processes

MDPI

*Article*

# Minimizing Tardiness Penalty Costs in Job Shop Scheduling under Maximum Allowable Tardiness

**Jae-Gon Kim [1], Hong-Bae Jun [2], June-Young Bang [3], Jong-Ho Shin [4,*] and Seong-Hoon Choi [5]**

[1] Department of Industrial and Management Engineering, Incheon National University, Incheon 22012, Korea; jaegkim@inu.ac.kr
[2] Department of Industrial Engineering, Hongik University, Seoul 04066, Korea; hongbae.jun@hongik.ac.kr
[3] Department of Industrial and Management Engineering, Sungkyul University, Gyeonggi-do 14097, Korea; jybang@sungkyul.ac.kr
[4] Department of Industrial Engineering, Chosun University, Gwangju 61452, Korea
[5] Department of Management Engineering, Sangmyung University, Chungnam 31066, Korea; shchoi@smu.ac.kr
* Correspondence: jhshin@chosun.ac.kr

Received: 19 August 2020; Accepted: 30 October 2020; Published: 2 November 2020

**Abstract:** In many manufacturing or service industries, there exists maximum allowable tardiness for orders, according to purchase contracts between the customers and suppliers. Customers may cancel their orders and request compensation for damages, for breach of contract, when the delivery time is expected to exceed maximum allowable tardiness, whereas they may accept the delayed delivery of orders with a reasonable discount of price within maximum allowable tardiness. Although many research works have been produced on the job shop scheduling problem relating to minimizing total tardiness, none of them have yet considered problems with maximum allowable tardiness. In this study, we solve a job shop scheduling problem under maximum allowable tardiness, with the objective of minimizing tardiness penalty costs. Two kinds of penalty costs are considered, i.e., one for tardy jobs, and the other for canceled jobs. To deal with this problem within a reasonable time at actual production facilities, we propose several dispatching rules by extending well-known dispatching rules for the job shop scheduling problem, in cooperation with a probabilistic conception of those rules. To evaluate the proposed rules, computational experiments were carried out on 300 test instances. The test results show that the suggested probabilistic dispatching rules work better than the existing rules and the optimization solver CPLEX, with a time limit.

**Keywords:** job shop scheduling; tardiness penalty; maximum allowable tardiness; probabilistic dispatching rules; semiconductor

## 1. Introduction

In many manufacturing or service industries, customers may specify maximum allowable tardiness for their orders in purchase contracts, according to agreements with suppliers. Here, maximum allowable tardiness is a delayed time over the due date within which the customer can allow a delayed delivery. That is, customers can cancel their orders and/or request compensation for damages for the unfulfilled contract when the delayed time over the due date exceeds maximum allowable tardiness. A typical example can be found in the semiconductor market, where the power of major customers such as Apple or Dell is continuously increasing in the supply chain. If a delivery for a customer's order is late for its due date, and within the maximum allowable tardiness, the supplier pays a penalty fee or gives price discounts to the customer for the late fulfillment of the orders. Usually, the later the order is fulfilled, the higher the penalty cost. Customers in the semiconductor market usually allow one or two weeks lateness over the due dates, although allowable tardiness mainly depends on the levels of their safety stocks. If the delivery is late over the maximum allowable tardiness, customers will cancel their orders and try to find other suppliers. In this case, the supplier may have to scrap the products of the canceled orders if there are no other customers to buy the products. This happens occasionally if the products of cancelled orders are non-commodity ones, such as application process (AP) chips or multi-chip-package chips. Thus, it is critical to have competitiveness in meeting the due dates of customers' orders in the semiconductor manufacturing industry. According to [1], individual job release times and due dates, combined with other processing environment characteristics of a semi-conductor manufacturing facility, have formed a complex job shop scheduling problem (JSP). Mason et al. [2] also mentioned that massive penalties for not satisfying the due date of orders require semiconductor manufacturers to control their manufacturing facilities efficiently. So far, most research on the job shop scheduling problem has been focused on the makespan criterion. However, due date related performance is becoming more significant in the make-to-order manufacturing environment nowadays [3].

In this study, we deal with the job shop scheduling problem with maximum allowable tardiness (JSP-MAT). In the job shop, any job consisting of a number of different operations could be performed by different machines. We were inspired by a real factory problem in a semiconductor chip packaging and test process in Korea. In the test process, three kinds of tests are performed: burn-in tests, electrical power-consumption tests, and functional speed tests. Among these tests, the electrical power consumption test and functional speed test consist of multiple routes, with sub-steps which are determined by the product types and the former test results. Furthermore, customers designate, both a normal shipping date, and a final shipping date (with late penalty) for the ordering contract. This production process can be modeled as a job shop with maximum allowable tardiness (MAT).

We study the JSP-MAT with the objective of minimizing the total penalty cost, where two kinds of penalty costs are considered, i.e., tardiness penalty costs for allowable tardy jobs, and lost-sale penalty costs for cancelled jobs. The JSP-MAT is a special case of the job shop scheduling problem, which is one of the most studied problems in the scheduling area.

Although there have been lots of studies on dispatching rules for job shop scheduling, with the objective of minimizing total tardiness, there is a lack of research works considering the two penalty costs. Furthermore, some optimal solution tools, such as CPLEX, have the limitation of not providing the optimal solutions for the large-sized job shop scheduling problem within a reasonable time. In many industries, operation managers require practical tools that can solve a large-sized problem quickly, rather than using time-consuming optimization tools. In addition, existing dispatching rules have a limitation in improving the performance of job shop operations, because fixed priorities, determined by a certain dispatching rule, are assigned to all operations so that they offer the same solution for the same problem. To overcome these limitations, we developed probabilistic dispatching rules to solve the JSP-MAT, with the objective of minimizing the total penalty cost. The proposed rules are an extended version of earliest due date (EDD), modified due date (MDD), SLACK, original cost over time (COVERT), and apparent tardiness cost (ATC), which are known to work well for the job shop tardiness scheduling problem, by considering lost-sale penalty costs as well as tardiness costs, in order to set the priority of jobs to be scheduled in a probabilistic manner.

## 2. Previous Works

In general, three kinds of solution approaches to the job shop scheduling problem exist. First, meta-heuristics, such as genetic algorithms, simulated annealing algorithms, and a tabu search have been proposed to obtain near-optimal solutions efficiently for large-scale problem instances. Second, exact algorithms, such as Lagrangian relaxation approaches, queuing network models, and branch-and-bound methods have been proposed to solve the problem to optimality. Lastly, dispatching rules have been developed to obtain good solutions quickly for practical use. Holthaus and Rajendran [4] classified the dispatching rules in a job shop into four categories: (1) process-time based rules, (2) due-date based rules, (3) combination rules, and (4) rules that are neither process-time based nor due-date based. For an extensive literature review of the job shop scheduling problem, refer to Blazewicz et al. [5].

There have been research works for the JSP in semiconductor manufacturing. For example, Gupta and Sivakumar [3] reviewed the job shop scheduling techniques in semiconductor manufacturing, including dispatching rules, mathematical programming techniques, search methods, and AI techniques. Mönch et al. [6] discussed several problems and solution techniques in semiconductor manufacturing, and addressed the challenging issues in scheduling semiconductor manufacturing operations. Several researchers mentioned the necessity, or importance, of the research of semiconductor job shop scheduling. For example, Essafi et al. [7] reported that research works dealing with the minimization of due date related objectives in job-shops are very scarce, despite the increasing importance of customer service in terms of meeting due dates. Zhang and Wu [8] also mentioned that the research on the total weighted tardiness objective is comparatively scarce, because this objective function is more difficult and complex to optimize. Due date related performance is becoming more significant in the make-to-order manufacturing environment nowadays [8].

Various algorithms have been proposed to solve the JSP, with the objective of minimizing total tardiness. Singer and Pinedo [9] proposed branch and bound algorithms, having two types of branching schemes for minimizing the total weighted tardiness in job shops. Pinedo and Singer [10] proposed a shifting bottleneck heuristic to minimize the total weighted tardiness in a job shop. To resolve the problem efficiently, in the proposed heuristic they applied a decomposition concept to break down the job shop into a number of single-machine subproblems. Asano and Ohta [11] dealt with a JSP to minimize the total weighted tardiness, with job-specific due dates and delay penalties. To resolve the problem, they proposed a heuristic algorithm based on the tree search procedure. Mattfeld and Bierwirth [12] proposed a genetic algorithm for JSPs, with release and due-dates, and with various tardiness criteria (e.g., weighted mean tardiness, maximum tardiness of jobs, and weighted number of tardy jobs) as objectives. Furthermore, Mason et al. [2] investigated three rescheduling strategies for complex job shops, such as semiconductor wafer fabs having dynamic and uncertain environments, with the objective of minimizing order total weighted tardiness. Mason et al. [1] presented a mixed integer programming (MIP) heuristic, solved by CPLEX, to minimize total weighted tardiness in a complex job shop. In order to evaluate the proposed method, they compared it with a modified shifting bottleneck heuristic and various dispatching rules. De Bontridder [13] also addressed a JSP with release times, positive end-start time lags, and a general precedence graph under the objective of minimization of the total weighted tardiness. To resolve the problem, they proposed a tabu search algorithm to search for the order in which the operations were processed on each machine. Chiang and Fu [14] reviewed eighteen dispatching rules for the JSP with due data-based objectives and proposed a new dispatching rule, having a balanced performance when more than one objective is considered. Essafi et al. [7] addressed the job-shop problem with the objective of minimizing the total weighted tardiness, considering release dates and due dates. To resolve the problem, they proposed a genetic local search algorithm. Zhou et al. [15] also dealt with a JSP to minimize weighted tardiness. They proposed a hybrid framework, integrating a genetic algorithm (GA) and four dispatching rules, i.e., weighted COVERT, Rachamadagu and Morton (RandM), weighted shortest processing time (SPT), and EDD. In their scheduling framework, the first operation of each machine was determined by the GA, and the remaining operations are allocated to the machines by a dispatching rule. They showed that the proposed hybrid framework functioned well, regardless of different levels of due-date tightness. Recently, Zhang and Wu [8] dealt with a JSP with the objective of minimizing total weighted tardiness. To solve the problem more effectively, they revised a simulated annealing algorithm with a block-based neighborhood structure. Chen and Matis [16] proposed the weighted biased modified rule (WBMR) to minimize the mean weighted tardiness of jobs in a job shop. They showed that WBMR outperformed other traditional rules at various congesting and due-date tightness levels.

However, to the best of our knowledge, the JSP with maximum allowable tardiness (JSP-MAT) has not yet been studied, while there are only a few studies for the single machine scheduling problem with maximum allowable tardiness, including [17–21]. In this study, we deal with the JSP-MAT. The objective of the problem is to minimize the total penalty cost consisting of two kinds of penalty costs, i.e., tardiness penalty costs for allowable tardy jobs, and lost-sale penalty costs for cancelled jobs. Due to the dynamic characteristic of manufacturing lines, the scheduling problem should be solved very quickly, so dispatching rules are commonly used in practice [22]. In this study, we propose dispatching rules with a probability scheme for the JSP-MAT. This paper is organized as follows: In the next section, we declare the integer programming model of the JSP-MAT. The probabilistic dispatching rules for the job shop scheduling are presented in Section 3. In Section 4, we report the performance of the suggested rules through computational experiments. Finally, Section 5 concludes the paper with a short summary.

## 3. Mathematical Formulation

In this study, we introduce the following notations to formalize the JSP-MAT.

(Parameters)

| | |
|---|---|
| $i, k$ | indices for jobs |
| $j, l$ | indices for operations |
| $h$ | index for machines |
| $N$ | number of jobs |
| $M$ | number of machines |
| $d_i$ | due date of job $i$ |
| $\overline{d}_i$ | cancellation deadline for job $i$ |
| $\overline{\tau}_i$ | maximum allowable tardiness of job $i$, i.e., $\overline{\tau}_i = \overline{d}_i - d_i$ |
| $p_{i,j}$ | processing time of operation $j$ of job $i$ |
| $V$ | a very large value |
| $\alpha_i$ | tardiness penalty cost per unit time for job $i$ |
| $\beta_i$ | lost-sale penalty cost for job $i$ |
| $m_{i,j}$ | machine on which the $j$th operation of job $i$ is processed |

(Decision variables)

| | |
|---|---|
| $y_{i,k,h}$ | = 1 if job $k$ follows job $i$ on machine $h$, and 0 otherwise |
| $c_i$ | = 1 if job $i$ is cancelled by the customer, and 0 otherwise |
| $\tau_i$ | tardiness of job $i$ |
| $s_{i,j}$ | start time of operation $j$ of job $i$ |

Then, we define a mathematical model of the JSP-MAT as follows.

(P) Minimize

$$\sum_{i=1}^{N} (\alpha_i \cdot \tau_i + \beta_i \cdot c_i) \tag{1}$$

subject to

$$s_{i,j} \geq s_{i,j-1} + p_{i,j-1} \quad \forall i, j \tag{2}$$

$$s_{i,j} + p_{i,j} \leq s_{k,l} + V \cdot (1 - y_{i,k,h}) \quad \forall i < k, \quad h \text{ with } m_{i,j} = m_{k,l} = h \tag{3}$$

$$s_{k,l} + p_{k,l} \leq s_{i,j} + V \cdot y_{i,k,h} \qquad \forall i < k, \quad h \text{ with } m_{i,j} = m_{k,l} = h \tag{4}$$

$$s_{i,j} + p_{i,j} - d_i \leq \tau_i + V \cdot c_i \quad \forall i, j \tag{5}$$

$$s_{i,j} + p_{i,j} - d_i \leq \overline{\tau}_i + V \cdot c_i \quad \forall i, j \tag{6}$$

$$s_{i,j} \geq 0 \quad \forall i, j \tag{7}$$

$$\tau_i \geq 0, c_i \in \{0,1\} \quad \forall i \tag{8}$$

$$y_{i,k,h} \in \{0,1\} \quad \forall i,j,h \tag{9}$$

Equation (1) indicates that the objective of JSP-MAT is to minimize the total penalty costs of all jobs. The costs can be represented by the sum of tardiness penalty costs for allowable tardy jobs and lost-sale penalty costs for cancelled jobs which are scheduled behind their cancelation deadlines. Figure 1 depicts the piece-wise continuous penalty cost function, which increases linearly with tardiness and jumps to the lost-sale penalty cost when the order is cancelled, and then maintains the same value after that. Constraint (2) ensures that the start time of a job is not less than the completion time of its precedent. The disjunctive constraints (3) and (4) ensure that no two jobs can be scheduled on the same machine at the same time. Note that the parameter $V$ should be a large enough value to ensure the correctness of (3) and (4). Constraints (5) and (6) identify whether each order is cancelled or not, and specify tardiness of the order in case it is not cancelled. Constraints (7), (8), and (9) represent ranges of decision variables. Since most variants of the JSP are NP-hard, JSP-MAT is also NP-hard in the ordinary sense.

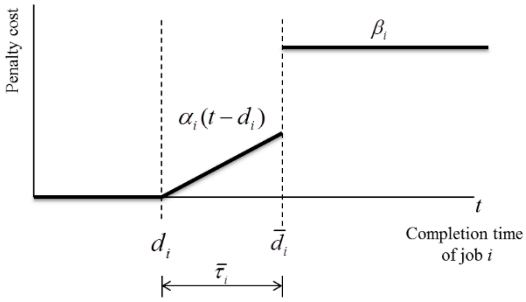

**Figure 1.** The penalty cost function.

## 4. Dispatching Rules

### *4.1. Extend Dispatching Rules*

In many industries, operation managers require practical tools that can solve large-sized operational problems quickly, rather than using time-consuming optimization tools. In this paper, we suggest several practical dispatching rules for the JSP-MAT, which can be implemented in the production planning and scheduling process at actual manufacturing facilities. Dispatching rules are used to determine the job sequence in which jobs should be worked. Various dispatching rules have been developed for the JSP [23,24], and five dispatching rules (EDD, MDD, SLACK, COVERT, and ATC) are most widely used in practice, and are known to perform better than others, in terms of the total tardiness criterion [14,25–27]. In this section, we extend these five dispatching rules to improve their performance for the JSP-MAT. For this, we introduce additional notation, as below:

$\beta'_i$ $= \beta_i/\bar{\tau}_i$
$P_i$ total processing time of job $i$
$\overline{P}$ average total processing time of the remaining jobs
$r_i$ total remaining processing time of job $i$

(1) EEDD

In the earliest due date (EDD) rule, jobs with earlier due dates have higher priorities and are processed before those with later due dates. The EDD rule that considers the weight of tardiness penalty cost can be represented by EDD: $\min_i(d_i/\alpha_i)$. For the JSP-MAT, the EDD rule can be extended by considering both the weight of tardiness penalty cost and lost-sale penalty cost. This rule is named the extended EDD (EEDD) rule in this study, and given as follows: EEDD: $\min_{i\in\Psi}(\min(d_i/\alpha_i, \bar{d}_i/\beta'_i))$, where $\Psi$ is a set of unselected jobs.

Note that $\beta_i'$ is used instead of $\beta_i$ in the proposed rule because $\beta_i'$ has the same measure as $\alpha_i$, i.e., a cost per unit time.

(2) EMDD

The modified due date (MDD) rule was originally suggested by [28] for the JSP. In this rule, the modified due date of a job is defined as the larger value of the due date of the job and the earliest possible completion time of the job. The MDD rule selects the job with the least modified due date among the unselected jobs to be processed next. The modified due dates of jobs are updated each time a new job is selected because the earliest possible completion times of the unselected jobs depend on those of selected ones. In this regard, the MDD rule is called the dynamic dispatching rule. The MDD rule that considers the weight of tardiness penalty cost can be represented by $\min_i(\max(d_i, t + r_i)/\alpha_i)$. Then the extended MDD (EMDD) rule that considers both the weight of tardiness penalty cost and lost-sale penalty cost can be represented as follows:

$$\text{EMDD} : \min_{i \in \Psi}(MD_i),\ \text{where}\ MD_i = \begin{cases} d_i/\alpha_i & \text{if } t + r_i \le d_i \\ \overline{d}_i/\beta_i' & \text{if } d_i < t + r_i \le \overline{d}_i \\ V & \text{if } t + r_i > \overline{d}_i \end{cases}$$

In the EMDD rule, priorities of jobs to be selected are determined by the three different criteria, according to the range of their earliest possible completion times. The basic concept of the EMDD rule is that a job has one of three different types of modified due dates according to the range of the earliest possible completion time of the job. In the case that the earliest possible completion time of job $i$ is not greater than $d_i$, $d_i/\alpha_i$ represents $MD_i$, i.e., the modified due date of job $i$, while $\overline{d}_i/\beta_i'$ replaces it when the earliest possible completion time exceeds the due date but not the cancellation deadline. Note that $\overline{d}_i/\beta_i'$ is usually less than $d_i/\alpha_i$ because $\beta_i'/\alpha_i > \overline{d}_i/d_i$ in general. On the other hand, a very large value $V$ is assigned to the modified due date when the job cannot help being cancelled. Once a job turns out to be cancelled, we assign it the lowest scheduling priority so as not to consider it any longer.

(3) ESLACK

In the SLACK rule, the job with the least slack time has the highest priority. The slack time of a job is defined as the maximum available time to delay the completion of the job without violating its due date. The job slack time is computed as the difference between the job due-date and the earliest possible completion time for the job. As with the MDD rule, the SLACK rule is also a dynamic dispatching rule, because the job slack time is not static, but should be updated each time a new job is selected for processing. The SLACK rule that considers the weight of tardiness penalty cost is represented by SLACK: $\min_i(d_i - t - r_i)/\alpha_i$. Then the Extended SLACK (ESLACK) rule that considers both the weight of tardiness penalty cost and lost-sale penalty cost can be represented as follows:

$$\text{ESLACK} : \min_{i \in \Psi}(ST_i),\ \text{where}\ ST_i = \begin{cases} (d_i - t - r_i)/\alpha_i & \text{if } t + r_i \le d_i \\ (\overline{d}_i - t - r_i)/\beta_i' & \text{if } d_i < t + r_i \le \overline{d}_i \\ V & \text{if } t + r_i > \overline{d}_i \end{cases}$$

(4) ECOVERT

The original cost over time (COVERT) priority index represents the expected tardiness cost per unit of imminent processing time, or cost over time [23]. The COVERT rule is a ratio-based dispatching rule, which combines the SLACK rule and the shortest processing time (SPT) rule. It puts the job with the largest COVERT ratio in the first position of the job sequence. The COVERT ratio of a job is computed by dividing a derived urgency ratio of the job by the pseudo-processing time of the job. It is known that it performs well on due date-based objectives, especially on the total tardiness measure for the JSP [29]. The COVERT rule that considers the weight of tardiness penalty cost can be represented by

COVERT: $\max_i\left\{\alpha_i/p_{ij}\cdot\max\left[0,1-\max(0,d_i-t-r_i)/k(r_i-p_{ij})\right]\right\}$. In COVERT, $k$ is a parameter, and its value is usually determined through experimental analysis. We extend the COVERT rule to ECOVERT rule by considering both the weight of tardiness penalty cost and lost-sale penalty cost as follows:

$$\text{ECOVERT}:\max_{i\in\Psi}(CR_i),\ \text{where}\quad CR_i=\begin{cases}\frac{\alpha_i}{p_{ij}}\cdot\max\left(0,1-\frac{d_i-t-r_i}{k(r_i-p_{ij})}\right) & \text{if } t+r_i\le d_i\\ \frac{\alpha_i}{p_{ij}}+\frac{\beta_i'}{p_{ij}}\cdot\max\left(0,1-\frac{\overline{d}_i-t-r_i}{k(r_i-p_{ij})}\right) & \text{if } d_i<t+r_i\le\overline{d}_i\\ -V & \text{if } t+r_i>\overline{d}_i\end{cases}$$

Note that the ECOVERT is different from the previous three dispatching rules proposed, in that it gives higher priorities to jobs with larger COVERT ratios. The COVERT ratio of job $i$, $CR_i$, has a value between 0 and $\alpha_i/p_{ij}$ if $t+r_i\le d_i$, and one between $\alpha_i/p_{ij}$ and $(\alpha_i+\beta_i')/p_{ij}$ if $d_i<t+r_i\le\overline{d}_i$, and $-V$ otherwise. Since the lost-sale penalty cost is much higher than the tardiness penalty cost, job $i$ has larger $CR_i$ as the possibility of the job being cancelled is increased.

(5) EATC

The apparent tardiness cost (ATC) rule was developed based on the COVERT rule for the JSP [23]. The basic concept of the ATC rule is the same as the COVERT rule, with two main differences. First, the ATC rule uses an exponential function rather than a linear one to emphasize the role of slack. Second, the ratio is calculated by dividing the job slack by the average processing time, instead of the job processing time. The ATC that considers the weight of tardiness penalty cost can be represented by ATC: $\max_i\left\{\alpha_i/p_{ij}\cdot\exp\left[-\max(0,d_i-t-r_i)/k\overline{P}\right]\right\}$. Here, $k$ is a parameter and $\overline{P}=\sum_{i\notin\Psi}P_i/|\{i\notin\Psi\}|$. Then the Extended ATC (EATC) can be represented as follows:

$$\text{EATC}:\max_{i\in\Psi}(TC_i),\ \text{where}\quad TC_i=\begin{cases}\frac{\alpha_i}{p_{ij}}\exp\left(\frac{-\max(0,d_i-t-r_i)}{k\overline{P}}\right) & \text{if } t+r_i\le d_i\\ \frac{\alpha_i}{p_{ij}}+\frac{\beta_i'}{p_{ij}}\exp\left(\frac{-\max(0,\overline{d}_i-t-r_i)}{k\overline{P}}\right) & \text{if } d_i<t+r_i\le\overline{d}_i\\ -V & \text{if } t+r_i>\overline{d}_i\end{cases}$$

### *4.2. Probabilistic Dispatching Rules*

The main disadvantage of the dispatching rules is that they always offer the same schedule for the same problem instance, because jobs with higher priorities are processed earlier in them, which however does not guarantee better solutions. To overcome this, we proposed probabilistic dispatching rules, based on the five dispatching rules proposed in the previous section. In the probabilistic dispatching rules, jobs with higher priorities are selected for processing with higher probabilities, which enables them to generate various solutions for the same problem instance. To assign a selection probability to each job, we use the softmax function denoted as $\Phi$ which takes a $K$-dimensional vector of arbitrary real values, and produces another $K$-dimensional vector with real values in the range (0, 1) that add up to 1.0. Let us consider a $K$-dimensional input vector $\boldsymbol{z}=(z_1,z_2,\ldots,z_K)$. Then $\Phi(z_i)=e^{z_i}/\sum_{k=1}^{K}e^{z_k}$, which can be regarded as a probability that event $i$ occurs when the value of event $i$ is $z_i$. By using the softmax function, a probability that a certain event occurs increases exponentially as the value of the event increases. In this study, probabilistic versions for the five dispatching rules proposed are denoted, respectively, as PEEDD, PEMDD, PESLACK, PECOVERT, and PEATC. Among these, PEMDD can be represented as below:

$$\text{PEMDD}:\text{Prob}(i)=e^{-MD_i}/\sum_{i\in\Psi}e^{-MD_i},\ \text{where}\quad MD_i=\begin{cases}d_i/\alpha_i & \text{if } t+r_i\le d_i\\ \overline{d}_i/\beta_i & \text{if } d_i<t+r_i\le\overline{d}_i\\ V & \text{if } t+r_i>\overline{d}_i\end{cases}$$

In the PEMDD rule, Prob($i$) represents the probability that job $i$ is selected for processing. To select a job for dispatching, we sort the $MD_i$ values of all jobs in the unselected job set $\Psi$ in a non-increasing order, and let [$i$] indicate the index of the $i$-th job in the sorting sequence. Then, job [$k$] is selected if $\sum_{i=1}^{k-1} e^{-MD_{[i]}} / \sum_{i\in\psi} e^{-MD_i} < \mathrm{U}(0,1) \leq \sum_{i=1}^{k} e^{-MD_{[i]}} / \sum_{i\in\psi} e^{-MD_i}$, for $1 \leq k \leq |\Psi|$, where U(0, 1) is a randomly selected real value with a range [0,1]. Finally, the best solution is selected for scheduling among prescribed number of replicates obtained by applying this rule iteratively.

The other four probabilistic dispatching rules PEEDD, PESLACK, PECOVERT, and PEATC can be represented in the same manner as PEMDD.

## 5. Computational Experiments

To test the performance of the suggested dispatching rules, we randomly generated 300 test instances varying the numbers of jobs, due-date tightness, and range of allowable tardiness: five levels for the number of jobs, three levels for the due-date tightness, and 20 replications for each combination. Note that currently open data sets of job shop problems are not suitable for this study because they do not contain information about the maximum allowable tardiness and the related penalty cost. Instead, we generated suitable data sets for this study based on a statistical specification from a real factory of a semiconductor chip packaging and test process in Korea. In the semiconductor chip test process, three kinds of tests are performed: burn-in tests, electrical power-consumption tests, and functional speed tests. Among these tests, the electrical power consumption test and functional speed test consist of multiple routes, with sub-steps according to the production process of the product types and their former test results. We modified the statistical specification to protect the company's industrial property right, without loss of generality, and published the full data set publicly [30].

In the generated problem, we assumed the following:

(1) Set-up times are included in the processing times.
(2) Each machine can perform at most one job at a time.
(3) Each job can be processed only on one machine at a time, without preemption.
(4) Each operation has a deterministic processing time.
(5) The transportation time between different machines can be neglected.
(6) No machine breakdown or maintenance happens.

In each problem instance, relevant data were generated as follows. Here, DU($a$, $b$) and U($a$, $b$) denote random numbers generated from the discrete and continuous uniform distributions, with a range ($a$, $b$), respectively.

(1) The number of jobs used in computational experiments is 10, 20, 30, 40, or 50.
(2) The number of machines is generated considering the number of jobs as follows: $3{\cdot}N/10$.
(3) The number of operations for each job is randomly generated with DU(1, 10). If the number of operations is bigger than the number of machines, then the number of operations is set to the number of machines.
(4) The processing time of each operation is randomly generated with DU(1, 20).
(5) The machine to take each operation of a job is randomly assigned.
(6) The due date of each job is randomly determined between 1 and 5 times of the total processing time of each job. That is, it is set to $\lambda \cdot \sum_i \sum_j p_{ij}$, where $\lambda$ is a random number between 1 and 5.
(7) There are three levels of allowable due date tightness: loose, normal, and tight. For this, cancellation deadline of job $i$ is set to $[\mu \cdot d_i]$, where $\mu$ is a factor for controlling the range of allowable due date tightness and it is set to U(1, 2) for tight allowable due dates, U(1, 3) for normal ones, and U(1, 4) for loose ones. Note that larger $\mu$ generates bigger allowable due date tardiness.
(8) Tardiness penalty cost per unit time for job $i$, i.e., $\alpha_i$, was set to U(1.0, 5.0)
(9) Lost-sale penalty cost of job $i$ is generated as follows: $\beta_i = \eta \cdot \alpha_i \cdot \overline{\tau}_i$, where $\eta$ is a random value between 5 and 15.

First, we compared five existing dispatching rules (i.e., EDD, SLACK, MDD, COVERT, and ATC) with our extended ones (i.e., EEDD, ESLACK, EMDD, ECOVERT, and EATC). Table 1 summarizes the relative deviation index (RDI) values of solutions obtained by each rule for each number of jobs. Here the RDI of solution s is calculated as below:

$$\text{RDI of solution } s = \frac{\text{objective value of solution } s - \text{objective value of the best solution}}{\text{objective value of the worst solution} - \text{objective value of the best solution}}$$

**Table 1.** Comparison of relative deviation index (RDI) values between five dispatching rules and their extended versions.

| No. of Jobs | Due Date Tightness | Original Dispatching Rules | | | | | Extended Dispatching Rules | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EDD | SLACK | MDD | COVERT | ATC | EEDD | ESLACK | EMDD | ECOVERT | EATC |
| 10 | Loose | 0.44<br>(0.38) [†] | 0.37<br>(0.39) | 0.66<br>(0.34) | 0.11<br>(0.18) | 0.24<br>(0.31) | 0.6<br>(0.39 | 0.31<br>(0.37) | 0.4<br>(0.36) | 0.06<br>(0.12) | 0.26<br>(0.38) |
| | Normal | 0.46<br>(0.38) | 0.48<br>(0.43) | 0.55<br>(0.41) | 0.29<br>(0.38) | 0.37<br>(0.42) | 0.49<br>(0.42) | 0.3<br>(0.38) | 0.31<br>(0.34) | 0.19<br>(0.32) | 0.17<br>(0.32) |
| | Tight | 0.58<br>(0.39) | 0.31<br>(0.41) | 0.66<br>(0.36) | 0.3<br>(0.39) | 0.37<br>(0.37) | 0.73<br>(0.32) | 0.25<br>(0.37) | 0.32<br>(0.35) | 0.15<br>(0.29) | 0.25<br>(0.35) |
| Average | | 0.49<br>(0.38) | 0.39<br>(0.41) | 0.62<br>(0.37) | 0.24<br>(0.34) | 0.33<br>(0.37) | 0.61<br>(0.39) | 0.29<br>(0.37) | 0.35<br>(0.35) | 0.13<br>(0.26) | 0.23<br>(0.35) |
| 20 | Loose | 0.49<br>(0.32) | 0.21<br>(0.29) | 0.65<br>(0.27) | 0.15<br>(0.23) | 0.28<br>(0.29) | 0.81<br>(0.31) | 0.15<br>(0.16) | 0.45<br>(0.3) | 0.08<br>(0.18) | 0.27<br>(0.31) |
| | Normal | 0.58<br>(0.34) | 0.24<br>(0.33) | 0.59<br>(0.35) | 0.3<br>(0.35) | 0.34<br>(0.36) | 0.65<br>(0.37) | 0.19<br>(0.3) | 0.43<br>(0.38) | 0.2<br>(0.29) | 0.23<br>(0.29) |
| | Tight | 0.63<br>(0.33) | 0.19<br>(0.23) | 0.67<br>(0.38) | 0.23<br>(0.31) | 0.34<br>(0.32) | 0.61<br>(0.34) | 0.2<br>(0.25) | 0.28<br>(0.29) | 0.07<br>(0.15) | 0.18<br>(0.26) |
| Average | | 0.57<br>(0.33) | 0.21<br>(0.28) | 0.63<br>(0.33) | 0.23<br>(0.3) | 0.32<br>(0.32) | 0.69<br>(0.35) | 0.18<br>(0.24) | 0.39<br>(0.33) | 0.12<br>(0.22) | 0.23<br>(0.28) |
| 30 | Loose | 0.54<br>(0.26) | 0.22<br>(0.27) | 0.79<br>(0.22) | 0.19<br>(0.18) | 0.34<br>(0.33) | 0.79<br>(0.28) | 0.18<br>(0.22) | 0.37<br>(0.29) | 0.08<br>(0.14) | 0.18<br>(0.16) |
| | Normal | 0.57<br>(0.37) | 0.29<br>(0.3) | 0.74<br>(0.29) | 0.27<br>(0.32) | 0.31<br>(0.25) | 0.63<br>(0.36) | 0.13<br>(0.19) | 0.26<br>(0.19) | 0.13<br>(0.24) | 0.19<br>(0.21) |
| | Tight | 0.54<br>(0.31) | 0.27<br>(0.33) | 0.78<br>(0.28) | 0.16<br>(0.23) | 0.46<br>(0.38) | 0.6<br>(0.31) | 0.2<br>(0.25) | 0.26<br>(0.27) | 0.21<br>(0.31) | 0.24<br>(0.33) |
| Average | | 0.55<br>(0.31) | 0.26)<br>(0.3) | 0.77<br>(0.26) | 0.21<br>(0.25) | 0.37<br>(0.32) | 0.67<br>(0.32 | 0.17<br>(0.22) | 0.3<br>(0.25) | 0.14<br>(0.24) | 0.2<br>(0.24) |
| 40 | Loose | 0.69<br>(0.24) | 0.33<br>(0.31) | 0.87<br>(0.19) | 0.24<br>(0.27) | 0.4<br>(0.31) | 0.73<br>(0.23) | 0.24<br>(0.21) | 0.44<br>(0.27) | 0.07<br>(0.16) | 0.11<br>(0.14) |
| | Normal | 0.59<br>(0.3) | 0.31<br>(0.28) | 0.83<br>(0.2) | 0.24<br>(0.29) | 0.31<br>(0.29) | 0.67<br>(0.33) | 0.25<br>(0.27) | 0.39<br>(0.31) | 0.13<br>(0.26) | 0.08<br>(0.17) |
| | Tight | 0.66<br>(0.3) | 0.3<br>(0.24) | 0.86<br>(0.21) | 0.29<br>(0.26) | 0.37<br>(0.33) | 0.69<br>(0.23) | 0.21<br>(0.26) | 0.4<br>(0.33) | 0.19<br>(0.25) | 0.14<br>(0.16) |
| Average | | 0.65<br>(0.28) | 0.31<br>(0.28) | 0.85<br>(0.2) | 0.26<br>(0.27) | 0.36<br>(0.31) | 0.7<br>(0.26) | 0.23<br>(0.24) | 0.41<br>(0.3) | 0.13<br>(0.23) | 0.11<br>(0.16) |
| 50 | Loose | 0.59<br>(0.24) | 0.25<br>(0.3) | 0.74<br>(0.24) | 0.35<br>(0.35) | 0.5<br>(0.39) | 0.77<br>(0.3) | 0.19<br>(0.16) | 0.3<br>(0.21) | 0.14<br>(0.16) | 0.17<br>(0.24) |
| | Normal | 0.61<br>(0.24) | 0.29<br>(0.28) | 0.84<br>(0.24) | 0.18<br>(0.19) | 0.39<br>(0.31) | 0.76<br>(0.27) | 0.21<br>(0.18) | 0.24<br>(0.14) | 0.07<br>(0.11) | 0.08<br>(0.09) |
| | Tight | 0.71<br>(0.28) | 0.27<br>(0.26) | 0.83<br>(0.18) | 0.22<br>(0.23) | 0.26<br>(0.16) | 0.76<br>(0.27) | 0.19<br>(0.24) | 0.33<br>(0.23) | 0.05<br>(0.09) | 0.14<br>(0.17) |
| | Average | 0.64<br>(0.26) | 0.27<br>(0.28) | 0.8<br>(0.22) | 0.25<br>(0.27) | 0.38<br>(0.31) | 0.76(0.28) | 0.2<br>(0.19) | 0.29<br>(0.2) | 0.08<br>(0.13) | 0.13<br>(0.18) |
| Average | | 0.58<br>(0.31) | 0.29<br>(0.31) | 0.74<br>(0.28) | 0.24<br>(0.28) | 0.35<br>(0.32) | 0.69<br>(0.32) | 0.21<br>(0.25) | 0.35<br>(0.28) | 0.12<br>(0.21) | 0.18<br>(0.24) |

[†] Average and standard deviation (in parenthesis) of the RDI values.

Note that the range of RDI is between 0 and 1, where 0 and 1 denote the best and the worst solutions, respectively.

The results of Table 1 tell us the following: first, most of the extended dispatching rules gave better performance than the original dispatching rules, although the performance of EEDD was not better than that of EDD. Second, overall, ECOVERT outperformed the other rules, irrespective of the number of jobs and levels of due date tightness. Table 2 shows the number of best solutions (NBS) found by each rule. In Table 2, we can also see similar results: first, many of the extended dispatching rules, such as EMDD, ECOVERT, and EATC, gave better results than the original dispatching rules. However, EEDD and ESLACK gave slightly worse results than EDD and SLACK. Second, ECOVERT found a total of 126 best solutions, which was overwhelming in comparison with the others. ECOVERT always found the highest number of best solutions among all the rules.

**Table 2.** Number of the best solutions found by five dispatching rules and their extended versions.

| No. of Jobs | Due Date Tightness | Original Dispatching Rules | | | | | Extended Dispatching Rules | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EDD | SLACK | MDD | COVERT | ATC | EEDD | ESLACK | EMDD | ECOVERT | EATC |
| | loose | 3 | 4 | 0 | 6 | 5 | 1 | 4 | 1 | 10 | 6 |
| 10 | normal | 3 | 5 | 3 | 6 | 7 | 3 | 4 | 5 | 7 | 6 |
| | tight | 3 | 6 | 3 | 7 | 5 | 1 | 3 | 4 | 13 | 8 |
| | subtotal | 9 | 15 | 6 | 19 | 17 | 5 | 11 | 10 | 30 | 20 |
| | loose | 0 | 4 | 0 | 8 | 1 | 0 | 4 | 0 | 12 | 2 |
| 20 | normal | 0 | 5 | 1 | 4 | 2 | 0 | 5 | 1 | 6 | 2 |
| | tight | 0 | 4 | 1 | 5 | 0 | 0 | 3 | 1 | 10 | 3 |
| | subtotal | 0 | 13 | 2 | 17 | 3 | 0 | 12 | 2 | 28 | 7 |
| | loose | 0 | 4 | 0 | 4 | 1 | 0 | 4 | 1 | 9 | 1 |
| 30 | normal | 0 | 2 | 0 | 2 | 0 | 0 | 5 | 1 | 10 | 1 |
| | tight | 0 | 4 | 0 | 5 | 2 | 0 | 3 | 0 | 3 | 4 |
| | subtotal | 0 | 10 | 0 | 11 | 3 | 0 | 12 | 2 | 22 | 6 |
| | loose | 0 | 1 | 0 | 3 | 1 | 0 | 1 | 0 | 12 | 5 |
| 40 | normal | 1 | 3 | 0 | 6 | 1 | 0 | 2 | 0 | 6 | 4 |
| | tight | 1 | 1 | 0 | 3 | 3 | 0 | 2 | 1 | 7 | 2 |
| | subtotal | 2 | 5 | 0 | 12 | 5 | 0 | 5 | 1 | 25 | 11 |
| | loose | 0 | 5 | 0 | 1 | 2 | 0 | 2 | 1 | 5 | 4 |
| 50 | normal | 0 | 2 | 0 | 5 | 1 | 0 | 1 | 0 | 6 | 5 |
| | tight | 0 | 1 | 0 | 2 | 0 | 0 | 4 | 1 | 10 | 3 |
| | subtotal | 0 | 8 | 0 | 8 | 3 | 0 | 7 | 2 | 21 | 12 |
| Total | | 11 | 51 | 8 | 67 | 31 | 5 | 47 | 17 | 126 | 56 |

Next, we compared the extended dispatching rules with the probabilistic dispatching rules (i.e., PEEDD, PEMDD, PESLACK, PECOVERT, and PEATC) and solutions from CPLEX 12.1, the commercial optimization solver. In CPLEX, we set the time limit to solve the JSP-MAT as 200 s, and set the parameter *mipemphasis* as CPXMIPEMPHASIS_FEASIBILITY to obtain a good and feasible solution rather than an optimal solution within the short time limit. In ECOVERT and EATC rules, the parameter $k$ was set to 2. For each probabilistic dispatching rule, the best solution was selected among 1000 replicates, obtained by applying each rule iteratively.

Table 3 shows the NBS found by each rule for each job size. Overall, the probabilistic dispatching rules found more NBS than the extended dispatching rules did. As the number of jobs increased, it seemed that NBS of most rules became worse. However, PECOVERT and PESLACK gave the best solutions, irrespective of the number of jobs. It was also observed that EDD based rules, such as EEDD and PEEDD, gave the worse performance. Table 4 indicates the averages and standard deviations of RDI values for extended dispatching rules and their probabilistic rules, compared with the solutions of CPLEX 12.1. The test results in Table 4 show that the PECOVERT rule outperformed the other rules. In terms of the solution quality, the result shows that as the number of jobs increased, the RDI values of most dispatching rules also increased, until the number of jobs was 30, which means that the number of jobs highly affected the performance of the dispatching rules in a negative way. However, from a number above 40, those of most dispatching rules seemed to be decreasing. CPLEX's solutions outperformed the other dispatching rules' solutions when the number of jobs was 10 or 20. However, CPLEX's RDI values tended to increase rapidly as the number of jobs increased. Especially when the number was over 40; it was relatively bad compared to those of ECOVERT or PECOVERT. It was also observed that the degree of due date tightness did not seem to affect the performance of solutions of the dispatching rules. Overall, the probabilistic extended dispatching rules were much better than the extended dispatching rules. Among the suggested rules, PECOVERT and PESLACK were the best, followed by PEATC. In particular, the PECOVERT rule gave the best solutions, irrespective of the number of jobs. The outstanding performance of the PECOVERT rule became more evident when the job size became larger, as shown in Figure 2. In summary, we could see that PECOVERT gave a robust performance, and its solution quality was also good when compared to the solutions of CPLEX, in particular, when the number of jobs was over 30.

**Table 3.** Number of the best solutions found by extended dispatching rules and their probabilistic versions.

| No. of Jobs | Due Date Tightness | Extended Dispatching Rules | | | | | Probabilistic Dispatching Rules | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EEDD | ESLACK | EMDD | ECOVERT | EATC | PEEDD | PESLACK | PEMDD | PECOVERT | PEATC |
| | loose | 0 | 1 | 0 | 1 | 2 | 3 | 4 | 4 | 3 | 4 |
| 10 | normal | 1 | 2 | 3 | 1 | 2 | 3 | 6 | 2 | 3 | 5 |
| | tight | 0 | 2 | 1 | 3 | 3 | 4 | 8 | 4 | 4 | 6 |
| | subtotal | 1 | 5 | 4 | 5 | 7 | 10 | 18 | 10 | 10 | 15 |
| | loose | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 1 | 2 | 1 |
| 20 | normal | 0 | 1 | 0 | 2 | 1 | 1 | 2 | 1 | 2 | 2 |
| | tight | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 1 |
| | subtotal | 0 | 1 | 0 | 3 | 2 | 1 | 5 | 2 | 7 | 4 |
| | loose | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 30 | normal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | tight | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | subtotal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | loose | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 3 | 0 |
| 40 | normal | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| | tight | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | subtotal | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 6 | 0 |
| | loose | 0 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 5 | 0 |
| 50 | normal | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 11 | 0 |
| | tight | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 4 | 0 |
| | subtotal | 0 | 0 | 1 | 4 | 3 | 0 | 3 | 0 | 20 | 0 |
| Total | | 1 | 6 | 5 | 14 | 12 | 11 | 27 | 12 | 45 | 19 |

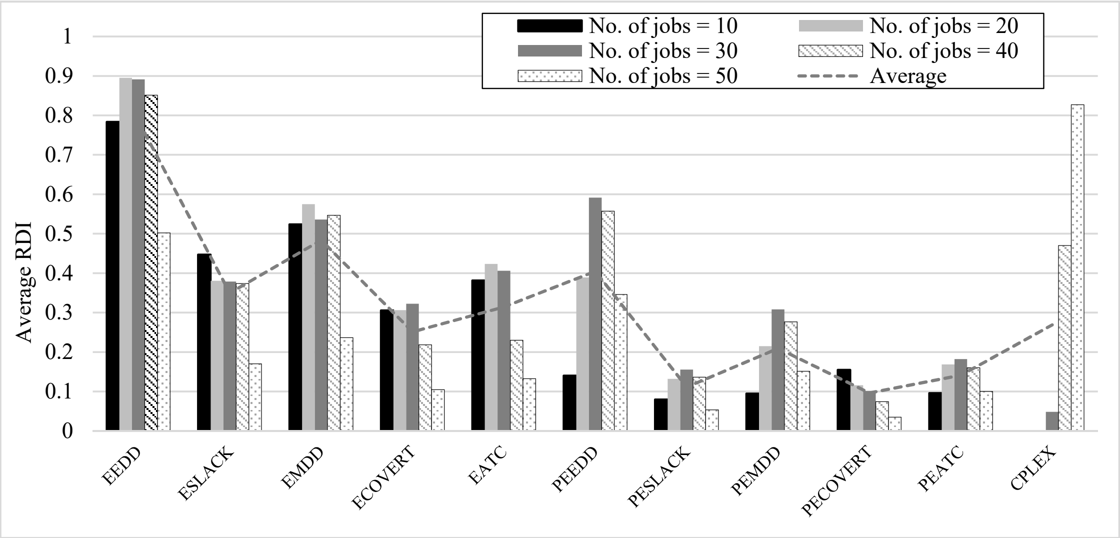

**Figure 2.** The average relative deviation index values of extended dispatching rules and their probabilistic ones.

To see if there was any statistically significant difference in the performances of the dispatching rules and CPLEX, statistical comparison tests were carried out. Prior to the comparison between two methods, the normality test with significance level 0.01 was carried out. As a result, normality was not guaranteed for all cases, so that a non-parametric method for the comparison of two groups, the Wilcoxon signed rank test with significance level 0.01, was carried out instead of paired *t*-tests. As per the summarized results shown in Table 5, we see that the rank is determined as follows PECOVERT > PESLACK > PEATC > CPLEX, ECOVERT, PEMDD > EATC > ESLACK > PEEDD > EMDD > EEDD. Overall, the proposed PECOVERT rule gave the best performance among tested dispatching rules. We also found that there was no significant difference in the results of the following two rules: CPLEX, ECOVERT, and PEMDD.

**Table 4.** Comparison of RDI values between extended dispatching rules and their probabilistic versions.

| No. of Jobs | Due Date Tightness | Extended Dispatching Rules | | | | | Probabilistic Dispatching Rules | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EEDD | ESLACK | EMDD | ECOVERT | EATC | PEEDD | PESLACK | PEMDD | PECOVERT | PEATC | CPLEX * |
| 10 | Loose | 0.77<br>(0.32) † | 0.45<br>(0.34) | 0.59<br>(0.36) | 0.24<br>(0.23) | 0.41<br>(0.38) | 0.17<br>(0.14) | 0.08<br>(0.08) | 0.1<br>(0.08) | 0.13<br>(0.14) | 0.08<br>(0.08) | 0<br>0 |
| | Normal | 0.7<br>(0.43) | 0.49<br>(0.38) | 0.51<br>(0.39) | 0.33<br>(0.35) | 0.34<br>(0.35) | 0.15<br>(0.16) | 0.1<br>(0.13) | 0.1<br>(0.12) | 0.13<br>(0.15) | 0.10<br>(0.13) | 0<br>0 |
| | Tight | 0.88<br>(0.23) | 0.39<br>(0.39) | 0.47<br>(0.37) | 0.34<br>(0.4) | 0.41<br>(0.38) | 0.1<br>(0.12) | 0.07<br>(0.08) | 0.08<br>(0.08) | 0.2<br>(0.3) | 0.12<br>(0.18) | 0<br>0 |
| Average | | 0.78<br>(0.32) | 0.45<br>(0.37) | 0.52<br>(0.37) | 0.31<br>(0.33) | 0.38<br>(0.37) | 0.14<br>(0.14) | 0.08<br>(0.1) | 0.1<br>(0.09) | 0.16<br>(0.2) | 0.10<br>(0.13) | 0<br>0 |
| 20 | Loose | 0.93<br>(0.18) | 0.32<br>(0.25) | 0.58<br>(0.25) | 0.23<br>(0.18) | 0.41<br>(0.28) | 0.45<br>(0.23) | 0.17<br>(0.11) | 0.24<br>(0.12) | 0.1<br>(0.09) | 0.17<br>(0.13) | 0<br>0 |
| | Normal | 0.83<br>(0.26) | 0.4<br>(0.3) | 0.6<br>(0.32) | 0.39<br>(0.27) | 0.42<br>(0.28) | 0.4<br>(0.24) | 0.14<br>(0.1) | 0.23<br>(0.13) | 0.12<br>(0.1) | 0.17<br>(0.12) | 0<br>0 |
| | Tight | 0.92<br>(0.17) | 0.43<br>(0.29) | 0.55<br>(0.31) | 0.31<br>(0.21) | 0.44<br>(0.3) | 0.32<br>(0.2) | 0.1<br>(0.07) | 0.18<br>(0.12) | 0.13<br>(0.12) | 0.16<br>(0.13) | 0<br>0 |
| Average | | 0.9<br>(0.2) | 0.38<br>(0.28) | 0.58<br>(0.29) | 0.31<br>(0.22) | 0.42<br>(0.28) | 0.39<br>(0.22) | 0.13<br>(0.1) | 0.22<br>(0.12) | 0.12<br>(0.1) | 0.17<br>(0.13) | 0<br>0 |
| 30 | Loose | 0.91<br>(0.17) | 0.35<br>(0.21) | 0.54<br>(0.26) | 0.26<br>(0.18) | 0.36<br>(0.23) | 0.6<br>(0.21) | 0.18<br>(0.09) | 0.37<br>(0.14) | 0.11<br>(0.1) | 0.20<br>(0.12) | 0.03<br>(0.07) |
| | Normal | 0.9<br>(0.16) | 0.37<br>(0.26) | 0.57<br>(0.29) | 0.32<br>(0.27) | 0.42<br>(0.28) | 0.62<br>(0.23) | 0.17<br>(0.11) | 0.32<br>(0.13) | 0.09<br>(0.07) | 0.18<br>(0.11) | 0.09<br>(0.24) |
| | Tight | 0.86<br>(0.21) | 0.42<br>(0.31) | 0.5<br>(0.3) | 0.39<br>(0.32) | 0.44<br>(0.35) | 0.56<br>(0.24) | 0.12<br>(0.12) | 0.24<br>(0.16) | 0.1<br>(0.1) | 0.16<br>(0.12) | 0.02<br>(0.07) |
| Average | | 0.9<br>(0.18) | 0.38<br>(0.26) | 0.54<br>(0.28) | 0.32<br>(0.25) | 0.41<br>(0.29) | 0.59<br>(0.23) | 0.16<br>(0.11) | 0.31<br>(0.14) | 0.1<br>(0.1) | 0.18<br>(0.12) | 0.05<br>(0.13) |
| 40 | Loose | 0.89<br>(0.18) | 0.39<br>(0.23) | 0.58<br>(0.22) | 0.18<br>(0.22) | 0.22<br>(0.17) | 0.67<br>(0.2) | 0.21<br>(0.12) | 0.39<br>(0.15) | 0.06<br>(0.11) | 0.19<br>(0.10) | 0.59<br>(0.41 |
| | Normal | 0.89<br>(0.23) | 0.38<br>(0.27) | 0.56<br>(0.3) | 0.23<br>(0.3) | 0.22<br>(0.28) | 0.48<br>(0.24) | 0.11<br>(0.12) | 0.26<br>(0.18) | 0.08<br>(0.19) | 0.12<br>(0.10) | 0.39<br>(0.4) |
| | Tight | 0.77<br>(0.32) | 0.35<br>(0.3) | 0.5<br>(0.38) | 0.24<br>(0.25) | 0.25<br>(0.24) | 0.52<br>(0.29) | 0.08<br>(0.12) | 0.18<br>(0.15) | 0.09<br>(0.13) | 0.17<br>(0.15) | 0.43<br>(0.39) |
| Average | | 0.85<br>(0.24) | 0.37<br>(0.27) | 0.55<br>(0.3) | 0.22<br>(0.26) | 0.23<br>(0.23) | 0.56<br>(0.24) | 0.14<br>(0.12) | 0.28<br>(0.16) | 0.07<br>(0.14) | 0.16<br>(0.12) | 0.47<br>(0.4) |
| 50 | Loose | 0.64<br>(0.39) | 0.18<br>(0.17) | 0.26<br>(0.24) | 0.13<br>(0.12) | 0.19<br>(0.24) | 0.45<br>(0.3) | 0.1<br>(0.11) | 0.2<br>(0.17) | 0.01<br>(0.03) | 0.11<br>(0.08) | 0.78<br>(0.3) |
| | Normal | 0.39<br>(0.38) | 0.14<br>(0.18) | 0.16<br>(0.15) | 0.08<br>(0.12) | 0.07<br>(0.1) | 0.31<br>(0.31) | 0.04<br>(0.06) | 0.14<br>(0.15) | 0.06<br>(0.22) | 0.12<br>(0.19) | 0.89<br>(0.25) |
| | Tight | 0.48<br>(0.46) | 0.19<br>(0.28) | 0.29<br>(0.34) | 0.1<br>(0.15) | 0.14<br>(0.18) | 0.27<br>(0.27) | 0.02<br>(0.04) | 0.12<br>(0.12) | 0.03<br>(0.07) | 0.08<br>(0.11) | 0.8<br>(0.33) |
| Average | | 0.5<br>(0.41) | 0.17<br>(0.21) | 0.24<br>(0.24) | 0.1<br>(0.13) | 0.13<br>(0.17) | 0.35<br>(0.29) | 0.05<br>(0.07) | 0.15<br>(0.15) | 0.04<br>(0.11) | 0.10<br>(0.12) | 0.83<br>(0.29) |
| Average | | 0.78<br>(0.27) | 0.35<br>(0.28) | 0.48<br>(0.3) | 0.25<br>(0.24) | 0.31<br>(0.27) | 0.4<br>(0.23) | 0.11<br>(0.1) | 0.21<br>(0.13) | 0.1<br>(0.13) | 0.14<br>(0.12) | 0.27<br>(0.16) |

† Average and standard deviation (in parenthesis) of the RDI values. * CPLEX gives optimal solutions within 200 s.

**Table 5.** Results of the Wilcoxon signed rank test.

| Rule | ARDI * | Result † | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| EEDD | 0.7849 | A | | | | | | | | |
| ESLACK | 0.3501 | | B | | | | | | | |
| EMDD | 0.4837 | | | C | | | | | | |
| ECOVERT | 0.2515 | | | | D † | | | | | |
| EATC | 0.3147 | | | | | E | | | | |
| PEEDD | 0.4048 | | | | | | F | | | |
| PESLACK | 0.1115 | | | | | | | G | | |
| PEMDD | 0.2093 | | | | D † | | | | | |
| PECOVERT | 0.0963 | | | | | | | | H | |
| PEATC | 0.1414 | | | | | | | | | I |
| CPLEX | 0.2690 | | | | D † | | | | | |

† Rules with the same letter are statistically indifferent at the significance level of 0.01; * Average relative deviation index.

To assess the efficiency of PECOVERT and CPLEX, the computational times of PECOVERT and CPLEX for 300 test problems are reported in Table 6. The test results show that CPLEX found optimal solutions for all the 10-job test instances within 0.5. However, the computational times of CPLEX exponentially increased as the number of jobs increased. CPLEX frequently timed out (200 s) in solving the large-sized problems. While the computational times of PECOVERT are very short (about 2 s or less), even in the cases with more than 40 jobs. This means that PECOVERT efficiently solves the large-sized JSP-MAT within a reasonable time, in practice.

**Table 6.** Results of the computational times.

| No. of Jobs | Due Date Tightness | PECOVERT | CPLEX |
|---|---|---|---|
| 10 | Loose | 0.010 | 0.500 |
| | Normal | 0.010 | 0.230 |
| | Tight | 0.010 | 0.209 |
| | Average | 0.010 | 0.313 |
| 20 | Loose | 0.136 | 81.668 |
| | Normal | 0.132 | 65.898 |
| | Tight | 0.128 | 56.869 |
| | Average | 0.132 | 68.145 |
| 30 | Loose | 0.550 | 173.445 |
| | Normal | 0.536 | 168.727 |
| | Tight | 0.528 | 178.542 |
| | Average | 0.538 | 173.571 |
| 40 | Loose | 1.231 | 200.062 |
| | Normal | 1.113 | 200.294 |
| | Tight | 1.175 | 198.404 |
| | Average | 1.173 | 199.587 |
| 50 | Loose | 2.140 | 200.461 |
| | Normal | 2.184 | 200.497 |
| | Tight | 2.075 | 200.631 |
| | Average | 2.133 | 200.529 |
| | Average | 0.797 | 128.429 |

## 6. Conclusions

In this study, we dealt with the JSP-MAT, with the objective of minimizing tardiness penalty costs, consisting of penalty costs for allowable tardy jobs and those for cancelled jobs. We proposed probabilistic dispatching rules for the JSP-MAT by extending five well-known dispatching rules for the JSP, and adding a probabilistic concept into them. We conducted computational experiments to test the performance of the proposed rules. The test results showed that the extended dispatching rules outperformed the original dispatching rules, and the probabilistic dispatching rules work better than the extended dispatching rules. Among the proposed rules, PECOVERT showed the best performance; its solution quality was as good as CPLEX, despite its fast execution. Compared to CPLEX, the larger the number of jobs, the worse the CPLEX solution was, while PECOVRT gave stable and good solutions.

The suggested dispatching rules, especially PECOVERT, can be used as a viable tool for machine scheduling in the manufacturing industries which have penalty costs for order cancellation due to tardiness, such as the semiconductor manufacturing industry. They also can be used to develop an order promising procedure in the semiconductor business. The demand management system of a semiconductor manufacturing company receives orders from customers, and gives back the information on the estimated completion times and the quantities of products that can be shipped within due dates. After several runs of order promising negotiation, the product quantities and the due dates of the orders are confirmed, and only the confirmed orders will be processed by a production system. During the negotiation, the JSP-MAT can be solved using the suggested dispatching rules to estimate the completion times of the orders, and to decide whether the orders should be accepted or not, by considering the maximum allowable tardiness from the customers.

In conclusion, although the probabilistic extended dispatching rules do not outperform the solutions of a commercial optimization solver in all cases, we believe that they could give satisfactory results, as good as the optimization solver's solutions. Furthermore, due to the simplicity and speed of the probabilistic extended dispatching rules, they could be easily used in practice and in simulation models of semiconductor production scheduling. For future research, it is necessary to study modifications and adaptations of more sophisticated heuristics or dispatching rules, which are known as relatively good algorithms for the single machine total tardiness scheduling problem, for further research of the JSP-MAT.

**Author Contributions:** Conceptualization and Idealization, J.-G.K.; Data analysis, H.-B.J.; Data preparation, S.-H.C. and J.-Y.B.; Review and editing, J.-H.S. All authors have read and agreed to the published version of the manuscript .

**Funding:** This research was funded by Incheon National University (International Cooperative) Research Grant in 2019.

**Acknowledgments:** This work was supported by Incheon National University (International Cooperative) Research Grant in 2019.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mason, S.J.; Fowler, J.W.; Carlyle, W.M.; Montgomery, D.C. Heuristics for minimizing total weighted tardiness in complex job shops. *Int. J. Prod. Res.* **2005**, *43*, 1943–1963. [CrossRef]
2. Mason, S.J.; Jin, S.; Wessels, C.M. Rescheduling strategies for minimizing total weighted tardiness in complex job shops. *Int. J. Prod. Res.* **2004**, *42*, 613–628. [CrossRef]
3. Gupta, A.K.; Sivakumar, A.I. Job shop scheduling techniques in semiconductor manufacturing. *Int. J. Adv. Manuf. Technol.* **2006**, *27*, 1163–1169. [CrossRef]
4. Holthaus, O.; Rajendran, C. Efficient dispatching rules for scheduling in a job shop. *Int. J. Prod. Econ.* **1997**, *48*, 87–105. [CrossRef]
5. Błażewicz, J.; Domschke, W.; Pesch, E. The job shop scheduling problem: Conventional and new solution techniques. *Eur. J. Oper. Res.* **1996**, *93*, 1–33. [CrossRef]
6. Moench, L.; Fowler, J.W.; Dauzère-Pérès, S.; Mason, S.J.; Rose, O. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *J. Sched.* **2011**, *14*, 583–599. [CrossRef]
7. Essafi, I.; Mati, Y.; Dauzère-Pérès, S. A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. *Comput. Oper. Res.* **2008**, *35*, 2599–2616. [CrossRef]
8. Zhang, R.; Wu, C. A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective. *Comput. Oper. Res.* **2011**, *38*, 854–867. [CrossRef]
9. Singer, M.; Pinedo, M. A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. *IIE Trans.* **1998**, *30*, 109–118. [CrossRef]
10. Pinedo, M.; Singer, M. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Nav. Res. Logist.* **1999**, *46*, 1–17. [CrossRef]
11. Asano, M.; Ohta, H. A heuristic for job shop scheduling to minimize total weighted tardiness. *Comput. Ind. Eng.* **2002**, *42*, 137–147. [CrossRef]
12. Mattfeld, D.C.; Bierwirth, C. An efficient genetic algorithm for job shop scheduling with tardiness objectives. *Eur. J. Oper. Res.* **2004**, *155*, 616–630. [CrossRef]
13. De Bontridder, K.M.J. Minimizing Total Weighted Tardiness in a Generalized Job Shop. *J. Sched.* **2005**, *8*, 479–496. [CrossRef]
14. Chiang, T.C.; Fu, L.C. Using dispatching rules for job shop scheduling with due date-based objectives. *Int. J. Prod. Res.* **2007**, *45*, 3245–3262. [CrossRef]
15. Zhou, H.; Cheung, W.; Leung, L.C. Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. *Eur. J. Oper. Res.* **2009**, *194*, 637–649. [CrossRef]
16. Chen, B.; Matis, T.I. A flexible dispatching rule for minimizing tardiness in job shop scheduling. *Int. J. Prod. Econ.* **2013**, *141*, 360–365. [CrossRef]
17. Chand, S.; Schneeberger, H. A note on the single-machine scheduling problem with minimum weighted completion time and maximum allowable tardiness. *Nav. Res. Logist. Q.* **1986**, *33*, 551–557. [CrossRef]

18. Seo, J.H.; Kim, C.-B.; Lee, D.H. Minimizing mean squared deviation of completion times with maximum tardiness constraint. *Eur. J. Oper. Res.* **2001**, *129*, 95–104. [CrossRef]
19. Mönch, L.; Unbehaun, R.; Choung, Y.I. Minimizing earliness–tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint. *OR Spectr.* **2006**, *28*, 177–198. [CrossRef]
20. Mazdeh, M.M.; Nakhjavani, A.K.; Zareei, A. Minimizing total weighted tardiness with drop dead dates in single machine scheduling problem. *Int. J. Ind. Eng. Prod. Res.* **2010**, *21*, 89–95.
21. Kim, J.-G.; Bang, J.-Y.; Lim, S.-K.; Lee, J.-Y. Priority Rules for the Single Machine Total Weighted Tardiness Scheduling with Maximum Allowable Tardiness. *Green Smart Technol.* **2017**, *141*, 31–38. [CrossRef]
22. Dominic, P.D.D.; Kaliyamoorthy, S.; Kumar, M.S. Efficient dispatching rules for dynamic job shop scheduling. *Int. J. Adv. Manuf. Technol.* **2004**, *24*, 70–75. [CrossRef]
23. Vepsalainen, A.P.J.; Morton, T.E. Priority Rules for Job Shops with Weighted Tardiness Costs. *Manag. Sci.* **1987**, *33*, 1035–1047. [CrossRef]
24. Koulamas, C. The single-machine total tardiness scheduling problem: Review and extensions. *Eur. J. Oper. Res.* **2010**, *202*, 1–7. [CrossRef]
25. Kim, Y.-D. A comparison of dispatching rules for job shops with multiple identical jobs and alternative routeings. *Int. J. Prod. Res.* **1990**, *28*, 953–962. [CrossRef]
26. Kim, Y.-D. A backward approach in list scheduling algorithms for multi-machine tardiness problems. *Comput. Oper. Res.* **1995**, *22*, 307–319. [CrossRef]
27. Moon, D.H.; Christy, D.P. A simulation study for dynamic scheduling in a hybrid assembly/job shop considering the JIT context. *Prod. Plan. Control.* **1998**, *9*, 532–541. [CrossRef]
28. Baker, K.R.; Bertrand, J. A dynamic priority rule for scheduling against due-dates. *J. Oper. Manag.* **1982**, *3*, 37–42. [CrossRef]
29. Russell, R.S.; Dar-El, E.M.; Taylor, B.W. A comparative analysis of the COVERT job sequencing rule using various shop performance measures. *Int. J. Prod. Res.* **1987**, *25*, 1523–1540. [CrossRef]
30. Data Sets for Job Shop Problems with Maximum Allowable Tardiness. Available online: http://bit.ly/2WbPPui (accessed on 2 November 2020).

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.