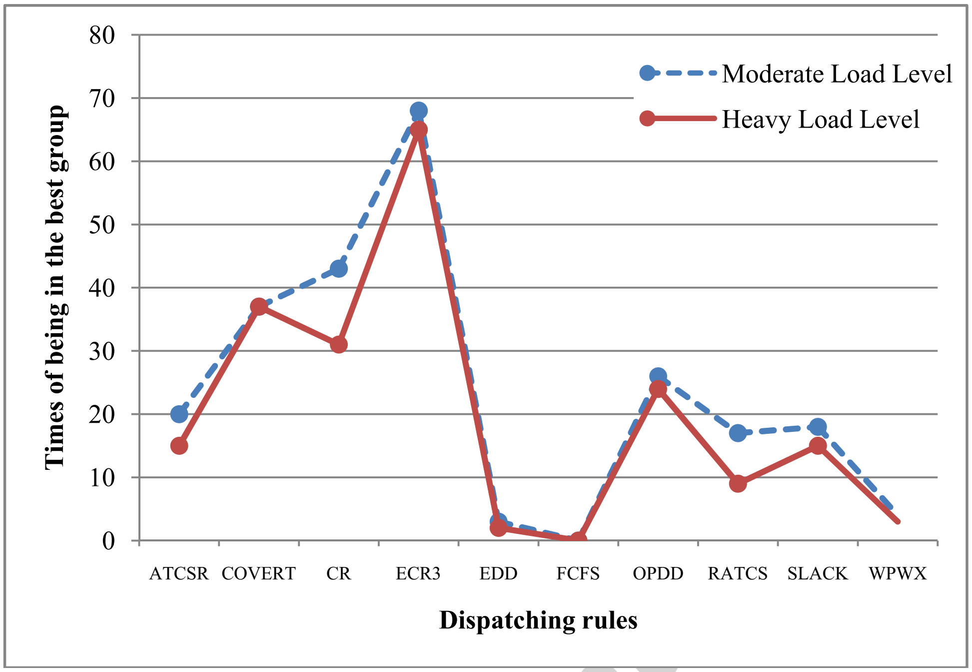

Fig. 3 Times of being in the best group of dispatching rules with respect to different load levels

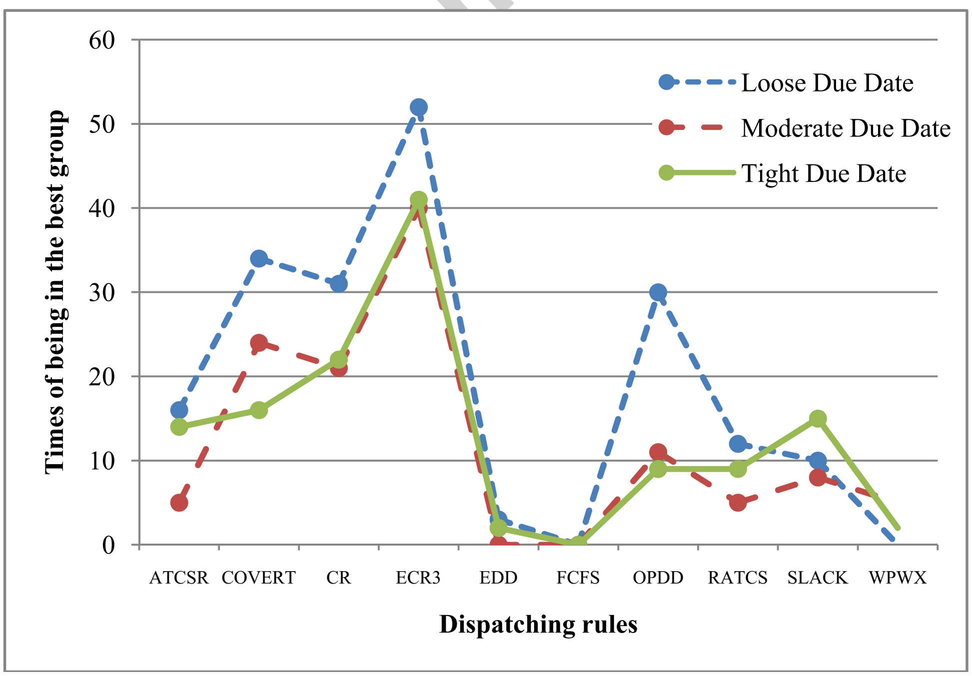

Fig. 4 Times of being in the best group of dispatching rules with respect to different due date tightness

Table 1 An example for the ECR3 rule

					Sequence dependent setup time		
	d_i	r_i	p_i	s_{ki}	s_{i1}	s_{i2}	s_{i3}
Lot 1	30	10	5	0	0	2	2
Lot 2	30	20	4	2	2	0	2
Lot 3	50	40	2	4	4	4	0

Algorithm 1 Lot filtering procedure in the ECR3 rule

Q: the set of all waiting lots; S: the set of lots to be considered as the next processing target; u_i: degree of urgency of lot i

```
LotFilteringProcedure(Q)
Begin
    S = Q
    u^max = max{u_i}, u^min = min{u_i}, ū =∑_i u_i/|S|, ∀ i ∈ S
    While u^max − u^min > Y_4 Do
        AtLeastOneLotIsFiltered = FALSE
        For all i ∈ S
            If e_i = 0 and u_i < ū Then
                S = S/{i}
                AtLeastOneLotIsFiltered = TRUE
            End if
        End for
        If AtLeastOneLotIsFiltered = FALSE
            Break
        End if
        u^max = max{u_i}, u^min = min{u_i}, ū =∑_i u_i/|S|, ∀ i ∈ S
    End while
    Return S
End
```

Table 2 Brief description of nine tested fabs

Fab	N_P / N_R	N_E	Total number of operations in N_R routes	*FF*
1	2 / 2	83	455	1.76 / 2.03*
2	7 / 7	97	1606	1.10 / 1.11
3	11 / 11	73	4139	1.19 / 1.26
4	7 / 2	35	111	1.54 / 1.62
5	21 / 14	85	2581	1.44 / 1.58
6	9 / 9	104	2541	1.71 / 1.99
7	1 / 1	24	172	1.35 / 1.41
8	4 / 4	27	170	1.32 / 1.38
9	1 / 1	43	316	1.10 / 1.13

*: FF in the moderate load level scenario / FF in the heavy load level scenario

Table 3 Summary of performance of ten tested rules in six scenarios with respect to three objective functions

	OTD%						T_{mean}						T_{max}					
	Moderate load level			Heavy load level			Moderate load level			Heavy load level			Moderate load level			Heavy load level		
	L	M	T	L	M	T	L	M	T	L	M	T	L	M	T	L	M	T
ATCSR	1	0	0	2	0	0	3	0	4	2	0	2	4	3	5	4	2	3
COVERT	5	2	1	6	2	2	6	7	5	6	7	4	5	3	3	6	3	1
CR	3	0	0	2	0	0	7	5	5	7	2	4	6	8	9	6	6	4
ECR3	8	4	4	9	4	3	9	7	9	8	7	7	9	9	9	9	9	9
EDD	0	0	0	1	0	0	0	0	0	1	0	0	1	0	2	0	0	0
FCFS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPDD	4	6	5	6	4	4	5	0	0	4	0	0	6	0	0	5	1	0
RATCS	0	0	0	2	0	0	2	0	3	1	0	1	5	3	4	2	2	1
SLACK	0	0	0	1	0	0	2	0	2	1	0	1	3	5	6	3	3	6
WPWX	0	3	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0

L, M, and T denote the loose, moderate, and tight due date scenarios, respectively.

Table 4 Average OTD rates of ten rules in nine fabs with six scenarios

Loose due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
OPDD	0.9945	ECR3	0.8643	OPDD	0.9894	ECR3	0.9353	COV	0.9223	ECR3	0.9976	OPDD	0.9393	ECR3	0.8442	ECR3	0.9349
ECR3	0.9943	COV	0.8594	ATCSR	0.9881	CR	0.9335	ECR3	0.9207	OPDD	0.9972	ECR3	0.9372	COV	0.8424	ATCSR	0.9314
RATCS	0.9903	ATCSR	0.8567	ECR3	0.9880	COV	0.9329	CR	0.9125	COV	0.9972	COV	0.9359	CR	0.8398	COV	0.9313
ATCSR	0.9894	CR	0.8562	CR	0.9872	OPDD	0.9286	RATCS	0.9083	CR	0.9971	CR	0.9324	ATCSR	0.8349	OPDD	0.9307
SLACK	0.9846	RATCS	0.8544	RATCS	0.9871	ATCSR	0.9275	ATCSR	0.9072	RATCS	0.9956	RATCS	0.9177	RATCS	0.8347	RATCS	0.9298
COV	0.9829	OPDD	0.8527	SLACK	0.9862	RATCS	0.9272	OPDD	0.9042	ATCSR	0.9953	ATCSR	0.9161	OPDD	0.8302	CR	0.9296
EDD	0.9807	SLACK	0.8515	COV	0.9843	SLACK	0.9265	SLACK	0.9029	SLACK	0.9943	SLACK	0.9099	SLACK	0.8288	SLACK	0.9288
CR	0.9703	WPWX	0.8408	EDD	0.9674	EDD	0.9142	EDD	0.8788	EDD	0.9794	WPWX	0.8816	FCFS	0.8246	FCFS	0.9051
FCFS	0.9442	FCFS	0.8387	FCFS	0.9491	FCFS	0.8893	FCFS	0.8384	FCFS	0.9637	FCFS	0.8741	EDD	0.8065	EDD	0.8997
WPWX	0.8486	EDD	0.8258	WPWX	0.9263	WPWX	0.8677	WPWX	0.8031	WPWX	0.8817	EDD	0.8724	WPWX	0.7984	WPWX	0.8982

Moderate due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	0.6745	OPDD	0.5859	OPDD	0.6469	WPWX	0.7397	ECR3	0.5329	COV	0.7150	OPDD	0.5234	ECR3	0.6414	OPDD	0.6253
COV	0.6688	ECR3	0.5843	ECR3	0.5847	ECR3	0.6582	OPDD	0.5274	OPDD	0.6766	WPWX	0.4747	COV	0.6337	WPWX	0.5305
OPDD	0.6617	WPWX	0.5808	COV	0.5801	COV	0.6533	COV	0.5253	ECR3	0.6563	FCFS	0.4718	ATCSR	0.6217	ECR3	0.5118
WPWX	0.6502	COV	0.5728	CR	0.5734	CR	0.6530	RATCS	0.5219	CR	0.6059	ATCSR	0.4517	RATCS	0.6186	FCFS	0.4914
RATCS	0.6266	ATCSR	0.5709	ATCSR	0.5645	RATCS	0.6434	ATCSR	0.5210	WPWX	0.5958	ECR3	0.4492	CR	0.6180	COV	0.4753
ATCSR	0.6243	RATCS	0.5703	RATCS	0.5636	ATCSR	0.6398	CR	0.5189	RATCS	0.5291	RATCS	0.4405	FCFS	0.6109	RATCS	0.4731
SLACK	0.6037	FCFS	0.5672	SLACK	0.5500	SLACK	0.6369	SLACK	0.5103	ATCSR	0.5250	EDD	0.4350	WPWX	0.6063	ATCSR	0.4726
CR	0.6014	SLACK	0.5642	WPWX	0.5330	OPDD	0.6294	WPWX	0.5073	FCFS	0.4905	CR	0.4336	SLACK	0.6009	CR	0.4644
EDD	0.5316	EDD	0.5624	EDD	0.5265	EDD	0.6171	EDD	0.4973	SLACK	0.3981	COV	0.4334	OPDD	0.5982	SLACK	0.4615
FCFS	0.4882	CR	0.5567	FCFS	0.5090	FCFS	0.5879	FCFS	0.4824	EDD	0.3345	SLACK	0.4286	EDD	0.5610	EDD	0.4437

Tight due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	0.6110	OPDD	0.5401	OPDD	0.5784	WPWX	0.6047	OPDD	0.5133	COV	0.6116	OPDD	0.5355	ECR3	0.5624	OPDD	0.5860
OPDD	0.5921	ECR3	0.5299	COV	0.5404	ECR3	0.5873	ECR3	0.5056	ECR3	0.5997	WPWX	0.4783	COV	0.5490	WPWX	0.4985
ATCSR	0.5832	WPWX	0.5279	ECR3	0.5376	COV	0.5748	COV	0.5036	OPDD	0.5970	FCFS	0.4762	RATCS	0.5392	ECR3	0.4915
COV	0.5831	FCFS	0.5198	CR	0.5349	ATCSR	0.5714	RATCS	0.4933	CR	0.5898	ECR3	0.4749	ATCSR	0.5392	FCFS	0.4798
RATCS	0.5808	COV	0.5179	ATCSR	0.5337	CR	0.5699	ATCSR	0.4921	RATCS	0.5512	COV	0.4633	CR	0.5377	COV	0.4634
SLACK	0.5667	RATCS	0.5170	RATCS	0.5330	RATCS	0.5694	SLACK	0.4888	ATCSR	0.5486	ATCSR	0.4605	FCFS	0.5336	ATCSR	0.4623
WPWX	0.5480	ATCSR	0.5170	SLACK	0.5286	SLACK	0.5679	CR	0.4866	SLACK	0.5378	RATCS	0.4569	SLACK	0.5283	RATCS	0.4622
EDD	0.5438	EDD	0.5146	EDD	0.5181	EDD	0.5579	EDD	0.4750	WPWX	0.5233	EDD	0.4516	OPDD	0.5227	SLACK	0.4590
CR	0.5398	SLACK	0.5080	WPWX	0.5152	OPDD	0.5559	WPWX	0.4661	FCFS	0.4995	CR	0.4512	WPWX	0.5214	CR	0.4584
FCFS	0.4918	CR	0.5060	FCFS	0.5054	FCFS	0.5422	FCFS	0.4645	EDD	0.4729	SLACK	0.4494	EDD	0.5009	EDD	0.4469

Table 4 Average OTD rates of ten rules in nine fabs with six scenarios (continued)

Loose due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
OPDD	0.9998	ECR3	0.8585	OPDD	0.9990	ECR3	0.9298	COV	0.9859	COV	0.9996	COV	0.9698	ECR3	0.8383	ECR3	0.9600
RATCS	0.9997	COV	0.8533	ECR3	0.9986	CR	0.9240	ECR3	0.9852	ECR3	0.9989	ECR3	0.9667	COV	0.8374	OPDD	0.9571
ATCSR	0.9997	CR	0.8485	CR	0.9985	COV	0.9238	CR	0.9834	CR	0.9987	OPDD	0.9635	CR	0.8320	COV	0.9564
SLACK	0.9995	ATCSR	0.8468	SLACK	0.9973	ATCSR	0.9226	OPDD	0.9833	OPDD	0.9985	CR	0.9515	ATCSR	0.8230	CR	0.9556
EDD	0.9973	RATCS	0.8442	COV	0.9972	OPDD	0.9205	RATCS	0.9801	ATCSR	0.9969	ATCSR	0.9332	RATCS	0.8230	RATCS	0.9542
ECR3	0.9970	OPDD	0.8415	RATCS	0.9971	RATCS	0.9189	ATCSR	0.9798	RATCS	0.9962	RATCS	0.9294	OPDD	0.8149	ATCSR	0.9541
COV	0.9932	SLACK	0.8371	ATCSR	0.9968	SLACK	0.9166	SLACK	0.9795	SLACK	0.9857	SLACK	0.9197	SLACK	0.8116	SLACK	0.9514
CR	0.9602	WPWX	0.8262	EDD	0.9535	EDD	0.9036	EDD	0.9667	FCFS	0.9666	FCFS	0.8844	FCFS	0.8074	FCFS	0.9117
FCFS	0.9577	FCFS	0.8221	FCFS	0.9453	WPWX	0.8472	FCFS	0.9309	EDD	0.9235	WPWX	0.8836	WPWX	0.7742	EDD	0.8947
WPWX	0.8600	EDD	0.8075	WPWX	0.8861	FCFS	0.8409	WPWX	0.8509	WPWX	0.8660	EDD	0.8722	EDD	0.7697	WPWX	0.8930

Moderate due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	0.8320	OPDD	0.5323	OPDD	0.6749	WPWX	0.7359	ECR3	0.6195	COV	0.8695	OPDD	0.5499	ECR3	0.6236	OPDD	0.6657
COV	0.7322	WPWX	0.5314	COV	0.6141	ECR3	0.5845	COV	0.6179	ECR3	0.7793	WPWX	0.4682	COV	0.6170	WPWX	0.5481
OPDD	0.7315	ECR3	0.5299	CR	0.5967	CR	0.5790	RATCS	0.5807	WPWX	0.6850	FCFS	0.4574	CR	0.5946	ECR3	0.5207
WPWX	0.7310	COV	0.5133	ECR3	0.5913	RATCS	0.5725	ATCSR	0.5777	OPDD	0.6765	ECR3	0.4464	ATCSR	0.5844	FCFS	0.4698
RATCS	0.7117	ATCSR	0.5132	RATCS	0.5688	ATCSR	0.5724	OPDD	0.5770	ATCSR	0.5631	RATCS	0.4265	RATCS	0.5809	COV	0.4617
ATCSR	0.6902	FCFS	0.5125	ATCSR	0.5507	COV	0.5720	SLACK	0.5713	RATCS	0.5535	ATCSR	0.4263	WPWX	0.5806	RATCS	0.4588
SLACK	0.6413	RATCS	0.5125	SLACK	0.5246	SLACK	0.5634	CR	0.5706	FCFS	0.5053	COV	0.4009	FCFS	0.5783	ATCSR	0.4572
CR	0.6079	EDD	0.5003	WPWX	0.5032	OPDD	0.5530	WPWX	0.5253	CR	0.4201	SLACK	0.3879	OPDD	0.5631	CR	0.4558
EDD	0.5051	SLACK	0.4990	FCFS	0.4855	EDD	0.5364	EDD	0.5150	SLACK	0.0645	EDD	0.3746	SLACK	0.5564	SLACK	0.4468
FCFS	0.4770	CR	0.4872	EDD	0.4391	FCFS	0.4902	FCFS	0.4877	EDD	0.0443	CR	0.3575	EDD	0.4835	EDD	0.3541

Tight due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	0.7127	OPDD	0.5068	OPDD	0.5907	WPWX	0.6050	COV	0.5829	COV	0.6817	OPDD	0.5393	ECR3	0.5461	OPDD	0.5965
RATCS	0.6747	ECR3	0.4952	COV	0.5723	ECR3	0.5258	ECR3	0.5816	ECR3	0.6726	WPWX	0.4833	COV	0.5344	WPWX	0.5053
ATCSR	0.6658	WPWX	0.4925	CR	0.5529	CR	0.5175	ATCSR	0.5654	CR	0.6488	FCFS	0.4790	CR	0.5204	ECR3	0.4995
SLACK	0.6549	FCFS	0.4868	ECR3	0.5512	SLACK	0.5141	RATCS	0.5651	OPDD	0.5803	ECR3	0.4689	ATCSR	0.5172	FCFS	0.4699
COV	0.6329	COV	0.4824	ATCSR	0.5512	COV	0.5115	SLACK	0.5554	WPWX	0.5579	CR	0.4581	RATCS	0.5142	ATCSR	0.4606
OPDD	0.6258	ATCSR	0.4756	RATCS	0.5474	RATCS	0.5102	OPDD	0.5539	RATCS	0.5533	COV	0.4571	FCFS	0.5063	RATCS	0.4605
EDD	0.6091	RATCS	0.4752	SLACK	0.5313	ATCSR	0.5092	CR	0.5499	ATCSR	0.5503	ATCSR	0.4343	SLACK	0.5014	COV	0.4600
WPWX	0.5976	EDD	0.4687	WPWX	0.5023	OPDD	0.5055	EDD	0.5345	FCFS	0.4979	RATCS	0.4274	WPWX	0.5013	CR	0.4564
CR	0.5652	SLACK	0.4606	FCFS	0.4950	EDD	0.4971	FCFS	0.4945	SLACK	0.4771	EDD	0.4164	OPDD	0.4896	SLACK	0.4543
FCFS	0.4885	CR	0.4605	EDD	0.4943	FCFS	0.4874	WPWX	0.4812	EDD	0.3383	SLACK	0.4069	EDD	0.4572	EDD	0.4125

Table 5 Average mean tardiness of ten rules in nine fabs with six scenarios

Loose due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
OPDD	0.1102	ECR3	0.8592	OPDD	0.0484	ECR3	1.2770	COV	1.2506	ECR3	0.0416	OPDD	1.5275	ECR3	2.8498	ECR3	0.3810
ECR3	0.1168	CR	0.8773	ECR3	0.0527	CR	1.3311	ECR3	1.2862	CR	0.0448	COV	1.5581	CR	2.8669	COV	0.3891
RATCS	0.2192	COV	0.8834	CR	0.0548	OPDD	1.3609	CR	1.4181	COV	0.052	ECR3	1.7796	COV	2.8690	ATCSR	0.3922
ATCSR	0.2366	ATCSR	0.9167	ATCSR	0.0549	SLACK	1.4047	RATCS	1.5372	OPDD	0.0578	CR	1.9727	RATCS	2.9793	RATCS	0.3936
COV	0.3199	RATCS	0.9399	RATCS	0.0616	COV	1.5861	ATCSR	1.5728	ATCSR	0.0868	RATCS	2.5628	ATCSR	3.0186	CR	0.3960
SLACK	0.4223	SLACK	0.9607	SLACK	0.0655	EDD	1.6963	SLACK	1.7002	RATCS	0.0897	ATCSR	2.9543	SLACK	3.1296	OPDD	0.3976
EDD	0.5160	OPDD	1.0844	COV	0.0711	ATCSR	1.7046	EDD	2.2585	SLACK	0.1318	SLACK	3.1332	OPDD	3.2515	SLACK	0.4103
CR	0.5693	FCFS	1.1579	EDD	0.1928	RATCS	1.7057	OPDD	2.3330	EDD	0.4065	EDD	5.0416	FCFS	3.3462	EDD	0.5950
FCFS	2.0060	EDD	1.3875	FCFS	0.2714	FCFS	2.2684	FCFS	3.4127	FCFS	0.9175	WPWX	5.3864	EDD	3.5199	FCFS	0.6201
WPWX	24.107	WPWX	1.3912	WPWX	0.6044	WPWX	9.3754	WPWX	7.4823	WPWX	11.020	FCFS	5.4744	WPWX	4.8266	WPWX	1.9157

Moderate due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
COV	9.923	ECR3	3.1770	CR	3.4194	ECR3	5.7645	COV	16.149	COV	6.4562	CR	26.022	ECR3	6.2589	COV	2.7378
ECR3	10.850	COV	3.1932	ECR3	3.4707	CR	5.7781	ECR3	16.427	ECR3	7.0157	COV	26.992	COV	6.2787	ECR3	2.7609
CR	11.221	CR	3.2280	COV	3.8476	SLACK	6.1519	CR	16.994	CR	7.9213	ECR3	27.147	CR	6.4089	CR	2.7695
RATCS	13.136	ATCSR	3.2375	ATCSR	3.8746	COV	6.8821	RATCS	17.183	RATCS	13.941	RATCS	30.421	ATCSR	6.5466	ATCSR	2.7846
ATCSR	13.165	RATCS	3.2527	RATCS	3.8873	EDD	6.9306	ATCSR	17.242	ATCSR	14.098	ATCSR	30.441	RATCS	6.5584	RATCS	2.7934
SLACK	14.228	SLACK	3.3215	SLACK	3.9210	RATCS	7.2351	SLACK	17.894	SLACK	17.054	SLACK	31.573	SLACK	6.8535	SLACK	2.8331
EDD	19.057	FCFS	3.5667	EDD	5.6721	OPDD	7.3236	EDD	19.685	FCFS	24.314	WPWX	34.462	FCFS	7.0650	FCFS	3.2071
OPDD	23.255	WPWX	3.7360	FCFS	6.0428	ATCSR	7.3335	FCFS	22.437	EDD	27.506	FCFS	35.811	EDD	7.6900	EDD	3.4022
FCFS	28.305	EDD	3.8692	WPWX	6.4809	FCFS	8.3228	OPDD	26.306	OPDD	28.389	EDD	37.094	OPDD	8.0504	OPDD	4.3730
WPWX	48.827	OPDD	3.8732	OPDD	7.1708	WPWX	13.798	WPWX	26.535	WPWX	34.046	OPDD	40.204	WPWX	8.6716	WPWX	4.5361

Tight due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	20.836	ATCSR	4.2182	ECR3	8.6819	ECR3	7.9453	ECR3	23.156	ECR3	24.975	ECR3	43.333	ECR3	7.5304	ATCSR	4.4647
COV	21.896	ECR3	4.2595	CR	8.6960	CR	7.9456	COV	23.267	COV	25.088	COV	44.043	COV	7.5495	COV	4.4650
RATCS	22.834	RATCS	4.2638	ATCSR	8.7172	SLACK	8.1707	RATCS	23.654	CR	25.256	ATCSR	44.382	CR	7.6566	ECR3	4.4671
ATCSR	22.991	COV	4.2816	RATCS	8.8317	EDD	8.7868	CR	23.673	RATCS	26.804	CR	44.455	ATCSR	7.7350	RATCS	4.4862
CR	23.133	CR	4.2878	SLACK	8.9512	COV	9.1142	ATCSR	23.779	ATCSR	26.965	RATCS	44.745	RATCS	7.8007	CR	4.5039
SLACK	24.234	SLACK	4.3559	COV	9.1526	RATCS	9.1415	SLACK	23.997	SLACK	27.536	SLACK	47.718	SLACK	7.9551	SLACK	4.5083
EDD	26.627	FCFS	4.7411	EDD	10.539	ATCSR	9.1470	EDD	25.467	EDD	35.379	EDD	54.529	FCFS	8.4563	EDD	4.9980
OPDD	37.024	WPWX	4.9353	FCFS	11.762	OPDD	10.460	FCFS	29.650	OPDD	47.284	WPWX	55.860	EDD	8.7430	FCFS	5.2036
FCFS	48.271	EDD	4.9473	WPWX	12.062	FCFS	12.119	OPDD	30.340	FCFS	47.537	FCFS	56.610	OPDD	9.4306	WPWX	6.7508
WPWX	62.997	OPDD	5.2235	OPDD	13.782	WPWX	16.125	WPWX	33.364	WPWX	50.213	OPDD	59.877	WPWX	10.282	OPDD	7.3539

Table 5 Average mean tardiness of ten rules in nine fabs with six scenarios (continued)

Loose due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
RATCS	0.0068	ECR3	0.9304	OPDD	0.0047	ECR3	0.7101	COV	0.3425	COV	0.0059	COV	0.7011	ECR3	3.1392	OPDD	0.2516
OPDD	0.0070	COV	0.9440	ECR3	0.0065	CR	0.7707	ECR3	0.3523	ECR3	0.0294	ECR3	0.7730	COV	3.1956	ECR3	0.2570
ATCSR	0.0110	CR	0.9549	CR	0.0068	OPDD	0.9130	CR	0.3733	CR	0.0307	OPDD	0.8550	CR	3.3294	ATCSR	0.2651
SLACK	0.0310	ATCSR	1.0362	SLACK	0.0129	SLACK	0.9294	OPDD	0.5295	ATCSR	0.0607	CR	1.2913	ATCSR	3.5633	RATCS	0.2696
ECR3	0.0652	RATCS	1.0399	RATCS	0.0153	COV	0.9808	ATCSR	0.5786	OPDD	0.0704	ATCSR	2.1060	RATCS	3.5640	COV	0.2703
EDD	0.0699	SLACK	1.0877	COV	0.0173	ATCSR	1.0967	RATCS	0.6372	RATCS	0.0818	RATCS	2.2518	SLACK	3.9534	CR	0.2717
COV	0.1547	OPDD	1.2714	ATCSR	0.0180	RATCS	1.1110	SLACK	0.6508	SLACK	0.4724	SLACK	2.6337	OPDD	4.0446	SLACK	0.2780
CR	0.9042	FCFS	1.3167	EDD	0.4363	EDD	1.1623	EDD	0.9143	FCFS	1.2028	EDD	5.1368	FCFS	4.0513	FCFS	0.6103
FCFS	1.5368	WPWX	1.5966	FCFS	0.4506	FCFS	2.5042	FCFS	1.6240	EDD	2.3060	FCFS	5.3142	EDD	4.7457	EDD	0.6723
WPWX	59.085	EDD	1.5998	WPWX	1.9287	WPWX	15.345	WPWX	13.120	WPWX	34.831	WPWX	8.3638	WPWX	6.8452	WPWX	3.1164

Moderate due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	4.5895	ECR3	3.7879	CR	3.0471	ECR3	7.0034	COV	10.276	COV	2.8953	COV	27.702	COV	7.3367	ECR3	2.9912
COV	5.7534	COV	3.8170	ECR3	3.1710	CR	7.1147	ECR3	11.024	ECR3	4.3441	ECR3	27.905	ECR3	7.3685	COV	3.0410
CR	8.5469	ATCSR	3.8874	COV	4.0293	SLACK	7.7414	CR	12.132	ATCSR	14.511	CR	31.217	CR	7.8922	CR	3.0760
RATCS	8.6118	CR	3.8900	RATCS	4.1104	EDD	8.6381	RATCS	12.550	RATCS	14.605	ATCSR	34.678	RATCS	8.0263	RATCS	3.0981
ATCSR	9.1460	RATCS	3.8957	ATCSR	4.2922	COV	9.1213	ATCSR	12.833	CR	17.690	RATCS	35.429	ATCSR	8.0808	ATCSR	3.1342
SLACK	11.539	SLACK	4.0350	SLACK	4.5274	OPDD	9.1300	SLACK	12.855	FCFS	31.513	SLACK	36.850	SLACK	8.7921	SLACK	3.2080
EDD	19.993	FCFS	4.3198	FCFS	9.1143	RATCS	9.3995	EDD	16.483	OPDD	43.950	FCFS	40.003	FCFS	8.8641	FCFS	4.0950
OPDD	20.299	WPWX	4.3759	EDD	10.378	ATCSR	9.5455	FCFS	21.470	SLACK	48.634	WPWX	41.825	OPDD	10.471	EDD	4.9642
FCFS	36.432	EDD	4.7771	OPDD	10.816	FCFS	11.836	OPDD	25.981	WPWX	65.280	EDD	45.321	EDD	10.506	WPWX	6.3657
WPWX	85.168	OPDD	4.8136	WPWX	11.753	WPWX	20.923	WPWX	34.191	EDD	81.655	OPDD	48.869	WPWX	11.795	OPDD	6.4207

Tight due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	15.008	ECR3	4.9392	ECR3	9.4024	CR	9.0957	COV	20.300	ECR3	21.980	ECR3	43.990	COV	8.5707	ATCSR	4.9313
RATCS	16.870	COV	4.9678	CR	9.4632	ECR3	9.2778	ECR3	20.465	COV	22.636	CR	44.415	ECR3	8.7299	RATCS	5.0167
ATCSR	17.294	RATCS	4.9955	ATCSR	9.8682	SLACK	9.3826	RATCS	20.730	CR	22.933	COV	44.691	CR	9.0456	ECR3	5.0385
COV	17.914	ATCSR	5.0016	SLACK	9.9474	EDD	10.324	CR	20.992	RATCS	29.165	ATCSR	49.164	ATCSR	9.1462	COV	5.0458
SLACK	18.228	CR	5.0097	RATCS	9.9812	RATCS	11.469	ATCSR	21.072	ATCSR	29.561	RATCS	49.172	RATCS	9.2899	CR	5.0559
CR	19.023	SLACK	5.1082	COV	10.170	COV	11.731	SLACK	21.328	SLACK	31.125	SLACK	50.702	SLACK	9.5265	SLACK	5.1238
EDD	21.402	FCFS	5.6103	EDD	13.922	ATCSR	11.763	EDD	23.477	EDD	50.720	EDD	61.068	FCFS	10.411	EDD	6.4028
OPDD	33.830	WPWX	5.8209	FCFS	16.934	OPDD	12.079	OPDD	31.810	OPDD	62.825	FCFS	64.478	EDD	11.142	FCFS	6.7522
FCFS	68.455	EDD	5.9612	WPWX	18.249	FCFS	16.244	FCFS	32.727	FCFS	67.062	WPWX	66.293	OPDD	11.841	WPWX	9.1687
WPWX	99.046	OPDD	6.3417	OPDD	18.975	WPWX	23.200	WPWX	42.394	WPWX	78.983	OPDD	72.918	WPWX	13.194	OPDD	10.202

Table 6 Average maximum tardiness of ten rules in nine fabs with six scenarios

Loose due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
OPDD	19.058	ECR3	29.885	OPDD	15.160	OPDD	40.278	ECR3	74.430	COV	24.943	COV	39.946	CR	87.841	OPDD	18.702
ECR3	21.812	RATCS	31.133	ECR3	15.627	ECR3	40.595	RATCS	74.651	CR	25.252	OPDD	42.037	ECR3	88.796	SLACK	19.129
ATCSR	26.468	SLACK	31.202	RATCS	17.903	CR	42.613	COV	75.728	ECR3	26.207	ECR3	47.064	RATCS	89.218	ECR3	19.187
RATCS	28.454	CR	31.721	ATCSR	18.113	SLACK	42.682	ATCSR	77.846	OPDD	29.655	CR	53.520	COV	89.433	CR	19.556
SLACK	34.652	ATCSR	31.882	CR	18.614	EDD	46.350	SLACK	88.992	ATCSR	34.697	ATCSR	61.528	SLACK	90.001	RATCS	19.600
CR	35.392	COV	32.272	SLACK	19.176	FCFS	59.013	CR	89.811	RATCS	39.627	RATCS	62.388	ATCSR	90.025	COV	19.769
COV	36.215	FCFS	36.652	COV	20.184	ATCSR	71.079	EDD	90.497	SLACK	45.293	SLACK	73.183	EDD	91.244	ATCSR	20.547
EDD	48.809	EDD	40.888	EDD	27.192	RATCS	71.079	OPDD	99.959	EDD	67.979	EDD	90.279	FCFS	94.320	EDD	21.895
FCFS	117.10	OPDD	41.674	FCFS	30.088	COV	73.668	FCFS	103.08	FCFS	97.630	FCFS	113.26	OPDD	95.222	FCFS	24.408
WPWX	745.14	WPWX	57.836	WPWX	43.678	WPWX	357.35	WPWX	241.57	WPWX	430.45	WPWX	116.47	WPWX	151.96	WPWX	126.67

Moderate due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	126.48	ECR3	39.293	CR	44.592	SLACK	68.997	ECR3	133.45	ECR3	124.38	ECR3	151.31	CR	102.12	SLACK	29.351
CR	132.92	RATCS	41.147	ECR3	44.636	ECR3	70.117	CR	138.63	CR	136.36	CR	154.18	ECR3	102.20	ECR3	29.960
SLACK	142.39	ATCSR	41.214	ATCSR	45.458	CR	70.134	COV	139.08	SLACK	143.44	COV	156.32	RATCS	104.93	CR	30.217
COV	148.14	SLACK	41.409	RATCS	46.986	EDD	73.018	ATCSR	141.42	COV	148.34	ATCSR	157.72	COV	105.62	EDD	32.117
EDD	164.39	COV	42.125	SLACK	47.338	OPDD	80.086	SLACK	144.98	RATCS	157.82	RATCS	162.68	ATCSR	106.16	COV	32.397
ATCSR	165.30	CR	42.426	COV	48.476	FCFS	85.315	RATCS	145.45	ATCSR	159.86	SLACK	169.31	SLACK	106.88	ATCSR	32.568
RATCS	169.05	FCFS	44.686	EDD	58.671	ATCSR	117.68	EDD	157.18	EDD	173.83	EDD	186.41	EDD	108.98	RATCS	32.760
FCFS	231.82	EDD	50.049	FCFS	58.845	COV	121.82	FCFS	159.16	FCFS	201.19	FCFS	209.38	FCFS	109.82	FCFS	34.630
OPDD	237.14	OPDD	57.916	OPDD	74.717	RATCS	121.88	OPDD	159.64	OPDD	229.23	WPWX	212.60	OPDD	110.12	OPDD	82.704
WPWX	853.33	WPWX	72.156	WPWX	74.990	WPWX	383.65	WPWX	260.56	WPWX	505.40	OPDD	237.54	WPWX	169.58	WPWX	136.89

Tight due dates & moderate load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	218.03	ECR3	43.898	ECR3	67.682	ECR3	77.052	ECR3	175.51	CR	206.07	ATCSR	206.14	ECR3	107.09	SLACK	35.496
CR	227.16	SLACK	44.485	CR	71.009	SLACK	77.093	COV	180.69	ECR3	208.45	SLACK	211.63	SLACK	107.32	ECR3	35.570
COV	235.01	ATCSR	45.009	ATCSR	71.166	CR	78.121	ATCSR	184.20	SLACK	215.45	ECR3	214.05	RATCS	108.83	CR	35.720
ATCSR	235.97	CR	45.178	SLACK	71.592	EDD	80.515	CR	184.78	COV	224.93	RATCS	219.81	COV	109.06	EDD	38.197
SLACK	237.44	RATCS	45.361	RATCS	71.972	OPDD	86.762	RATCS	184.80	ATCSR	229.35	COV	220.81	CR	109.29	RATCS	39.189
EDD	239.35	COV	46.689	COV	77.157	FCFS	105.31	SLACK	186.64	EDD	231.05	CR	223.09	ATCSR	109.53	ATCSR	39.504
RATCS	240.10	FCFS	49.840	EDD	80.705	ATCSR	122.56	EDD	187.92	RATCS	233.41	EDD	249.59	EDD	110.52	COV	39.697
OPDD	273.20	EDD	55.506	FCFS	85.949	COV	122.60	OPDD	188.51	OPDD	267.45	OPDD	276.71	OPDD	112.91	FCFS	44.343
FCFS	348.10	OPDD	62.502	OPDD	97.442	RATCS	125.58	FCFS	215.33	FCFS	293.16	WPWX	277.17	FCFS	115.98	OPDD	73.750
WPWX	942.69	WPWX	73.591	WPWX	100.78	WPWX	383.45	WPWX	332.28	WPWX	594.69	FCFS	278.46	WPWX	166.30	WPWX	138.90

Table 6 Average maximum tardiness of ten rules in nine fabs with six scenarios (continued)

Loose due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
OPDD	4.3371	ECR3	34.496	OPDD	6.3138	ECR3	27.471	ECR3	93.700	COV	5.2466	COV	26.284	ECR3	99.201	OPDD	14.270
ATCSR	4.3403	SLACK	35.686	ECR3	7.0606	OPDD	27.486	CR	98.868	CR	14.164	ECR3	29.088	CR	100.64	ECR3	14.628
RATCS	4.5342	CR	36.235	CR	8.3385	CR	28.957	COV	114.20	ECR3	14.644	OPDD	29.518	COV	101.50	SLACK	14.875
ECR3	4.7399	ATCSR	37.222	SLACK	12.333	SLACK	29.934	ATCSR	132.83	ATCSR	30.434	CR	36.303	FCFS	105.24	RATCS	15.070
SLACK	8.5263	RATCS	37.563	ATCSR	13.610	EDD	33.388	RATCS	135.78	RATCS	38.176	ATCSR	55.738	RATCS	106.56	ATCSR	15.138
EDD	12.128	COV	38.518	RATCS	13.722	COV	40.084	OPDD	149.69	OPDD	41.346	RATCS	59.226	ATCSR	107.04	COV	15.267
COV	20.026	FCFS	42.412	COV	15.029	ATCSR	40.520	EDD	184.31	SLACK	94.495	SLACK	61.735	SLACK	108.83	CR	15.759
CR	37.766	EDD	44.875	FCFS	41.035	RATCS	42.603	FCFS	184.70	EDD	130.46	FCFS	89.704	OPDD	109.11	EDD	23.273
FCFS	120.65	OPDD	50.054	EDD	42.445	FCFS	51.954	SLACK	193.82	FCFS	138.81	EDD	95.529	EDD	111.63	FCFS	24.099
WPWX	2331.5	WPWX	76.259	WPWX	102.78	WPWX	577.16	WPWX	909.02	WPWX	1274.7	WPWX	211.19	WPWX	205.33	WPWX	190.11

Moderate due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	91.178	ECR3	46.064	CR	43.927	ECR3	56.505	COV	197.89	ECR3	142.33	ECR3	148.13	ECR3	118.13	SLACK	28.173
CR	102.55	SLACK	46.474	ECR3	45.943	CR	59.932	ECR3	207.42	COV	165.44	CR	157.93	CR	118.39	ECR3	28.518
COV	121.00	CR	47.931	SLACK	51.903	SLACK	60.101	ATCSR	214.15	CR	168.85	COV	157.98	COV	124.54	CR	28.664
SLACK	128.14	RATCS	48.399	ATCSR	57.233	EDD	64.674	RATCS	218.89	RATCS	199.70	RATCS	171.78	RATCS	125.00	COV	31.663
RATCS	138.73	ATCSR	48.990	RATCS	58.838	OPDD	76.961	CR	238.76	ATCSR	203.49	SLACK	173.67	SLACK	125.65	ATCSR	31.970
ATCSR	148.19	COV	49.988	COV	64.271	FCFS	82.154	OPDD	245.53	SLACK	223.25	ATCSR	174.57	FCFS	127.08	RATCS	32.033
EDD	158.87	FCFS	52.206	FCFS	81.371	COV	109.10	SLACK	250.07	FCFS	248.79	FCFS	201.86	EDD	127.43	FCFS	37.387
OPDD	257.42	EDD	56.295	EDD	84.258	RATCS	109.12	FCFS	279.83	EDD	287.08	EDD	208.14	ATCSR	127.77	EDD	37.404
FCFS	280.28	OPDD	61.644	OPDD	104.48	ATCSR	110.81	EDD	298.90	OPDD	302.03	OPDD	268.68	OPDD	133.10	OPDD	87.766
WPWX	2244.1	WPWX	83.134	WPWX	143.04	WPWX	607.36	WPWX	1045.6	WPWX	1413.3	WPWX	323.80	WPWX	232.68	WPWX	203.40

Tight due dates & heavy load

Fab1		Fab2		Fab3		Fab4		Fab5		Fab6		Fab7		Fab8		Fab9	
ECR3	215.33	ECR3	48.852	ECR3	78.229	CR	70.083	COV	229.75	ECR3	225.30	ECR3	212.66	ECR3	119.62	ECR3	36.667
CR	235.63	SLACK	50.127	SLACK	81.127	SLACK	70.315	ECR3	233.24	CR	241.76	RATCS	225.84	CR	121.25	SLACK	36.702
SLACK	238.04	RATCS	51.949	CR	83.670	ECR3	70.322	RATCS	242.13	COV	261.20	SLACK	229.45	SLACK	123.54	CR	37.139
COV	243.08	ATCSR	52.194	ATCSR	88.975	EDD	73.022	SLACK	248.33	SLACK	261.51	ATCSR	231.41	RATCS	125.99	COV	39.596
RATCS	244.04	CR	53.249	RATCS	91.453	OPDD	84.649	ATCSR	248.75	RATCS	280.49	CR	232.91	COV	126.80	RATCS	39.837
ATCSR	249.10	COV	53.681	COV	102.26	FCFS	107.91	CR	264.02	ATCSR	284.62	COV	234.40	ATCSR	128.78	ATCSR	40.246
EDD	253.63	FCFS	58.488	EDD	106.77	COV	113.98	OPDD	280.38	EDD	287.69	EDD	272.09	EDD	130.79	EDD	42.458
OPDD	289.67	EDD	61.252	FCFS	120.50	RATCS	115.36	EDD	291.91	OPDD	328.21	FCFS	294.90	OPDD	131.05	FCFS	50.058
FCFS	431.12	OPDD	70.338	OPDD	130.38	ATCSR	117.51	FCFS	313.45	FCFS	397.53	OPDD	312.76	FCFS	132.02	OPDD	81.716
WPWX	2319.0	WPWX	87.254	WPWX	168.88	WPWX	607.24	WPWX	961.45	WPWX	1420.4	WPWX	388.62	WPWX	227.35	WPWX	209.04

Table 7 Summary of performance of different parameter values of ECR3 in six scenarios with respect to three objective functions

Parameter	Value	*OTD%*						T_{mean}						T_{max}					
		Moderate load level			Heavy load level			Moderate load level			Heavy load level			Moderate load level			Heavy load level		
		L	M	T	L	M	T	L	M	T	L	M	T	L	M	T	L	M	T
Y_1	5	8	5	2	8	6	5	9	9	9	9	9	8	9	9	9	9	9	9
	10	9	5	3	9	6	5	9	9	9	8	9	9	9	9	9	9	9	9
	20	9	7	6	9	7	7	8	9	8	9	7	8	8	9	8	9	7	7
	40	9	9	8	9	9	9	8	6	6	7	6	7	6	6	8	8	6	8
Y_2	0.3	9	5	8	9	8	6	9	9	9	9	9	9	9	9	9	9	9	9
	0.4	9	9	9	9	9	9	9	9	9	8	9	9	9	9	9	9	8	9
Y_3	0	5	5	6	5	5	5	5	3	4	3	3	4	5	4	7	3	3	6
	1	8	7	7	8	7	7	8	8	8	8	8	8	8	9	9	8	9	9
Y_4	0.3	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
	0.4	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

L, M, and T denote the loose, moderate, and tight due date scenarios, respectively.

Research highlights

- We propose a dispatching rule for lot scheduling in wafer fabs regarding due date-based objectives.
- The rule prioritizes lots by the impact on the total urgency of competing lots.
- The rule deals with tardy lots by a due date extension mechanism.
- We conduct extensive experiments using nine fab models, six scenarios, and nine benchmark rules.
- Our rule performs well in terms of on-time delivery rate, mean tardiness, and maximum tardiness.