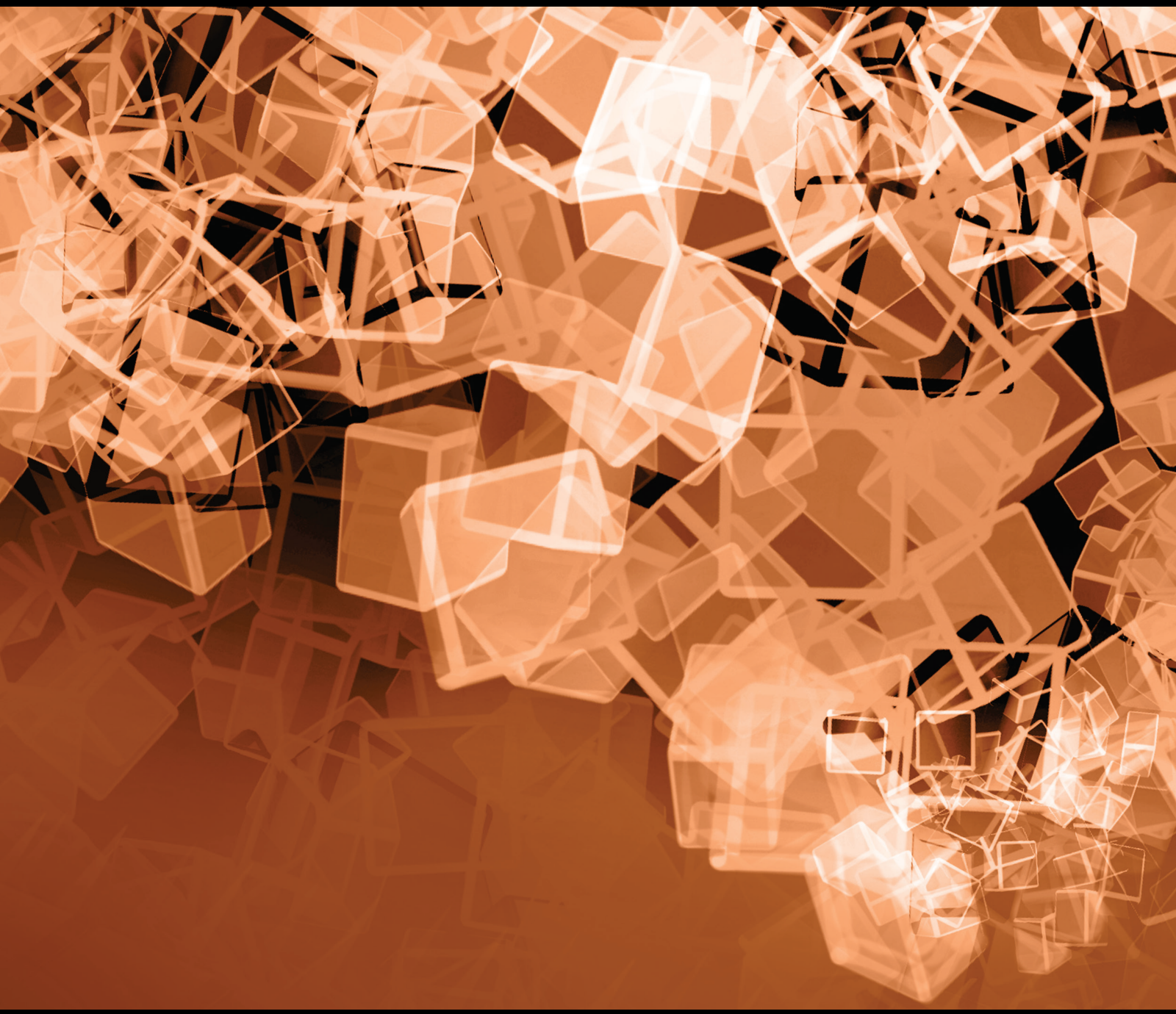
Discrete Dynamics in Nature and Society

# Theory, Applications, and Solutions in Scheduling Research

Lead Guest Editor: Yunqiang Yin
Guest Editors: Dujuan Wang, T. C. E. Cheng, and Chin-Chia Wu



# Theory, Applications, and Solutions in Scheduling Research

Discrete Dynamics in Nature and Society

# Theory, Applications, and Solutions in Scheduling Research

Lead Guest Editor: Yunqiang Yin
Guest Editors: Dujuan Wang, T. C. E. Cheng, and Chin-Chia Wu

# Contents

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 8625849, 9 pages
https://doi.org/10.1155/2020/8625849

*Research Article*

# Integrated Scheduling Problem on a Single Bounded Batch Machine with an Unavailability Constraint

**Jing Fan**

*College of Arts and Science, Shanghai Polytechnic University, Shanghai 201209, China*

Correspondence should be addressed to Jing Fan; sspu_fj@163.com

Received 28 July 2020; Revised 25 October 2020; Accepted 5 November 2020; Published 1 December 2020

Academic Editor: Chin-Chia Wu



We consider a scheduling problem where a set of jobs are first processed on a machine with an unavailability interval and, then, delivered to the customer directly. We focus on an integrated schedule of production and distribution such that the sum of the maximum delivery time and total delivery cost is optimized. We study two classes of processing machines in the production part. In the first class, the serial-batch machine, the processing time of a batch is the sum of the processing times of its jobs. In the second class, the parallel-batch machine, the processing time of a batch is the maximum processing time of the jobs contained in the batch. The machine has a fixed capacity, and the jobs are processed in batches under the condition that the total size of the jobs in a batch cannot exceed the machine capacity. Two patterns of job's processing, i.e., resumable and non-resumable, are considered if it is interrupted by the unavailability interval on the machine. In the distribution part, there are sufficient vehicles with a fixed capacity to deliver the completed jobs. The total size of the completed jobs in one delivery cannot exceed the vehicle capacity. We show that these four problems are NP-hard in the strong sense in which the jobs have the same processing times and arbitrary sizes, and we propose an approximation algorithm for solving these four problems. Moreover, we show that the performance ratio of the algorithm is 2 for the serial-batch machine setting, and the error bound is 71/99 for the parallel-batch machine setting. We also evaluate the performance of the approximation algorithm by the computational results.

## 1. Introduction

Joint consideration of production and distribution is so beneficial in making higher level decisions for the manufacturers in the realistic supply chain environment that many researchers kept on studying on various models of such integrated scheduling problems. Potts [1] was probably the first researcher who considered scheduling with job delivery. Hall and Potts [2] studied integrated scheduling that involves a supplier, a manufacturer, and a customer. Up to now, there are an increasing number of literature and a variety of new models on scheduling problems with job delivery. For one machine, Chen and Vairaktarakis [3] studied the problem to minimize the weighted sum of maximum delivery time and total transportation cost with a relative preference, in which sufficient vehicles without capacity limit were used to deliver completed jobs to customers. They proposed two optimal algorithms for one customer and multiple customers. Chen [4] and Wang et al. [5] have surveyed the integrated scheduling problems of production and distribution operation.

In most models of scheduling with delivery, one machine processes one job at a time. However, a new kind of integrated scheduling problems concerned with batch processing has already been investigated by many researchers. Batch scheduling is motivated by many industrial manufacturing processes. For example, in the burn-in stage of semiconductor manufacturing, burn-in ovens can handle up to multiple jobs simultaneously. Generally, batch-processing machine has two versions: serial-batch and parallel-batch, according to [6]. When processed on the serial-batch machine, jobs may be batched and one job in a batch is processed at a time so that the processing time of a batch is the sum of the processing times of its jobs. Learning effect of jobs is always accompanied with the serial-batching scheduling problems. Lee et al. [7] and Pei et al. [8–10]

investigated the serial-batching scheduling problems with deteriorating jobs or learning effects to minimize the makespan. There is no setup time in [7], but there are independent setup times in [8–10]. Pei et al. [11] considered the coordinated scheduling problem of production and transportation with deteriorating jobs. The batches including completed jobs would be delivered to a customer by a single vehicle with only one batch in one delivery. They analysed some useful properties and presented a heuristic for the general case and two optimal algorithms for two special cases. Lu et al. [12] studied the integrated production and delivery scheduling on a serial-batch machine to minimize the makespan and considered four different problems from the situations whether split is allowed in the production or delivery of the jobs.

When processed on the parallel-batch machine, several jobs can be processed as a batch simultaneously on a machine at one time such that the processing time of a batch is the maximum processing time of the jobs contained in the batch. The bounded parallel-batch-processing machine setting is introduced by Lee et al. [13], which is always encountered in burn-in operations in the semiconductor industry and heat treatment operations in the metalworking industries. Uzsoy [14] showed the problem is NP-hard to minimize makespan on a single bounded parallel-batch machine and provided some approximation algorithms. And Brucker et al. [15] discussed two variants: the unbounded model and the bounded model. Lee and Lee [16] developed a heuristic by iterative decomposition of a mixed integer programming model. Li et al. [17], Gong et al. [18], Lu and Yuan [19], and Cheng et al. [20] considered some integrated scheduling of production and distribution on parallel-batch-processing machines. In [20], the authors focused on finding a schedule of the jobs and a delivery plan so that the maximum delivery time of jobs is minimized. They showed that the problem can be solved optimally in $O(n\log n)$ time if the jobs have identical sizes and proposed two approximation algorithms with an asymptotical performance ratio of 11/9 if the jobs have identical processing times and 2 if the jobs have arbitrary processing times and arbitrary sizes.

Up to now, most works have studied the machine setting which is always available. But, the machines can become unavailable during the production stage due to the occurrence of breakdowns or the necessity for maintenance and repair, which is called as the scheduling under the constraint of machine unavailability. When job processing is interrupted by machine unavailability, the interrupted job may be resumable or nonresumable once the machine is available again. In the resumable case, the interrupted job can be processed continuously, but in the nonresumable case, the interrupted job needs to be processed anew. For the integrated scheduling problem with a capacitated vehicle to minimize the maximum delivery time, Wang and Cheng [21] showed that it is NP-hard and proposed a 3/2-approximation algorithm and provided the instance with the worst-case ratio 3/2. More details on this research stream can be found in the work of Wang et al. [5] and Ma et al. [22]. There are a few literatures concerned with the integrated scheduling problem on the batch-processing machine with the unavailable constraints. Pei et al. [23] considered a single-machine serial-batching scheduling problem with a machine availability constraint, position-dependent processing time, and time-dependent setup time. Fan et al. [24] studied the integrated scheduling problem to minimize the maximum delivery time to the customer on a single bounded parallel-batch machine with an unavailability interval. For the nonresumable jobs, two approximation algorithms with the worst-case ratios 3/2 and 5/3 are explored in two cases where the jobs have the same size and arbitrary processing times, and the jobs have the same processing time and arbitrary sizes, respectively.

In this paper, we consider an integrated scheduling problem for a customer on a bounded batch machine with a machine unavailability period, where the jobs have identical processing times and sufficient capacitated vehicles are used to deliver the completed jobs to the customer. The objective is to minimize the sum of the delivery time when the last completed job is delivered to the customer and the total delivery cost. There are four classes of problems corresponding to different conditions during the processing of the interrupted job after the unavailability period and the type of machine, i.e., serial-batching and parallel-batching. In the practical aluminum-making process, serial-batching is common. Cylindrical aluminum ingots are processed one after another in the same batch on an extrusion machine, which has a schedule to be checked. Also, in the burn-in stage of semiconductor manufacturing, the burn-in oven can handle up to multiple jobs simultaneously, which is the form of parallel-batching. The oven must be stopped to accept inspection as a plan. All completed jobs have to be transported to the customer as soon as possible. Therefore, how to schedule the production and the delivery effectively is worth to study in such complicated settings.

## 2. Problem Formulation

A set of $n$ jobs $\{J_1, J_2, \ldots, J_n\}$ is given, where job $J_j$ has the processing time $p_j$ and size $s_j$. Each job has an arbitrary size but an identical processing time $p_j = p$ for $j = 1, 2, \ldots, n$. In the production part, the machine has a capacity $U$, i.e., it can simultaneously process several jobs as a batch if their total size does not exceed the machine capacity. We discuss two types of batches, that is, serial-batching and parallel-batching. For the serial-batching, the processing time of a batch is the sum of the processing times of its jobs. For the parallel-batching, the processing time of a batch is the maximum processing time of the jobs contained in the batch. Meanwhile, the machine has an unavailability period $[\eta_1, \eta_2]$ because of maintenance and breakdown. Let $\eta$ be the length of the unavailability period, i.e., $\eta = \eta_2 - \eta_1$. If there is, at least, one job in a batch that is interrupted by the unavailable period $[\eta_1, \eta_2]$, the interrupted job may be processed continuously just after $\eta_2$, in which case we call it resumable and denoted by $r-a$, or needs to be processed anew after $\eta_2$, defined as nonresumable and denoted by $nr-a$. In the latter case, the processing time of the batch containing this job is not affected for the parallel-batching

mao the sum of $\eta$ and the total processing time of the jobs and the idle time of the machine for the serial-batching machine. In the delivery part, there are sufficient vehicles to deliver the completed jobs to the customer and each of them has the same capacity $V$. Similar to [20], we assume that $V = xU$, where $x \geq 2$ and $x$ is a positive integer. The transport price of one trip between the machine and the customer is denoted by $c$ and the total delivery cost is denoted by TC. Let $D_j$ be the delivery time of job $J_j$, i.e., the arrival time of the batch containing job $J_j$ to the customer. Let $D_{\max}$ be the maximum delivery time of all the jobs. Our goal is to minimize the sum of maximum delivery time and total delivery cost. Using the five-field notation proposed by Chen [4] to denote an integrated scheduling problem, we denote the four classes of the problem in this paper as follows:

$(P1)$: $1, h_1|r-a, s-\text{batch}, s_j, p_j = p|V(\infty, c), \text{direct}|1|D_{\max} + \text{TC}$

$(P2)$: $1, h_1|nr-a, s-\text{batch}, s_j, p_j = p|V(\infty, c), \text{direct}|1|D_{\max} + \text{TC}$

$(P3)$: $1, h_1|r-a, p-\text{batch}, s_j, p_j = p|V(\infty, c), \text{direct}|1|D_{\max} + \text{TC}$

$(P4)$: $1, h_1|nr-a, p-\text{batch}, s_j, p_j = p|V(\infty, c), \text{direct}|1|D_{\max} + \text{TC}$

Here, $h_1$ denotes one unavailability interval and $V(\infty, c)$ presents the situation of the vehicle. Also, because there is one customer, the vehicle should transport jobs directly from the manufacturer to the customer. In the problems $(P1)$ and $(P2)$, *s-batch* means serial-batching. On the contrary, in the problems $(P3)$ and $(P4)$, *p-batch* means parallel-batching.

We organize the rest of the paper as follows. In Section 3, we show that all problems are NP-hard in the strong sense and propose some elementary properties. In Section 4, we present an approximation algorithm for these four problems and prove the different error bounds for these four problems. In the last section, we conclude our results and discuss the direction of the future research.

## 3. Properties of Four Problems

In this section, we analyse the computational complexity of the problems $(P1)$–$(P4)$.

**Theorem 1.** *Problems* $(P1)$–$(P4)$ *are all strongly NP-hard.*

*Proof.* Consider the special case $(\text{P}')$ of problems $(P1)$ and $(P2)$, in which $p = 0$, $\eta_1 = \eta_2$, and $c = 0$, i.e., the processing time of each job is zero, and there is no unavailability interval on the machine and no delivery in problem $(\text{P}')$. Hence, the problem $(\text{P}')$ is equivalent to minimize the number of batches, i.e., the bin-packing problem, which is a well-known strongly NP-hard problem. Therefore, problems $(P1)$ and $(P2)$ are strongly NP-hard.

Similarly, we construct the special case $(\text{P}'')$ of problems $(P3)$ and $(P4)$, in which $\eta_1 = \eta_2$ and $c = 0$. Since each job has a processing time of $p$, every batch has a processing time of $p$. Hence, the problem $(\text{P}'')$ is also equivalent to the bin-packing problem. Therefore, problems $(P3)$ and $(P4)$ are both strongly NP-hard.

Associating with the bin-packing problem, we study some properties of the optimal solutions for these four problems. □

**Lemma 1.** *There exists an optimal schedule* $\gamma_i^*$ *of the problem* $(Pi)$ *for* $i = 1, 2, 3, 4$, *satisfying the following properties:*

*(1) Let* $X^*$ *to be the number of batches in the optimal schedule* $\gamma_i^*$*; then,* $X^* = \lceil n/U \rceil$

*(2) The batches are processed consecutively before and after the unavailability period*

*(3) The batch that becomes available earlier is delivered earlier*

*(4) The first delivery includes* $\beta^*$ *batches, and each of the last* $\alpha^*$ *deliveries* $x$ *batches, where* $\alpha^*$ *and* $\beta^*$ *are two positive integers satisfying* $X^* = \alpha^* x + \beta^*$ *and* $0 < \beta^* \leq x$

*This lemma can be proved similar with the proof in [21].*

## 4. Algorithm and Analyses

To minimize the objective function value, we must assign many enough jobs in every batch to reduce the number of batches and deliveries. Therefore, associating with the algorithm of the bin-packing problem, we present the following algorithm to solve four scheduling problems.

### 4.1. Algorithm H

Step 1: reindex all jobs as $\{J_1, J_2, \ldots, J_n\}$ so that $s_1 \geq s_2 \geq \ldots \geq s_n$.

Step 2: use the First Fit Decreasing (FFD) rule to assign the jobs into batches, in which jobs are to be processed together on the machine. Build the first empty batch $B_1$ and put the job into it one by one in the sequence in Step 1 if the current total sizes of the batch $B_1$ are no more than $U$. When all the jobs have been checked and there are still jobs need to be assigned, build the second batch $B_2$ and assign the remaining jobs as the sequence in Step 1. Repeat the assignment until there is no job left. Let X be the number of batches produced in Step 2.

Step 3: assign the batches in a nondecreasing order of their processing times to process continuously from time 0 except the unavailability interval.

Step 4: Let. $X = \alpha x + \beta$ and $0 < \beta \leq x$ deliver the first completed $\beta$ batches in $D_1$. Also, deliver $x$ batches immediately in the following every delivery. When the last $x$ batches are delivered to the customer, the algorithm finishes.

Recalling the FFD rule for the bin-packing problem, we can easily obtain Lemma 2.

**Lemma 2** (see [25]). $X \le 11/9X^* + 6/9$.

Based on Lemma 2, we can explore the corresponding values of $X$ and $X^*$ if $X^* \le 9$ (see Table 1).

In order to list the corresponding relationship of values between $X$ and $X^*$ if $X^* \ge 10$, we use two positive integers $a \ge 1$ and $1 \le b \le 9$ such that $X^* = 9a + b$ similar to that in [26]. Hence, we can obtain Table 2.

Furthermore, we analyse the relationship of the objective function values produced by algorithm H and the optimal schedule in the problems $(P1)$–$(P4)$, respectively. For convenience, we use $\gamma_i$ to denote the schedule obtained by algorithm H for the problem $(Pi)$.

Table 1: The corresponding values of $X$ and $X^*$ if $X^* \le 9$.

| Variables | Values | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $X^*$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $X$ | 1 | 2, 3 | 3, 4 | 4, 5 | 5, 6 | 6, 7, 8 | 7, 8, 9 | 8, 9, 10 | 9, 10, 11 |

Table 2: The maximal values of $X$ and the upper bound of $X/X^*$ if $X^* \ge 10$.

| Variables | Values | | | | |
|---|---|---|---|---|---|
| $X^*$ | $9a+1$ | $9a+2$ | $9a+3$ | $9a+4$ | $9a+5$ |
| The maximal value of $X$ | $11a+1$ | $11a+3$ | $11a+4$ | $11a+5$ | $11a+6$ |
| Upper bound of $X/X^*$ | 11/9 | 14/11 | 5/4 | 16/13 | 11/9 |
| $X^*$ | $9a+6$ | $9a+7$ | $9a$+8 | $9a+9$ | |
| The maximal value of $X$ | $11a+8$ | $11a+9$ | $11a$+10 | $11a+11$ | |
| Upper bound of $X/X^*$ | 19/15 | 5/4 | 21/17 | 11/9 | |

**Lemma 3**

(1) *$d(P2) - d^*(P2) \le np$, where $d(P2)$ and $d^*(P2)$ are the idle times on the machine before the unavailability interval in schedules $\gamma_2$, $\gamma_2^*$ for the problem $(P2)$, respectively*

(2) *$d(P4) \le d^*(P4)$, where $d(P4)$ and $d^*(P4)$ are the idle times on the machine before the unavailability interval in schedules $\gamma_4$, $\gamma_4^*$ for the problem $(P4)$, respectively*

*Proof* (1) According to Step 3 of Algorithm H, the batches are processed in a nondecreasing order of their processing times. Let $n_b$, $n_b^*$ be the number of jobs completed before the beginning time of $[\eta_1, \eta_2]$ in schedules $\gamma_2$, $\gamma_2^*$, respectively. Because the length of the available time before $\eta_1$ on the machine is the fixed, we have $n_b p + d(P2) = n_b^* p + d^*(P2)$. Hence, $d(P2) - d^*(P2) \le (n_b^* - n_b)p \le n_b * p \le np$.

(2) Because the processing time of each batch is equal to $p$ no matter how many jobs are in it and the starting time of processing is zero by the algorithm H, it is obvious that the idle time in $\gamma_4$ must be no more than that in $\gamma_4^*$, i.e., $d(P4) \le d^*(P4)$.

For the problem $(Pi)$, we use $D_{\max}(Pi)$, $D*_{\max}(Pi)$, $\mathrm{TC}(Pi)$, and $\mathrm{TC}^*(Pi)$ to denote the maximum delivery time, the total delivery cost in $\gamma_i$, and $\gamma_i^*$ $(i = 1, 2, 3, 4)$, respectively. We can easily conclude every value of $D_{\max}(Pi)$, $D_{\max}^*(Pi)$, $\mathrm{TC}(Pi)$, and $\mathrm{TC}^*(Pi)$ for $i = 1, 2, 3, 4$. □

**Lemma 4** *(1) In the problem $(P1)$, $D_{\max}(P1) = \eta + np$, $TC(P1) = (\alpha + 1)c$ and $D*_{\max}(P1) = \eta + np$, $TC^*(P1) = (\alpha^* + 1)c$*

*(2) In the problem $(P2)$, $D_{\max}(P2) = \eta + np + d(P2)$, $TC(P2) = (\alpha + 1)c$ and $D*_{\max}(P2) = \eta + np + d^*(P2)$, $TC^*(P2) = (\alpha^* + 1)c$*

*(3) In the problem $(P3)$, $D_{\max}(P3) = \eta + Xp$, $TC(P3) = (\alpha + 1)c$ and $D_{\max}^*(P3) = \eta + X^*p$, $TC^*(P3) = (\alpha^* + 1)c$*

*(4) In the problem $(P4)$, $D_{\max}(P4) = \eta + Xp + d(P4)$, $TC(P4) = (\alpha + 1)c$ and $D*_{\max}(P4) = \eta + X^*p + d^*(P4)$, $TC^*(P4) = (\alpha^* + 1)c$*

*Proof.* Because the processes of batches are resumable in the problem $(P1)$, the difference between the result of the algorithm H and the optimal solution is only the number of deliveries. In the problem $(P2)$, the idle time of the machine is also different between the result of the algorithm H and the optimal solution. Moreover, in the problem $(P3)$, the total processing time $X_p$ is needed in the solution of the algorithm H, but $X_p^*$ is needed in the optimal solution. Also, the difference of the idle time of the machine is obvious in the objective function value of the algorithm H and the optimal schedule in the problem $(P4)$.

It is notable that the completion time of the last batch is larger than the ending time of the unavailability interval in the optimal schedule $\gamma_i*$ $(i = 1, 2, 3, 4)$. Otherwise, the performance ratio will approach infinity.

From the objectives of every problem, we find that the relationship between $\alpha$ and $\alpha^*$ is critical. Hence, we explore the following lemma. □

**Lemma 5** (See [24]). $\alpha^* \le \alpha < 11/9\alpha^* + 14/9$.

*Proof.* $\alpha^* \le \alpha$ is obvious. Due to $x \ge 2$, Lemma 2, and the explanation of $X^* = \alpha^* x + \beta^*$, $X = \alpha x + \beta$, we can deduce

$$\alpha \le \frac{X}{x} \le \frac{1}{x}\left(\frac{11}{9}X^* + \frac{6}{9}\right) = \frac{1}{x}\left(\frac{11}{9}\left(\alpha^* x + \beta^*\right) + \frac{6}{9}\right) \le \frac{11}{9}\alpha^* + \frac{14}{9}. \tag{1}$$

Moreover, if $\alpha = 11/9\alpha^* + 14/9$, we have

$$\frac{11}{9}X^* + \frac{6}{9} = \frac{11}{9}(\alpha^* x + \beta^*) + \frac{6}{9} \le \frac{11}{9}\alpha^* x + \left(\frac{11}{9}x + \frac{6}{9}\right) \le \frac{11}{9}\alpha^* x + \frac{14}{9}x = \alpha x < X, \tag{2}$$

which contradicts Lemma 2.

Consequently, we can easily obtain the corresponding values of $\alpha$ and $\alpha^*$ if $\alpha^* \le 4$ (see Table 3).

To simplify the expression the objective function value of each problem, we use $Z(Pi)$ and $Z^*(Pi)$ to denote the objective function values obtained by algorithm H and the optimal schedule, respectively. □

**Theorem 2.** *Algorithm H is a 2-approximation algorithm for solving the problem (P1) and the problem (P2), and the worst-case ratio is tight.*

*Proof.* According to Lemma 4, it is obvious that

$$\frac{Z(P1) - Z^*(P1)}{Z^*(P1)} = \frac{(\alpha - \alpha^*)c}{\eta + np + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)c}{\eta + np + \alpha^* c} \le \frac{\alpha - \alpha^*}{\alpha^*}. \tag{3}$$

From Table 3, we have that the algorithm reaches the worst-case ratio when $\alpha = 2$ and $\alpha^* = 1$.

For the problem $(P2)$, because of (1) in Lemma 3, we have

$$\frac{Z(P2) - Z^*(P2)}{Z^*(P2)} = \frac{(\alpha - \alpha^*)c + (\mathrm{d}(P2) - \mathrm{d}^*(P2))}{\eta + np + \mathrm{d}^*(P2) + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)c + np}{np + \alpha^* c} \le 1. \tag{4}$$

Table 3: The corresponding values of $\alpha$ and $\alpha^*$ if $\alpha^* \le 4$.

| Variables | Values | | | |
|---|---|---|---|---|
| $\alpha^*$ | 1 | 2 | 3 | 4 |
| $\alpha$ | 1, 2 | 2, 3 | 3, 4, 5 | 4, 5, 6 |

Similarly, when $\alpha = 2$ and $\alpha^* = 1$, the algorithm reaches the worst-case ratio.

Consider the following instance: $n = 6$, $p = 2$, $U = 7$, $x = 2$, $[\eta_1, \eta_2] = [2, 2 + \varepsilon]$, and $s_1 = s_2 = 3$, $s_3 = s_4 = s_5 = s_6 = 2$. The schedule produced by algorithm H is as follows: the first delivery including job $J_6$ as the first batch is delivered at time 2; the second delivery including two batches, which consist of jobs $J_1$ and $J_2$ and jobs $J_3$, $J_4$, and $J_5$ respectively, is delivered at time $12 + \varepsilon$. Hence, $Z(P1) = Z(P2) = 12 + \varepsilon + 2c$. However, in the optimal schedule, there is only one delivery, including two batches which consist of jobs $J_1$, $J_3$, and $J_4$ and jobs $J_2$, $J_5$, and $J_6$ respectively. This delivery occurs at time $12 + \varepsilon$, and the optimal objective value is $Z^*(P1) = Z^*(P2) = 12 + \varepsilon + c$. If $c$ is sufficiently large, we have $Z(P1)/Z^*(P1) = Z(P2)/Z^*(P2) \longrightarrow 2$.

For the problem $(P3)$ and the problem $(P4)$, we obtain the error bound of algorithm H in the following theorem. □

**Theorem 3.** *Algorithm H has an error bound 71/99 when solving the problem (P3) and the problem (P4).*

*Proof.* Because of (2) of Lemma 3, we have

$$\frac{Z(P3) - Z^*(P3)}{Z^*(P3)} = \frac{(X - X^*)p + (\alpha - \alpha^*)c}{\eta + X^* p + (\alpha^* + 1)c} \le \frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c},$$
$$\frac{Z(P4) - Z^*(P4)}{Z^*(P4)} = \frac{(X - X^*)p + d(P4) - d^*(P4) + (\alpha - \alpha^*)c}{\eta + X^* p + (\alpha^* + 1)c} \le \frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c}, \tag{5}$$

We only need analyse the upper bound of $(X - X^*)p + (\alpha - \alpha^*)c / X^* p + (\alpha^* + 1)c$. Moreover, we prove the result in two cases according to $\alpha^* \le 4$ and $\alpha^* \ge 5$, respectively.

Case 1: $\alpha^* \le 4$. We discuss this case in two subcases.

Subcase 1.1: when we consider the situation that $\beta^* \le \beta$, we can obtain the following inequality for $\alpha = \alpha^* + k$ $(k = 0, 1, 2)$ according to Table 3:

$$(\alpha^* + k)x + \beta^* \le (\alpha^* + k)x + \beta = \alpha x + \beta = X \le \frac{11}{9}X^* + \frac{6}{9} = \frac{11}{9}(\alpha^* x + \beta^*) + \frac{6}{9}. \tag{6}$$

$$\left(k - \frac{2}{9}\alpha^* - \frac{2}{9}\right)x \le \frac{6}{9}. \tag{7}$$

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} = \frac{(\alpha - \alpha^*)xp + (\beta - \beta^*)p + (\alpha - \alpha^*)c}{(\alpha^* x + \beta^*)p + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)xp + xp + (\alpha - \alpha^*)c}{\alpha^* xp + (\alpha^* + 1)c}. \tag{8}$$

For $k = 0$, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le X - X^*/X^* \le 3/11 < 71/99$ according to Tables 1 and 2.

For $k = 1$, $\alpha^* = 4$, we have $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le xp + xp + c/\alpha^* xp + (\alpha^* + 1)c \le 2xp + c/4xp + 5c < 1/2 < 71/99$.

Subcase 1.2: when we consider the situation that $\beta^* > \beta$, we have the following inequality from Table 3 except for $\alpha^* = 1$ and $\alpha = 2$:

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} = \frac{(\alpha - \alpha^*)xp + (\beta - \beta^*)p + (\alpha - \alpha^*)c}{(\alpha^* x + \beta^*)p + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)xp + (\alpha - \alpha^*)c}{\alpha^* xp + (\alpha^* + 1)c} \le \frac{\alpha - \alpha^*}{\alpha^*} \le \frac{2}{3} < \frac{71}{99}. \tag{9}$$

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} \le \frac{X - X^*}{X^*} + \frac{\alpha - \alpha^*}{\alpha^* + 1}. \tag{10}$$

Moreover, associating with $\beta^* \le x$, we have

When k=0, k=1 with $\alpha^* = 4$, it is easy to deduce that inequality (7) holds when, $k = 1$ with $\alpha^* = 4$.

For $\alpha^* = 1$ and $\alpha = 2$, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha * + 1)c = (X - X^*)p + c/X^* p + 2c$. Moreover, we have $X/X^* \le 3/2$ according to Tables 1 and 2. Hence, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le 1/2 < 71/99$.

Case 2: $\alpha^* \ge 5$. We can easily obtain the inequality (10).

Because $X/X^* \le 14/11$ according to Table 2 and $\alpha - \alpha^*/\alpha^* + 1 < 2/9\alpha^* + 14/9/\alpha^* + 1 = 2/9 + 4/3 \cdot 1/\alpha^* + 1 \le 2/9 + 4/3 \cdot 1/6 = 4/9$ according to Lemma 5, we obtain

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} \le \frac{3}{11} + \frac{4}{9} = \frac{71}{99}, \tag{11}$$

and $Z(P3) - Z^*(P3)/Z^*(P3) \le 71/99$ and $Z(P4) - Z^*(P4)/Z^*(P4) \le 71/99$. □

## 5. Computational Results

Since the four problems are all strongly NP-hard and the bin-packing problem is contained in these problems, solving them will be costly and time consuming. But, in order to evaluate the effectiveness of the algorithm H, we consider four numerical simulations for the small size problems. The optimal schedule is obtained by processing the batches according to the branch-and-bound algorithm for the bin-packing problem in [27] and delivering as (4) in Lemma 1.

The approximation algorithm and the branch-and-bound algorithm are coded in VC++, and the computational experiments are run on a personal computer. We utilize three different job numbers, $n = 30$ and 50. Let the processing time of each job be 1, and the sizes of all jobs are generated randomly from a discrete uniform distribution [1, 10]. Let the capacity of the machine $U = 3$, 7 and the capacity of the vehicle $V = xU = 2U$, $10U$. We set two levels of the length of the unavailability period $\eta = 0.2$, 1.2 and two levels of the beginning time of the unavailability interval $\eta_1 = 3$, 10 in the experiments. The cost of each delivery batch is set as $c = 0.5$, 2. For each combination of $n$, $\eta$, $\eta_1$, and $c$, 20 random instances are used, and the computational results are summarized in from Tables 4 to 7, which reports the average number of seconds of CPU time of the optimal solutions and the solutions of algorithm H for each combination of parameters. The percentage error of the solution produced by the algorithm H is calculated as $\rho = Z(Pi) - Z^*(Pi)/Z^*(Pi)$.

It is observed from Tables 4–7 that the average CPU times increase in polynomial times for the algorithm H when the number of jobs increase and increase in exponential times for the branch-and-bound algorithm. It is found that the average worst-case error bounds of the optimal solution of ($P2$) and ($P4$) are generally more than ($P1$) and ($P3$) because of the nonresumable conditions. Finally, the most important thing is that these four tables show that the approximation algorithm performs efficiently and the actual error bounds of four problems are lower than the theoretical error bounds.

Table 4: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P1$).

| $n$ | $U$ | $x$ | $\eta$ | $\eta_1$ | $c$ | Average CPU times of the algorithm H | Average CPU times of B&B algorithm | Average value of $\rho$ |
|---|---|---|---|---|---|---|---|---|
| 50 | 3 | 2 | 0.2 | 3 | 0.5 | 0.0603 | 15.0346 | 1.012 |
| | | | | | 2 | 0.0615 | 15.0191 | 1.013 |
| | | | 0.2 | 10 | 0.5 | 0.0618 | 15.0233 | 1.011 |
| | | | | | 2 | 0.0612 | 15.0212 | 1.011 |
| | | | 1.2 | 3 | 0.5 | 0.0609 | 15.0305 | 1.013 |
| | | | | | 2 | 0.0620 | 15.0145 | 1.012 |
| | | | 1.2 | 10 | 0.5 | 0.0605 | 15.0221 | 1.012 |
| | | | | | 2 | 0.0618 | 15.0314 | 1.013 |
| 50 | 7 | 10 | 0.2 | 3 | 0.5 | 0.0607 | 15.0324 | 1.011 |
| | | | | | 2 | 0.0608 | 15.0219 | 1.012 |
| | | | 0.2 | 10 | 0.5 | 0.0621 | 15.0309 | 1.012 |
| | | | | | 2 | 0.0624 | 15.0242 | 1.013 |
| | | | 1.2 | 3 | 0.5 | 0.0619 | 15.0287 | 1.011 |
| | | | | | 2 | 0.0618 | 15.0237 | 1.012 |
| | | | 1.2 | 10 | 0.5 | 0.0620 | 15.0228 | 1.010 |
| | | | | | 2 | 0.0621 | 15.0158 | 1.011 |

Table 5: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P2$).

| $n$ | $U$ | $x$ | $\eta$ | $\eta_1$ | $c$ | Average CPU times of the algorithm H | Average CPU times of B&B algorithm | Average value of $\rho$ |
|---|---|---|---|---|---|---|---|---|
| 30 | 3 | 2 | 0.2 | 3 | 0.5 | 0.0512 | 3.5346 | 1.130 |
| | | | | | 2 | 0.0526 | 3.6121 | 1.126 |
| | | | 0.2 | 10 | 0.5 | 0.0518 | 3.5193 | 1.213 |
| | | | | | 2 | 0.0572 | 3.4023 | 1.204 |
| | | | 1.2 | 3 | 0.5 | 0.0501 | 3.5012 | 1.127 |
| | | | | | 2 | 0.0513 | 3.3986 | 1.123 |
| | | | 1.2 | 10 | 0.5 | 0.0542 | 3.4591 | 1.113 |
| | | | | | 2 | 0.0537 | 3.5029 | 1.138 |
| 30 | 7 | 10 | 0.2 | 3 | 0.5 | 0.0527 | 3.3092 | 1.214 |
| | | | | | 2 | 0.0528 | 3.4290 | 1.209 |
| | | | 0.2 | 10 | 0.5 | 0.0518 | 3.4232 | 1.120 |
| | | | | | 2 | 0.0519 | 3.5014 | 1.118 |
| | | | 1.2 | 3 | 0.5 | 0.0529 | 3.4879 | 1.104 |
| | | | | | 2 | 0.0608 | 3.3871 | 1.106 |
| | | | 1.2 | 10 | 0.5 | 0.0538 | 3.4381 | 1.169 |
| | | | | | 2 | 0.0521 | 3.3080 | 1.184 |

Table 6: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P3$).

| $n$ | $U$ | $x$ | $\eta$ | $\eta_1$ | $c$ | Average CPU times of the algorithm H | Average CPU times of B&B algorithm | Average value of $\rho$ |
|---|---|---|---|---|---|---|---|---|
| 50 | 3 | 10 | 0.2 | 3 | 0.5 | 0.0601 | 15.0412 | 1.103 |
| | | | | | 2 | 0.0612 | 15.0381 | 1.104 |
| | | | 0.2 | 10 | 0.5 | 0.0623 | 15.0315 | 1.113 |
| | | | | | 2 | 0.0625 | 15.0406 | 1.118 |
| | | | 1.2 | 3 | 0.5 | 0.0614 | 15.0337 | 1.217 |
| | | | | | 2 | 0.0623 | 15.0341 | 1.212 |
| | | | 1.2 | 10 | 0.5 | 0.0625 | 15.0502 | 1.204 |
| | | | | | 2 | 0.0631 | 15.0513 | 1.210 |
| 50 | 7 | 2 | 0.2 | 3 | 0.5 | 0.0605 | 15.0372 | 1.132 |
| | | | | | 2 | 0.0611 | 15.0381 | 1.179 |
| | | | 0.2 | 10 | 0.5 | 0.0608 | 15.0401 | 1.124 |
| | | | | | 2 | 0.0621 | 15.0427 | 1.119 |
| | | | 1.2 | 3 | 0.5 | 0.0620 | 15.0381 | 1.221 |
| | | | | | 2 | 0.0619 | 15.0348 | 1.220 |
| | | | 1.2 | 10 | 0.5 | 0.0618 | 15.0329 | 1.217 |
| | | | | | 2 | 0.0623 | 15.0414 | 1.215 |

Table 7: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P4$).

| $n$ | $U$ | $x$ | $\eta$ | $\eta_1$ | $c$ | Average CPU times of the algorithm H | Average CPU times of B&B algorithm | Average value of $\rho$ |
|---|---|---|---|---|---|---|---|---|
| 30 | 3 | 10 | 0.2 | 3 | 0.5 | 0.0525 | 3.5281 | 1.219 |
| | | | | | 2 | 0.0527 | 3.5356 | 1.193 |
| | | | 0.2 | 10 | 0.5 | 0.0541 | 3.5027 | 1.201 |
| | | | | | 2 | 0.0547 | 3.5301 | 1.208 |
| | | | 1.2 | 3 | 0.5 | 0.0535 | 3.5531 | 1.120 |
| | | | | | 2 | 0.0539 | 3.5890 | 1.119 |
| | | | 1.2 | 10 | 0.5 | 0.0541 | 3.6211 | 1.301 |
| | | | | | 2 | 0.0545 | 3.6236 | 1.271 |
| 30 | 7 | 3 | 0.2 | 3 | 0.5 | 0.0529 | 3.7210 | 1.271 |
| | | | | | 2 | 0.0532 | 3.8104 | 1.229 |
| | | | 0.2 | 10 | 0.5 | 0.0537 | 3.7147 | 1.201 |
| | | | | | 2 | 0.0539 | 3.7818 | 1.203 |
| | | | 1.2 | 3 | 0.5 | 0.0531 | 3.8201 | 1.308 |
| | | | | | 2 | 0.0610 | 3.9032 | 1.320 |
| | | | 1.2 | 10 | 0.5 | 0.0549 | 3.7138 | 1.301 |
| | | | | | 2 | 0.0556 | 3.9102 | 1.284 |

## 6. Conclusions

We consider four integrated scheduling problems to minimize the sum of the maximum delivery time and the total delivery cost, in which a set of jobs are first processed on a single batch machine with an unavailability interval and, then, delivered to the customer directly, and the jobs have the same processing time and arbitrary sizes. We show that these four problems are all strongly NP-hard and propose an approximation algorithm for them. Moreover, we obtain that the worst-case error bound of the algorithm for the serial-batch machine setting and the parallel-batch machine setting, respectively. We also provide computational results to evaluate the performance of the heuristics.

The direction of future research may focus on other settings of jobs and vehicles, such as jobs with arbitrary processing times and arbitrary sizes or one vehicle with limited capacity.

## Data Availability

Radom data used to support this study can be obtained from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (116013 16), the key discipline "Applied Mathematics" of Shanghai Polytechnic University (XXKP Y1604), the subproject of "first class postgraduate education leading plan," Research Center of Resource Recycling Science and Engineering, and Gaoyuan Discipline of Shanghai—Environmental Science and Engineering (Resource Recycling Science and Engineering) of Shanghai Polytechnic University.

## References

[1] C. N. Potts, "Technical note-analysis of a heuristic for one machine sequencing with release dates and delivery times," *Operations Research*, vol. 28, no. 6, pp. 1436–1441, 1980.

[2] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.

[3] Z.-L. Chen and G. L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.

[4] Z.-L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.

[5] D. Y. Wang, O. Grunder, and A. E. Moudni, "Integrated scheduling of production and distribution operations: a review," *International Journal of Industrial and Systems Engineering*, vol. 19, no. 1, pp. 94–122, 2015.

[6] H. S. Mirsanei, B. Karimi, and F. Jolai, "Flow shop scheduling with two batch processing machines and nonidentical job sizes," *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 5-6, pp. 553–572, 2009.

[7] W. C. Lee, C. C. Wu, and P. H. Hsu, "A single-machine learning effect scheduling problem with release times," *Omega*, vol. 38, no. 1, pp. 3–11, 2010.

[8] J. Pei, X. Liu, P. M. Pardalos, W. Fan, and S. Yang, "Single machine serial-batching scheduling with independent setup time and deteriorating job processing times," *Optimization Letters*, vol. 9, no. 1, pp. 91–104, 2015.

[9] J. Pei, X. Liu, P. M. Pardalos, A. Migdalas, and S. Yang, "Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine," *Journal of Global Optimization*, vol. 67, no. 1, pp. 251–262, 2017.

[10] J. Pei, X. Liu, P. M. Pardalos et al., "Scheduling deteriorating jobs on a single serial batching machine with multiple job types and sequence-dependent setup times," *Annals of Operations Research*, vol. 249, no. 1, pp. 175–195, 2017.

[11] J. Pei, P. M. Pardalos, X. Liu, W. Fan, and S. Yang, "Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan," *European Journal of Operational Research*, vol. 244, no. 1, pp. 13–25, 2015.

[12] L. Lu, L. Zhang, and L. Wan, "Integrated production and delivery scheduling on a serial batch machine to minimize the makespan," *Theoretical Computer Science*, vol. 572, pp. 50–57, 2015.
[13] C.-Y. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Operations Research*, vol. 40, no. 4, pp. 764–775, 1992.
[14] R. Uzsoy, "Scheduling a single batch processing machine with non-identical job sizes," *International Journal of Production Research*, vol. 32, no. 7, pp. 1615–1635, 1994.
[15] P. Brucker, A. Gladky, H. Hoogeveen et al., "Scheduling a batching machine," *Journal of Scheduling*, vol. 1, no. 1, pp. 31–54, 1998.
[16] Y. H. Lee and Y. H. Lee, "Minimising makespan heuristics for scheduling a single batch machine processing machine with non-identical job sizes," *International Journal of Production Research*, vol. 51, no. 12, pp. 3488–3500, 2013.
[17] S. Li, J. Yuan, and B. Fan, "Unbounded parallel-batch scheduling with family jobs and delivery coordination," *Information Processing Letters*, vol. 111, no. 12, pp. 575–582, 2011.
[18] H. Gong, D. Chen, and K. Xu, "Parallel-batch scheduling and transportation coordination with waiting time constraint," *The Scientific World Journal*, vol. 2014, Article ID 356364, 8 pages, 2014.
[19] L. Lu and J. Yuan, "Unbounded parallel batch scheduling with job delivery to minimize makespan," *Operations Research Letters*, vol. 36, no. 4, pp. 477–480, 2008.
[20] B.-Y. Cheng, J. Y.-T. Leung, K. Li, and S.-L. Yang, "Single batch machine scheduling with deliveries," *Naval Research Logistics (NRL)*, vol. 62, no. 6, pp. 470–482, 2015.
[21] X. Wang and T. C. E. Cheng, "Machine scheduling with an availability constraint and job delivery coordination," *Naval Research Logistics*, vol. 54, no. 1, pp. 11–20, 2007.
[22] Y. Ma, C. Chu, and C. Zuo, "A survey of scheduling with deterministic machine availability constraints," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 199–211, 2010.
[23] J. Pei, X. Liu, P. M. Pardalos, K. Li, W. Fan, and A. Migdalas, "Single-machine serial-batching scheduling with a machine availability constraint, position-dependent processing time, and time-dependent set-up time," *Optimization Letters*, vol. 11, no. 7, pp. 1257–1271, 2017.
[24] J. Fan, C. T. Ng, T. C. E. Cheng et al., "Single bounded parallel-batch machine scheduling with an unavailability constraint and job delivery," in *Algorithmic Aspects in Information and Management*Springer, Berlin, Germany, 2020.
[25] J. Pei, B. Cheng, X. Liu, P. M. Pardalos, and M. Kong, "Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time," *Annals of Operations Research*, vol. 272, no. 1, pp. 217–241, 2019.
[26] G. Dosa, Z. Tan, Z. Tuza, Y. Yan, and C. S. Lányi, "Improved bounds for batch scheduling with nonidentical job sizes," *Naval Research Logistics (NRL)*, vol. 61, no. 5, pp. 351–358, 2014.
[27] J. M. Valério, "Exact solution of bin-packing problems using column generation and branch-and-bound," *Annals of Operations Research*, vol. 86, pp. 629–659, 1999.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 8838601, 9 pages
https://doi.org/10.1155/2020/8838601

*Research Article*

# A Scalable Mechanism Based on Blockchain for Information Processing in Energy Trading

**Xunyan Jiang** [1,2] **and Lei Wu** [2]

[1]*School of Economics & Management, Hunan Provincial Engineering Research Center of Electric Transportation and Smart Distributed Network, Changsha University of Science & Technology, Changsha 410114, China*
[2]*College of Mathematics and Computer, Xinyu University, Xinyu 338004, China*

Correspondence should be addressed to Lei Wu; jxxywulei@126.com

Received 25 September 2020; Revised 25 October 2020; Accepted 29 October 2020; Published 16 November 2020

Academic Editor: Chin-Chia Wu



Energy Internet (EI) can provide a fair, transparent, and safe environment for the energy market through the rational use of the Internet and blockchain technology. However, the existing EI has complicated management in energy, capital, and information flow. Furthermore, blockchain technology (BT) is a new technique with a scalability problem. To overcome the problem, we proposed a scalable mechanism based on blockchain for information processing in energy trading. The proposed mechanism adopts three-way communication (TWC) between energy utilities and consumers to effectively manage the energy, capital, and information flow. In the running of TWC, three modules of BT-based Internet of Things (IoT) system, BT-based virtual currency system, and BT-based EI information platform are constructed. In addition, we proposed that these modules are managed by carrying out weak centralized management to make the EI system scalable. In the operation of this mechanism, the functions of the three modules are independent and parallel. The proposed mechanism can realize the efficient transmission of energy and information and help to solve the supervision problem of energy trading. Compared with the existing information process mechanism, it is superior in promoting the dynamic balance of resources in a wide range and meeting the fast, efficient, and safe energy trade needs between the supply and demand sides.

## 1. Introduction

According to the BP Statistical Review of World Energy 2015 [1], since 2010, China has been the world's largest energy consumer. For a long time, large centralized power plants use nonrenewable fossil fuels to generate power, which causes serious threats to energy security, environmental degeneration, climate change, and human health. To address the environmental energy dilemma [2], the best solution is to reduce energy consumption and integrate the increasing renewable energy sources (RES) into the current energy system. Nevertheless, energy efficiency needs to be taken into account. Energy efficiency can be achieved whenever volatile demands and renewable energy are managed. The existing energy-efficient power grid mainly is the smart grid [3], where uncertainties and fluctuations in renewable energy management and volatile demands have always been concerned. Furthermore, existing wholesale markets lack a real-time response to the volatility and intermittent power generated by RES, and market prices do not reflect local energy shortages or oversupply [4]. In order to secure energy efficiency and support the reducing energy consumption and the integration of distributed RES into the energy system, new market approaches and information processing architectures should reflect the location of their services [5].

The locality of their services is manifested mainly by that consumers and prosumers trade self-produced energy on microgrid energy markets (MEM). The prosumers are referred to as the groups that consume and create energy. MEM allows consumers and prosumers to actively trade energy with small-

scale participants in their community in real time [1]. MEM, on which energy trading is based, facilitates that energy is provided and consumed in a sustainable, reliable, and balanced way. If the microgrid is small for every household, the future energy network will be similar to the existing Internet. This network system is called Energy Internet (EI) [6]. The multilevel microgrid is the Energy Internet subsystem to achieve a two-way flow of energy and information. The EI extends the local energy trading to that of the big scale. Every family can create and use their own energy and exchange and share energy through the EI with other users. Therefore, EI provides a viable option to integrate the distributed RES economically into the current energy system [7, 8]. Moreover, investments in local generation of EI will be incentivized with the empowerment of small-scale-energy consumers and prosumers, which help to develop the microgrid communities with the self-sustainable characteristic and develop a stochastic dynamic programming model that cooptimizes the use of energy storage for multiple applications, such as energy, capacity, and backup services [9]. However, the implementation of a smart, secure, and innovative information system in EI is the key of the successful operation for EI [10, 11].

Blockchain technology (BT) [12], as a new database technology, has the advantages of decentralization, intelligent contract, cybersecurity, and privacy protection, etc. This technology provides transparent and user-friendly applications [13] in EI and is naturally suitable for the energy market [14]. At present, BT has come to the fore in the field of EI energy market. For example, in [15], the author proposed to manage the daily energy exchange through the application of a demand response framework based on decentralized cooperatives within a community of Smart Buildings. In the framework, a fully decentralized algorithm based on BT was used to trustily communicate, bill, and autonomously monitor via smart contracts among the participants. In [16, 17], the author proposed to manage transactions in the smart grid and supply chain members by using BT. The trusted communication and immutability of the transaction between generators and consumers were executed. In [18], the potential and applications of blockchain in the EI market as distributed renewable resources and smart grid networks were discussed, in which, the author proposed distributed and decentralized solutions to the future energy market. Blockchain technology has been adopted to solve the prevailing problems of existing EI, and demand response management, complex energy transactions, and data exchange, etc. can be automated by carrying out smart contracts. The above example shows that the projects based on BT and applied in EI have been developed. Among them, some research results were tested through the process of small-scale experiments in the industry at present.

Recently, in the academic world, several researchers have given attention to the conceptualization and implementation of blockchain-based EI energy markets. The study [1] proposed the concept of the BT-based MEM and built the framework for the BT-based MEM. This research validated that BT is a qualified technology for distributed MEM through simulating the performance of BT in Brooklyn microgrid. The study [19] studied the application of BT in the chemical industry by constructing a machine-to-machine electricity market. The article [20] used the decentralized traceability characteristics of BT to address the malicious actors in the carbon emissions trading system. That effectively confirmed that the application of BT in the carbon emissions trading system was feasible. The article [21] proposed the operation model and method of the decentralized distribution network to ensure the economic and cybersecurity of the power distribution network. It minimized the costs from eliminating the deviation of electricity through running multilateral trading of deviation power by bidding. The above research mainly focuses on the high cost and trust problems of centralized transaction management vulnerable to attack and the difficulty to guarantee user's privacy in the EI trade area. There are very few research articles on scalable information processing for the energy market of EI.

Based on the above analysis, BT is expected to achieve multiparty body automatic, reliable, fair, and real-time transactions in the EI system. However, BT has a scalability issue [22], and the EI system is complicated for the administration of energy, information, and capital flow. Furthermore, due to the insufficient consideration of the special attributes of energy goods, the EI system cannot promote the large-scale dynamic balance of resources and improve the efficiency of energy trading. To solve the problem, a scalable mechanism based on blockchain for information processing in energy trading is proposed. This mechanism constructs three modules: the BT-based Internet of Things (IoT) system, the BT-based virtual currency system, and the BT-based EI information platform to effectively manage energy, information, and capital flow. The three modules run in parallel, and their functions are independent. In this paper, we call this way as three-way communication (TWC) between merchants and consumers. In the process of TWC, to make the existing intelligent technologies be incorporated across the entire system, from energy generation, transmission, and distribution, to energy consumption at the customers' premises, the EI system was used to realize the open interconnection and coordination management of multienergy system. At the same time, with the aim of improving the reliability, efficiency, transparency, safety, and fair of the system, BT is the underlying technology. In addition, energy is different from general goods, when the BT-based energy trading lacks the supervision of the central organization, the determination of relevant standards can be solved through the implementation of weak centralization of energy transaction management. At the same time, the flows of energy, information, and capital are independent and need to be managed and run in parallel. As a result, a scalable information processing mechanism is implemented, which makes the energy exchange reasonable, secure, reliable, and flexible. Moreover, it can also promote a wide range of resource dynamic balance in the energy trading market, resulting in quick, efficient, and safe energy trading between supply and demand sides.

The contributions of our mechanism are summarized as follows:

(1) A scalable information process mechanism is proposed to improve the performance of the EI system. This mechanism can supply a safe, fair, transparent, flexible environment for the energy markets.

(2) In this mechanism, the energy, capital, and information flow in EI are addressed, respectively, by the TWC approach. In the TWC approach, we constructed three modules: the BT-based IoT system, the BT-based virtual currency system, and the BT-based EI information platform to improve the efficiency of energy trading.

(3) In energy trading, we consider the characteristic of energy. Each energy trading was audited through weak centralization management. The combination of the weak centralization management and the TWC approach makes the EI system scalable.

## 2. Relevant Work

In this section, the relevant theories for the proposed mechanism are described, including the EI energy market, blockchain technology, and weak centralized management.

*2.1. EI Energy Market.* The traditional energy market is mainly composed of exchange transactions and over-the-counter (OTC) [23]. In exchange transaction mode, market participants usually include users, central institutions, and third parties. Energy trading is planned and managed by the exchange center which is responsible for the overall balance of the energy market. Third-party institutions perform ratings, insurance, trust and financing, transaction clearing, and other types of work. Therefore, energy trading in the exchange mode will generate high third-party costs. Trading data is easy to be lost or tampered with, and user privacy is difficult to guarantee. Furthermore, trading information is asymmetric. Frequent information proofreading and high time cost are not conducive to efficient real-time transactions in EI. In OTC transaction mode, there is no fixed place to trade and there are no restrictions on market participants. Market makers are responsible for marketing organizations. Buyers and sellers negotiate prices outside the exchange through telephone and computer networks. These transactions are carried out by agents of market makers, and there is no limit to the number and units of transactions. Trading is basically a direct transaction between buyer and seller, and the price does not include the trading commission. However, the market maker who sets the bid-ask price will get a certain amount of money from the bid-ask spread he sets. Therefore, OTC transactions add additional market costs to energy trading.

EI has introduced the concept of the Internet on the basis of the traditional energy network. It has new connotations which are user-centered and distributed peer-to-peer sharing [24]. In EI, different energy industries are interconnected and managed in a coordinated way, many of which are connected to distributed renewable energy sources such as wind, solar, and tidal power. In the future, EI will cover a large number of users who may be consumers or prosumers. At the same time, energy trading will be characterized by diversification of subjects, diversification of commodities, decentralization of decision-making, transparency of information, and timeliness [25]. The new energy structure will be based on the distributed power generation technology, microgrid technology, smart grid technology [26–28], etc., and the energy production, manufacture, storage, and transportation will be coordinated and managed in EI. It is expected to reach comprehensive energy trading, including power transactions, primary energy transactions, and auxiliary power services transactions [23]. As a result, in the energy market under the EI environment, the user can voluntarily choose the type of energy service on the basis of the self-load characteristic, with the power exchange as the center. The transaction participants can independently decide whether to issue or purchase energy according to different energy service types in EI.

*2.2. Blockchain Technology.* BT, which is introduced as the underlying technology of bitcoin, is a database technology without a dedicated central authority to manage data [12, 29]. In a broader field of economic applications, the use of BT is called a verification mechanism for cryptocurrency. The blockchain of BT is a kind of data structure linked by chronological order [30, 31]. In this structure, blocks are the basic units of forming a blockchain, and relevant information and records are stored in blocks. Each block consists of a header and a body, where the header is used to link with the previous block. The body is used to record information. The network of blockchain is a peer-to-peer (P2P) network [32, 33], which has no central server and router, encapsulating specific communication protocols and data verification mechanisms. In the network, each node has a peer status and saves all the data information of the whole blockchain. All participants distributed in different locations jointly own the right of management and supervision data. Therefore, BT has the characteristics of decentration, transparency, and traceability.

The core technologies of blockchain include asymmetric encryption algorithm [34] and smart contracts [35, 36]. The former uses public key and private key to solve the problem of user information security and trust. Each participant has his own public and private key. After using one of the keys that encrypted information, only one other corresponding key is used to unlock it. The smart contract is a kind of digital contract similar to business rules, which stipulates the obligations of each party to fulfill and the judgment conditions of contract execution. For users, smart contracts are usually considered an automatic security account, whose code and status are in BT. Therefore, in an information system based on BT, it is expected to reach complete decentralization by using cryptographic security, smart contracts, and a decentralized consensus mechanism. For all this, BT still has some problems, such as scalability issues, limited transaction loads, and its complex technical protocol and implementation [22]. Nevertheless, BT is an emerging technology with the potential to provide new ways to benefit various systems.

*2.3. Weak Centralized Management.* Chai et al. [23] suggested that energy with different physical properties, which is the national strategic resources, is related to the national economy and people's livelihood. Therefore, energy trading is different from ordinary commodity trading. It is essential to give certain regulatory measures about the access and exit of participants of the energy market, the process of transaction, etc. In addition, it is difficult to solve the problem completely when there are disputes in the transaction only by the system itself and the enforcement of intelligent contracts. The system, whose trading is completely decentralized, is not suitable for the energy market. Some supervision for the management of energy transactions based on BT is needed, which is called weak centralized management.

Weak central management is implemented by weak central institutions, which have the right to manage accounts and supervise transactions. It is a special node in the blockchain trading network. This node is different from the nodes which participate in the process of energy trading game and independently completes the determination of energy price and transaction records. The node is responsible for such work as confirmation of transaction qualification and contract review. It does not have the right to store energy transaction information and modify the information. Relevant modifications are subject to the consent of the nodes involved in energy trading. Therefore, in the implementation of a weak central management system for energy trading, weak central agencies should audit user qualifications and approve the right users obtained to register. After registering, the users' information is broadcast to all network nodes of the P2P network. When all the nodes reach a consensus, account registration is successful, and access to energy transactions is allowed. All the transaction information generated by these users shall be checked by the weak central institution so that the legitimate transaction can be broadcast to all network nodes. At the same time, transaction information can be stored in the blockchain. If users exit the energy market, the notification of exit will be issued in the whole network. After the nodes in the network reach consensus, the central institution will cancel the registration information in time.

## 3. The Proposed Mechanism

In this section, based on BT, we derive TWC for the efficient operation of the EI market. The aim of the TWC is to overcome the shortcomings of BT to make the information system scalable with a high trading volume of business. A schematic diagram of the EI based on TWC is given in Figure 1. In the framework, the source, grid, load, and reserve are composed of the modules of the BT-based EI energy market. The modules of the market are closely integrated and coordinated with the administration of the EI system. In virtue of energy technologies such as energy production, transmission, distribution, and use, the production, conversion, reserve, and other links can effectively execute scheduling, controlling, managing, and using in the source, grid, load, and reserve. In these modules, the source represents the traditional energy supply, distributed energy which mainly includes RES. These energies are transported and distributed to industrial parks, business enterprises, and individual users by energy transmission and distribution systems. Reserves refer to battery energy, capacitive energy, and pumped storage which are produced by energy enterprises, communities, or prosumers. These energies can be supplied to users through energy storage control technology. Moreover, the energies in sources and reserves can be transformed for heat, electricity, and cold energy. The devices, which derive from source, grid, load, and reserve, are interlinked by IoT technology to form the physical network of EI. Considering that there is a large amount of personal data in IoT, and there is arguably no single point of failure or vulnerability in blockchain, to strengthen cybersecurity and protect privacy, this paper proposes that the storage of energy information derived from the devices is based on BT to form the energy flow (shown in Figure 1 with green thread). BT-based IOT is responsible for the management of the energy flow like the logistics companies, whose functions are independent. In Figure 1, the red line represents the capital flows which derive from the financial transactions between buyers and sellers, e.g., clearing information, the pattern of payment, payment amount, etc. Taking into account the BT-based bitcoin system, which is a virtual currency system, it has the advantage of distributed accounting. To construct a multisignature-protected decentralized marketplace, this paper proposes the BT-based administration of capital flow. In this way, user information will not be managed and stored. Even if a hacker successfully breaches the BT-based EI system, the private key of the virtual wallet cannot be accessed. Another blue line represents the information flow generated by communication between energy utilities and consumers, such as proof of identity for individuals as well as assets, user information, and system operation status, since the running of energy and capital flow can produce information from energy and finance trading. To effectively manage the information, we propose that the role of administration for information flow is played by the BT-based EI information platform. The platform can implement integrated information from the energy market for the user to access. Based on the above description, the energy, capital, and information flow are three breaches of the BT-based EI system, respectively. In this paper, we call the approach for the three breaches TWC. The TWC-based scalable information process mechanism will be described in detail.

*3.1. TWC-Based EI.* The specific design for the building elements of various blockchains can vary according to the specific use case. For the BT-based EI energy market, it has a distributed software architecture that allows for a complete and continuously tracking of even the smallest energy transactions. Its main advantages are transparent, distributed, and secure transaction logs. However, it has scalability issues. Furthermore, a certain amount of information confirmation cost has to be generated in the process of energy trading creation. Thus, as an information system, a

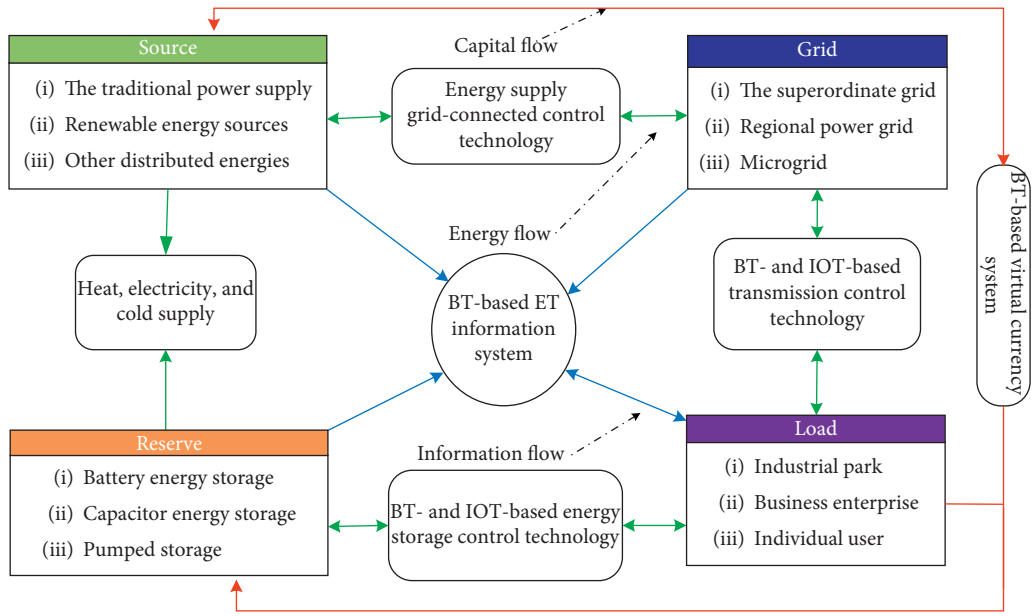



Figure 1: A schematic diagram of TWC-based EI.

trade-off between high information accuracy and storage mode needs to be considered. To improve the scalability of the BT-based EI system and the efficiency of energy trading, we believe that the market participants in EI will be a less costly hash-based user authentication mechanism. However, the level of security and resilience required for efficient integration of distributed energy resources may not be ensured through proof-of-identity authentication mechanisms. Therefore, we propose a TWC-based EI energy trading system. In the system, there are three ways to manage this information flow. One way is to correct and store information by the BT-based IoT system. The second way is to address financial transactions by BT-based virtual currency system. The third way is to manage trading information by BT-based EI information platform.

*3.2. BT-Based IoT System.* In the traditional EI system, the traditional IoT has a data center, which collects information from the connected devices. Furthermore, the information derived from devices is stored in a centralized cloud model. This way may become even more problematic and potentially risky as the number of network nodes increases. For example, some smart devices violate personal privacy, and government security agencies can review data stored in central servers without authorization [37]. Operators can also sell users' private data to advertising companies for commercial profit [38]. Blockchain provides a good way to solve these problems through promoting collaboration between transaction processing and interactive devices in the decentralized IoT. Moreover, the IoT can be scaled up to ensure privacy, security, and the establishment of untrusted transactions. Thus, the BT-based IoT approach would be more suitable and more effective for managing energy information in the EI energy market.

In IoT, the interconnection devices are several feet apart from each other; the connection between them needs to go through EI. Most devices of IoT send data to EI and messages can be sent from the EI to the devices, where processing and storage are often carried out. If a device connected to a server is hacked, it can disrupt the whole network. On the other hand, it is difficult for the data center to ensure the validity and management of data. The BT-based IoT system can ensure that the good and proper use of information and the message exchanges between devices can be treated as financial transactions in a bitcoin network. From Figure 1, the energy information in the model of source, grid, load, and reserve is sent through the blockchain to verify these cryptographic signatures to ensure that only the originator of the message could send it. At the same time, the messages are exchanged through the blockchain cryptographically sign transactions. Thus, when energy trading with security holes is not returned, the owner of the device can be tracked and easily contacted.

In the EI system, the devices of IOT mainly include energy router, energy exchange, and energy interface. These devices connect the server as the back end in the EI system. Among them, energy routers implement the communication connection with the back end. Energy exchanges supply energy information for the subnet, such as total input, output, and storage of electrical energy; total generation, consumption, and storage of thermal energy; and storage and consumption of other forms of energy. Energy interfaces are the port of transportation, petrochemicals, and other networks. Among them, various distributed renewable energy, devices for generation of power and heating, and distributed energy storage devices join up EI to achieve the identification of the distributed devices, the detection of the status of the distributed devices, and various forms of energy calibration. The schematic diagram of the BT-based IoT

system is shown in Figure 2. From Figure 2, the accessing devices join up the EI through the devices of IoT to construct the subnet of IoT. All subnets consist of the physical networks of EI. Its operation achieves distribution and management of energy flow, safe and reliable network operation, efficiency, and economy of energy use.

*3.3. BT-Based EI Information Platform.* In Section 3.2, the BT-based IoT system, which constitutes the physical network of EI, is described. The energy information flows EI energy market that combines BT-based virtual currency system and BT-based EI information platform to produce energy trading information, financial information, and communication information, all of which are accessed and used by the user in BT-based EI information platform. This process is implemented by the virtual network of EI. The schematic diagram of the virtual network of EI is shown in Figure 3. From Figure 3, the energy flow is represented by green threads. The capital flow (shown in Figure 3 with red threads) was managed by a BT-based virtual currency system. The communication information flow is shown with blue threads.

In the EI energy market, users incarnate the sellers and buyers, namely, the participants, such as energy traders, energy enterprises, and energy agents. In BT, smart contracts have the characteristic of the absence of accountability. The subject of energy trading is often a virtual account, and there are a series of ethical problems to be solved. If users want to log in to the EI energy market by registering at the EI information platform, the identity of the user needs authentication. Therefore, we proposed to introduce a "digital identity authentication service" which is based on BT. At the same time, the process of authentication is implemented as weak center data management and provides different kinds of digital identity authentication services. That is to say, the third-party organization, which proves the personal details, but does not store the user information, undertakes the regulatory task to audit the information of participants. In this way, it is to ensure the rationality and legitimacy of the energy traders.

Based on the above analysis, the authorized users enter the EI energy market for energy trading. In this platform, users can not only order energy from the energy traders or sell energy to the energy traders but also buy/sell energy to each other. When they conduct energy trading, they trust each other by a consensus mechanism derived from a certain mathematical algorithm. Moreover, the authentication of energy information follows the data encryption principle of blockchain technology. The distributed "bookkeeping" based on blockchain and metrological certification which is authenticated by all nodes of the whole system can guarantee the authority of measurement and certification for energy information. Therefore, the energy information which is produced by the BT-based IoT system is safe and reliable. Furthermore, since the smart contracts can guarantee the unification and synchronization of the control modes, control interval time, and signal instructions of multienergy systems with different physical characteristics and scheduling modes, energy trading can be automatically accomplished the underlying communication between buyers and sellers by means of BT-based EI information platform. However, considering that the energy commodities are different from general goods, energy trading cannot be completely out of the country's credit endorsement. It needs safeguards and supervision of policy. Thus, when energy derived from the electric power system, heating system, gas system, etc. enters the energy market to participate in trading, third-party organizations are needed to audit the trading information. In energy trading, the third-party organization is the node of EI, but it does not have the jurisdiction to store the trading information. In financial trading, the information on energy trading is used for clearing transactions. Users can freely choose the means of payment according to the actual situation and preferences and complete the receiving and payment transactions in the BT-based EI information platform.

*3.4. BT-Based Virtual Currency System.* From Figure 3, the virtual currency is issued, circulated, and traded in the system. The financial trading between buyers and sellers is based on a BT-based virtual currency system. At the same time, this system achieves the issuance and renewal of virtual currency, and the circulation of virtual currency can be driven by energy trading in the BT-based EI energy market. Energy participants play the role of each network node of virtual currency. They actively engage in energy trading to compete with the award by a game approach which ensures a stable virtual currency consensus and security. The system of virtual currency issues a certain number of virtual currencies for a new block to reward the creator. To establish revenue sharing fees, partial participants can cooperate with each other to play the game together. Thus, after the issuance of the virtual currency, it goes into circulation. Users who own virtual currency can pay virtual currency for energy and services through the BT-based EI information platform. It is worth noting that this virtual currency is different from the bitcoin, its value is steadfast, and its attributes are various at different times. As we all know, the blockchain has the characteristic of timestamps. This paper proposes that the issuance and renewal of virtual currency are based on blockchain and the virtual currency can have the characteristic of timestamp. We give the different attributes of virtual currency in view of its different issuance times. For example, to obtain different amounts of virtual currency as incentives, buyers purchase various energies at different times. At the same time, the virtual currency with various attributes has different operating life.

In view of the above analysis, how much energy participants gain to depend on the outcome of the game for energy trading. The game winner gains and gets the right to pack the blocks. In addition, because of the flexibility of the energy market, investors or speculators can carry out virtual currency trading based on BT-based virtual currency systems. Such trading can be developed between the participants of energy trading. The financial institutions can take part in the virtual currency as participants to promote the

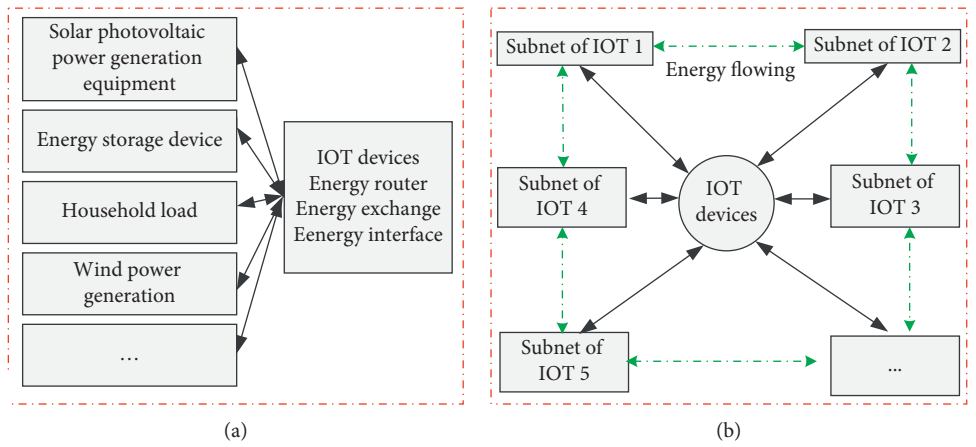



Figure 2: The schematic diagram of the BT-based IoT system. (a) Subnet of IoT. (b) The physical network of EI.

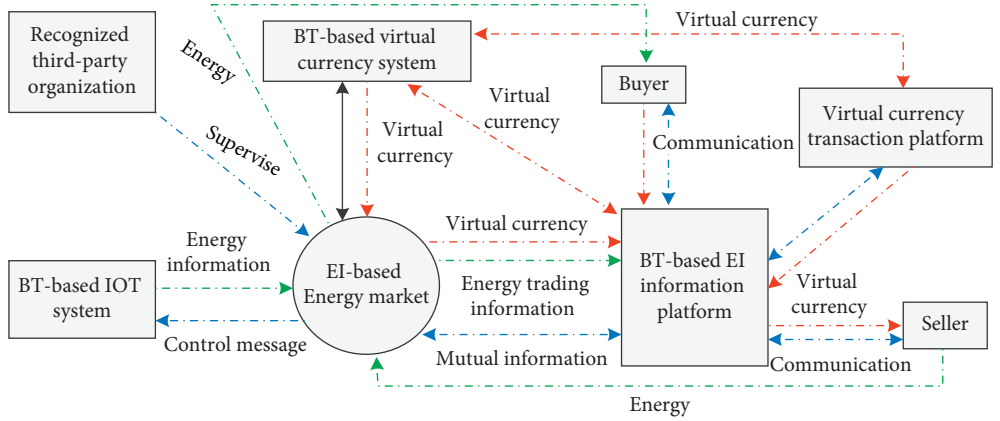



Figure 3: The schematic diagram of the BT-based virtual currency system.

development of the energy market. Regardless of financial trading, after the transaction is completed, each transaction will be validated by all nodes in the BT-based EI and recorded in blocks.

The main contributions of this paper are summarized as follows:

(1) We present an overview of the literature for the EI energy market, BT, weak central management, and their combination.

(2) Based on this existing literature, we proposed to construct three modules of BT-based IoT system, BT-based virtual currency system, and BT-based EI information platform for the efficient design and operation of BT-based EI energy market.

(3) Thus, we developed the TWC approach and module-based weak central management to improve the scalability of the EI system.

However,

(4) The TWC approach which is based on three modules still needs to be tested and adequately adapted for the future energy market. The implementation of the TWC approach requires further development to adequately implement intelligent supply-demand response strategies.

(5) The regulation of weak centers needs to be adapted for allocation efficiency. The corresponding smart contracts seem to need different departments to formulate.

## 4. Conclusion

This paper presents a scalable mechanism based on blockchain for information processing in energy trading. In the mechanism, a communication approach (called TWC) between merchants and consumers is proposed to deal with energy trade by incorporating the existing intelligent technologies, especially, the BT technology, across the entire system. We have derived three modules (including BT-based IoT system, BT-based virtual currency system, and BT-based EI information platform) and a weak centralized management mechanism to implement TWC. In the process of TWC, the problem of energy flow was addressed by a BT-based IoT system. This system can avoid the whole network to be disrupted because the simple notes are hacked. At the

same time, because of the characteristics of BT, it strengthens cybersecurity and privacy protection and ensures the correctness of data. In addition, the capital flow is managed through a BT-based virtual currency system. This system is like the bitcoin system, but it is different. In this study, the system launches and updates virtual currencies with different attributes at different times, to improve the efficiency of the energy market. Furthermore, the communication information is administrated by a BT-based EI information platform. In this platform, buyers and sellers have completed the auditing of identity authentication from the third-party organizations. The authorized users can use the game strategy to carry out energy trading. As a result, the information on energy trading is used for clearing to generate financial information. Finally, the winner in this game gets the right to package the information into a blockchain and broadcast it to all nodes in the network. With all that, we put forward that there are several advantages of the proposed mechanism. At the same time, the current type of BT will promote various industrial and academic projects in the EI energy market. The project's findings need to be further tested to evaluate economic and socioeconomic impacts. Thus, it is hoped that this study can provide a reference for researchers of BT-based EI.

## Data Availability

The corresponding data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 71420107027).

## References

[1] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: a case study: the Brooklyn Microgrid," *Applied Energy*, vol. 210, pp. 870–880, 2018.

[2] L. Tianguang and A. Qian, "Interactive energy management of networked microgrids-based active distribution system considering large-scale integration of renewable energy resources," *Applied Energy*, vol. 163, pp. 408–422, 2016.

[3] J. Vardakas, N. Zorba, and C. Verikoukis, "A survey on demand response programs in smart grids: pricing methods and optimization algorithms," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 607–178, 2015.

[4] A. Monacchi and W. Elmenreich, "Assisted energy management in smart microgrids," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 6, pp. 901–913, 2016.

[5] A. Nieße, S. Lehnhoff, M. Tröschel et al., "Market-based self-organized provision of active power and ancillary services: an agent-based approach for smart distribution grids," *Aachen, Germany: IEEE Complexity in Engineering (COMPENG)*, pp. 1–5, 2012.

[6] A. Ahl, M. Yarime, K. Tanaka, and D. Sagawa, "Review of blockchain-based distributed energy: implications for institutional development," *Renewable and Sustainable Energy Reviews*, vol. 107, pp. 200–211, 2020.

[7] V. Lesic, A. Martinčević, and M. Vasak, "Modular energy cost optimization for buildings with integrated microgrid," *Applied Energy*, vol. 197, pp. 14–28, 2017.

[8] P. Kou, D. Liang, and L. Gao, "Distributed EMPC of multiple microgrids for coordinated stochastic energy management," *Applied Energy*, vol. 185, pp. 939–952, 2017.

[9] V. N. Coelho, M. Weiss Cohen, I. M. Coelho, N. Liu, and F. G. Guimarães, "Multi-agent systems applied for energy systems integration: state-of-the-art applications and trends in microgrids," *Applied Energy*, vol. 187, pp. 820–832, 2017.

[10] R. Schleicher-Tappeser, "How renewables will change electricity markets in the next five years," *Energy Policy*, vol. 48, pp. 64–75, 2012.

[11] K. Boroojeni, M. H. Amini, A. Nejadpak, T. Dragicevic, S. S. Iyengar, and F. Blaabjerg, "A novel cloud-based platform for implementation of oblivious power routing for clusters of microgrids," *IEEE Access*, vol. 5, pp. 607–619, 2017.

[12] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, 2019.

[13] M. Yang, T. Zhu, K. Liang, W. Zhou, and R. H. Deng, "A blockchain-based location privacy-preserving crowdsensing system," *Future Generation Computer Systems*, vol. 94, pp. 408–418, 2019.

[14] N. Zhang, Y. Wang, C. Q. Kang, J. N. Cheng, and D. W. He, "Blockchain technique in the energy internet: preliminary research framework and typical applications," *Proceedings of the CSEE*, vol. 36, pp. 4011–4022, 2016.

[15] O. V. Cutsem, D. H. Dac, P. Boudou, and M. Kayal, "Cooperative energy management of a community of smart-buildings: a Blockchain approach," *Electrical Power and Energy Systems*, vol. 117, pp. 1–11, 2020.

[16] M. Li, S. J. Shao, Q. W. Ye, G. Y. Xu, and G. Q. Huang, "Blockchain-enabled logistics finance execution platform for capital-constrained E-commerce retail," *Robotics and Computer Integrated Manufacturing*, vol. 65, pp. 1–14, 2020.

[17] Y. F. Li, B. Wang, and D. Yang, "Research on supply chain coordination based on block chain technology and customer random demand," *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 4769870, 2019.

[18] A. Miglani, N. Kumar, V. Chamola, and S. Zeadally, "Blockchain for internet of energy management: review, solutions, and challenges," *Computer Communications*, vol. 151, pp. 395–418, 2020.

[19] J. J. Sikorski, J. Haughton, and M. Kraft, "Blockchain technology in the chemical industry: machine-to-machine electricity market," *Applied Energy*, vol. 195, pp. 234–246, 2017.

[20] K. N. Khaqqi, J. J. Sikorski, K. Hadinoto, and M. Kraft, "Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application," *Applied Energy*, vol. 209, pp. 8–19, 2018.

[21] Z. Li, S. Bahramirad, A. Paaso, M. Yan, and M. Shahidehpour, "Blockchain for decentralized transactive energy management system in networked microgrids," *The Electricity Journal*, vol. 32, no. 4, pp. 58–72, 2019.

[22] R. Beck, J. S. C. Stenum, N. Lollike, and S. Malone, "Blockchain-the gateway to trust-free cryptographic transactions," in *Proceedings of the European Conference on Information Systems*, pp. 1–14, Istanbul, Turkey, June 2016.

[23] J. Q. Cai, S. X. Li, B. Fan, and L. R. Tang, "Blockchain based energy trading in energy internet," *Electric Power Construction*, vol. 38, pp. 24–31, 2017.
[24] C. Park and T. Yong, "Comparative review and discussion on P2P electricity trading," *Energy Procedia*, vol. 128, pp. 3–9, 2017.
[25] J. Reynolds, Y. Rezgui, and J.-L. Hippolyte, "Upscaling energy control from building to districts: current limitations and future perspectives," *Sustainable Cities and Society*, vol. 35, pp. 816–829, 2017.
[26] W. P. Zarakas, "Growth prospects and shifting electric utility business models: retail, wholesale and telecom markets," *The Electricity Journal*, vol. 28, no. 5, pp. 59–67, 2015.
[27] C. Zhang, J. Wu, M. Cheng, Y. Zhou, and C. Long, "A bidding system for peer-to-peer energy trading in a grid-connected microgrid," *Energy Procedia*, vol. 103, pp. 147–152, 2016.
[28] E. Melville, I. Christie, K. Burningham, C. Way, and P. Hampshire, "The electric commons: a qualitative study of community accountability," *Energy Policy*, vol. 106, pp. 12–21, 2017.
[29] L. Zhong, Q. Wu, J. Xie, J. Li, and B. Qin, "A secure versatile light payment system based on blockchain," *Future Generation Computer Systems*, vol. 93, pp. 327–337, 2019.
[30] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," *Fast Software Encryption*, vol. 3017, pp. 371–388, Springer Berlin Heidelberg, New York, USA, 2004.
[31] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.
[32] A. Savelyev, "Copyright in the blockchain era: promises and challenges," *Computer Law & Security Review*, vol. 34, no. 3, pp. 550–561, 2018.
[33] M. Muzammal, Q. Qu, and B. Nasrulin, "Renovating blockchain with distributed databases: an open source system," *Future Generation Computer Systems*, vol. 90, pp. 105–117, 2019.
[34] L. Diestelmeier, "Changing power: shifting the role of electricity consumers with blockchain technology - policy implications for EU electricity law," *Energy Policy*, vol. 128, pp. 189–196, 2019.
[35] S. E. Chang, Y.-C. Chen, and M.-F. Lu, "Supply chain reengineering using blockchain technology: a case of smart contract based tracking process," *Technological Forecasting and Social Change*, vol. 144, pp. 1–11, 2019.
[36] X. Wang, W. Yang, S. Noor, C. Chen, M. Guo, and K. H. van Dam, "Blockchain-based smart contract for energy demand management," *Energy Procedia*, vol. 158, pp. 2719–2724, 2019.
[37] Z. Guan, X. Lu, W. Yang, L. Wu, N. Wang, and Z. Zhang, "Achieving efficient and Privacy-preserving energy trading based on blockchain and ABE in smart grid," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 34–45, 2021.
[38] S. Zhang, J. Rong, S. Noor, and B. Wang, "A privacy protection scheme of smart meter for decentralized smart home environment based on consortium blockchain," *Electrical Power and Energy Systems*, vol. 121, pp. 1–10, 2020.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4085718, 14 pages
https://doi.org/10.1155/2020/4085718

*Research Article*

# A Two-Stage Three-Machine Flow Shop Assembly Problem Mixed with a Controllable Number and Sum-of-Processing Times-Based Learning Effect by Simulated Annealing Algorithms

**Shang-Chia Liu**

*Department of Business Administration, FuJen Catholic University, New Taipei City 242062, Taiwan*

Correspondence should be addressed to Shang-Chia Liu; 056298@mail.fju.edu.tw

Received 31 July 2020; Revised 2 September 2020; Accepted 23 September 2020; Published 10 October 2020

Academic Editor: Dujuan Wang



The two-stage assembly scheduling problem is widely used in industrial and service industries. This study focuses on the two-stage three-machine flow shop assembly problem mixed with a controllable number and sum-of-processing times-based learning effect, in which the job processing time is considered to be a function of the control of the truncation parameters and learning based on the sum of the processing time. However, the truncation function is very limited in the two-stage flow shop assembly scheduling settings. Thus, this study explores a two-stage three-machine flow shop assembly problem with truncated learning to minimize the makespan criterion. To solve the proposed model, we derive several dominance rules, lemmas, and lower bounds applied in the branch-and-bound method. On the other hand, three simulated annealing algorithms are proposed for finding approximate solutions. In both the small and large size number of job situations, the SA algorithm is better than the JS algorithm in this study. All the experimental results of the proposed algorithm are presented on small and large job sizes, respectively.

## 1. Introduction

For decades, scheduling models usually assumed that the processing time of the job is known and fixed [1]. But, in a real production system, both the efficiency of the machine and the skill of the worker can increase with the working time, thereby reducing the actual work processing time with the development of production. Many researchers indicated certain types of learning effects, such as [2] in the position-based learning effect and [3] in the time-dependent learning effect. This topic continues to attract a lot of research interest [4–6].

On the other hand, Biskup, Kuo and Yang [2, 3], and Kuolamas and Kyparisis [7] introduced models that involve a learning effect on two-machine scheduling or flow shop scheduling settings following the same or different learning ideas. There are a multitude of studies related to two-machine scheduling or flow shop scheduling settings with the learning effect consideration, including [8–11] and [12]. Besides, there are some two-stage models also with the learning effect consideration. Wu et al. [13] adopted the learning model developed by Biskup [2] to solve a two-stage flow shop scheduling problem with three machines to minimize the makespan. They devised three simulated annealing (SA) algorithms and three cloud theory-based SA algorithms. Adopting the learning model in [3, 14], a branch-and-bound (B&B) algorithm is devised incorporating six (hybrid) particle swam optimization (PSO) methods to solve the two-stage flow shop assembly scheduling problem to minimize the total job completion time.

Most studies using the learning model are applied in the single-machine or flow shop setting. However, the two-stage assembly scheduling problem is relatively unexplored. Recently, Wu et al. and Zou et al. [15, 16] considered position-based learning in connection with a two-stage assembly scheduling problem. They proposed different versions of simulated annealing and cloud theory-based simulated annealing to solve this problem. Moreover, Wu et al. [17] considered this problem with general learning effects. They developed and evaluated the performances of six hybrid

particle swarm optimizations (PSO). Wu et al. [18] have conducted the research about a combined approach for two-agent scheduling with a sum-of-processing-times-based learning effect. To the best of our knowledge, no other research considers the two-stage assembly scheduling problem with learning effects. Table 1 presents a comparative study of existing models to solve the considered scheduling problem. Many studies claim that most of the productive items in manufacturing systems may be formulated in a two-stage assembly scheduling model. However, the literature neglects accumulated learning experience in solving a two-stage assembly scheduling problem. In fact, the sum-of-processing times-based learning model is pertinent to process manufacturing in which an initial setup is often followed by a lengthy uninterrupted production process. Motivated by this observation, this study introduces the 2-stage 3-machine assembly problem with a sum-of-processing times of already processed jobs learning to minimize the makespan criterion. This model assumes that there are two machines in the first stage and an assembly machine in the second stage. This study first provides some dominances and a lower bound applied in the branch-and-bound method for an optimal solution. Three heuristics modified from Johnson's rule with and without improving by an interchange pairwise method are, then, separately applied in three simulated annealing algorithms for finding near-optimal solutions. Finally, the statistical results of the three proposed algorithms are evaluated and reported.

Table 1: Small size number of jobs parameters.

| Parameter | $n = 10,\ m = 3$ |
|---|---|
| $T_i$ | $10^{-3}$ |
| $T_f$ | $10^{-8}$ |
| $c_f$ | 0.95 |
| $N_r$ | 10 |
| $\beta$ | 0.2 |
| $\alpha_1$ | −0.1 |
| $\alpha_2$ | −0.1 |
| $\alpha_3$ | −0.2 |

## 2. Problem Statement

The considered two-stage assembly scheduling problem is formally described in this section. We assume a series of $n$ given jobs to be processed on three machines. The main idea of the studied problem is executing $n$ given jobs on three machines in two stages. Each job has strictly more than two operations. For the operations of the $n$ job, the first stage is performed in a parallel way in two machines. The operation processing of the second stage only begins when the operations of the first stage are completed. All jobs are available at time zero. Each job $j$ is decomposed on three tasks as follows: the ordinary processing times (without learning effect) of job $j$ are $a_j$, $b_j$, and $c_j$ on machines $M_1$, $M_2$, and $M_3$, respectively. Other assumptions are no machine can process more than one job at a time; no idle time is allowed on machines $M_1$ and $M_2$; and once a job has begun processing, it cannot be interrupted.

Following the works of Kuo and Yang [3] and Liu et al. [19], the real processing times of $J_j$, if $J_j$ scheduling on the position $r$ of a job sequence $S$, are considered as $a_j \max\{\alpha, (1 + \sum_{k=1}^{r-1} a_{[k]})^{a_1}\}$, $b_j \max\{\alpha, (1 + \sum_{k=1}^{r-1} b_{[k]})^{a_2}\}$, and $c_j \max\{\alpha, (1 + \sum_{k=1}^{r-1} c_{[k]})^{a_3}\}$ on machines $M_1$, $M_2$, and $M_3$, respectively, where $\alpha$ denotes the controllable parameter with $\alpha < 0$, and $a_1, a_2, a_3$ denote the learning indices for $M_1$, $M_2$, and $M_3$. For a schedule $S$, the finishing time of a job, say $J_j$, to be scheduled on the $r$th position (in $S$) on machine $M_3$ is defined, and denoted, as $C_{3[r]}(S)$:

$$C_{3[r]}(S) = \max\left\{t_1 + a_j \max\left\{\alpha, \left(1 + \sum_{k=1}^{r-1} a_{[k]}\right)^{a_1}\right\}, t_2 + b_j \max\left\{\alpha, \left(1 + \sum_{k=1}^{r-1} b_{[k]}\right)^{a_2}\right\}, t_3\right\} + c_j \max\left\{\alpha, \left(1 + \sum_{k=1}^{r-1} c_{[k]}\right)^{a_3}\right\}, \tag{1}$$

where $t_1$, $t_2$, and $t_3$ are denoted as the beginning times of $J_j$ on machines $M_1$, $M_2$, and $M_3$ in $S$, respectively. The optimal criterion of this study is to find a scheduled $S$ to minimize the makespan (or $C_{3[n]}(S)$).

## 3. Lower Bounds and Some Lemmas

This section proposes some useful lower bounds used in a branch-and-bound method. Before deriving the lower bounds, we define some notations. Suppose $t_1$, $t_2$, and $t_3$ denote the starting time of the first job in $s_{\mathrm{US}}$ on all three machines. Then, we assume that $s_{\mathrm{PS}}$ are the partially scheduled $k$ jobs and $s_{\mathrm{US}}$ are the remaining unscheduled $(n-k)$ jobs. $a_{(1)} \le a_{(2)} \le \cdots \le a_{(n-k)}$, $b_{(1)} \le b_{(2)} \le \cdots \le b_{(n-k)}$, and $c_{(1)} \le c_{(2)} \le \cdots \le c_{(n-k)}$ denote increasing sequences of $a_1, a_2, \ldots, a_{n-k}$; $b_1, b_2, \ldots, b_{n-k}$; and $c_1, c_2, \ldots, c_{n-k}$ in $\sigma_{\mathrm{US}}$, respectively. Notably, $a_i, b_i, c_i$ are not necessarily from the same job, and $s_{\mathrm{spt1}}$, $s_{\mathrm{spt2}}$, and $s_{\mathrm{spt3}}$ are three subsequences of $n$ jobs $n$ increasing sequences of $a_1, a_2, \ldots, a_{n-k}$; $b_1, b_2, \ldots, b_{n-k}$; and $c_1, c_2, \ldots, c_{n-k}$ in $s_{\mathrm{US}}$, respectively.

Additionally, by using the ideas of Kuo and Yang [3] and Liu et al. [19], a lower bound for a subsequent can be yielded as follows:

$$C_{[k+1]}(s) = \max\left\{t_1 + a_{[k+1]}\max\left[\alpha, \left(1+\sum_{i=1}^{k} a_{[i]}\right)^{a_1}\right], t_2 + b_{[k+1]}\max\left[\alpha, \left(1+\sum_{i=1}^{k} b_{[i]}\right)^{a_2}\right], t_3\right\}$$
$$+ c_{[k+1]}\max\left[\alpha, \left(1+\sum_{i=1}^{k} c_{[i]}\right)^{a_3}\right]$$
$$\geq t_1 + a_{[k+1]}\frac{\alpha + \left(1+\sum_{i=1}^{k} a_{[i]}\right)^{a_1}}{2} + c_{[k+1]}\frac{\alpha + \left(1+\sum_{i=1}^{k} c_{[i]}\right)^{a_3}}{2}$$
$$C_{[k+2]}(s) \geq t_1 + a_{[k+1]}\frac{\alpha + \left(1+\sum_{i=1}^{k} a_{[i]}\right)^{a_1}}{2} + a_{[k+2]}\frac{\alpha + \left(1+\sum_{i=1}^{k} a_{[i]}\right)^{a_1}}{2} + c_{[k+2]}\frac{\alpha + \left(1+\sum_{i=1}^{k} c_{[i]}\right)^{a_3}}{2} \quad (2)$$
$$\vdots$$
$$C_{[k+n_1]}(s) \geq t_1 + \sum_{j=1}^{n_1} a_{[k+i]}\frac{\alpha + \left(1+\sum_{i=1}^{k+j-1} a_{[i]}\right)^{a_1}}{2} + c_{[k+n_1]}\frac{\alpha + \left(1+\sum_{i=1}^{k+n_1-1} c_{[i]}\right)^{a_3}}{2}$$
$$\text{lower } bd_1 = t_1 + \sum_{j=1}^{n_1} a_{(n-j+1)}\frac{\alpha + \left(1+\sum_{i=1}^{k+j-1} a_{(i)}\right)^{a_1}}{2} + c_{(1)}\frac{\alpha + \left(1+\sum_{i=1}^{n-1} c_{(i)}\right)^{a_3}}{2}.$$

In a similar way,

$$\text{lower } bd_2 = t_2 + \sum_{j=1}^{n_1} b_{(n-j+1)}\frac{\alpha + \left(1+\sum_{i=1}^{k+j-1} b_{(i)}\right)^{a_2}}{2}$$
$$+ c_{(1)}\frac{\alpha + \left(1+\sum_{i=1}^{n-1} c_{(i)}\right)^{a_3}}{2}, \quad (3)$$
$$\text{lower } bd_3 = t_3 + \sum_{j=1}^{n_1} c_{(n-j+1)}\frac{\alpha + \left(1+\sum_{i=1}^{k+j-1} c_{(i)}\right)^{a_3}}{2},$$

$$\text{lower } bd = \max\{\text{lower } bd_1, \text{lower } bd, \text{lower } bd_3\}.$$

In what follows, we will propose four properties used in a branch-and-bound to improve its search power. Before deriving our properties, we first give four useful lemmas which will be used in the proof of Property 1.

**Lemma 1.** *If* $y \geq x > 0$, $T > 0$, *and* $e < 0$, *then* $y(T+x)^e \geq x(T+y)^e$.

**Lemma 2.** *If* $\lambda \geq 1$, $0 \leq t$, *and* $e < 0$, *then* $\lambda[1-(1+t)^e] \geq 1-(1+\lambda t)^e$.

Lemmas 1 and 2 have been proved by Kuo and Yang and Liu et al. [3, 19], respectively.

**Lemma 3.** *If* $y \geq x > 0$, $T > 0$, $e < 0$, *and* $0 < \alpha < 1$, *then* $y \cdot \max\{\alpha, (T+x)^e\} \geq x \cdot \max\{\alpha, (T+y)^e\}$.

*Proof.* Note that $(T+x)^e \geq (T+y)^e$. □

*Case 1.* As $\alpha > (T+x)^e$,

$$y \cdot \max\{\alpha, (T+x)^e\} = y \cdot \alpha,$$
$$x \cdot \max\{\alpha, (T+y)^e\} = x \cdot \alpha. \quad (4)$$

So, the inequality holds.

*Case 2.* As $(T+x)^e \geq \alpha > (T+y)^e$,

$$y \cdot \max\{\alpha, (T+x)^e\} = y \cdot (T+x)^e,$$
$$x \cdot \max\{\alpha, (T+y)^e\} = x \cdot \alpha. \quad (5)$$

The inequality holds.

*Case 3.* As $(T+y)^e \geq \alpha$,

$$y \cdot \max\{\alpha, (T+x)^e\} = y \cdot (T+x)^e,$$
$$x \cdot \max\{\alpha, (T+y)^e\} = x \cdot (T+y)^e. \quad (6)$$

From Lemma 1, the inequality holds.

**Lemma 4.** *If* $\lambda \geq 1$, $t \geq 0$, $e < 0$, *and* $0 < \beta < 1$, *then* $\lambda[\max\{\beta, 1\} - \max\{\beta, (1+t)^e\}] \geq \max\{\beta, 1\} - \max\{\beta, (1+\lambda t)^e\}$.

*Proof.* Note that $(1+\lambda t) \geq 1+t > 0$; thus, $(1+\lambda t)^e \leq (1+t)^e \leq 1$ because $e < 0$. □

*Case 1.* As $\beta \leq (1+\lambda t)^e \leq 1$,

$$\lambda[\max\{\beta, 1\} - \max\{\beta, (1+t)^e\}] = \lambda[1-(1+t)^e],$$
$$\max\{\beta, 1\} - \max\{\beta, (1+\lambda t)^e\} = 1-(1+\lambda t)^e. \quad (7)$$

From Lemma 2, the inequality holds.

*Case 2.* As $(1+\lambda t)^e < \beta < (1+t)^e \le 1$,

$$\begin{aligned}\lambda[\max\{\beta,1\} - \max\{\beta,(1+t)^e\}] &= \lambda[1-(1+t)^e],\\ \max\{\beta,1\} - \max\{\beta,(1+\lambda t)^e\} &= 1-\beta,\end{aligned} \tag{8}$$

which is less than $1-(1+\lambda t)^e$. From Lemma 2, the equality holds.

*Case 3.* As $(1+t)^e \le \beta < 1$,

$$\begin{aligned}\lambda[\max\{\beta,1\} - \max\{\beta,(1+t)^e\}] &= \lambda[1-\beta],\\ \max\{\beta,1\} - \max\{\beta,(1+\lambda t)^e\} &= 1-\beta.\end{aligned} \tag{9}$$

The equality holds because $\lambda \ge 1$.

*Case 4.* As $\beta \ge 1$,

$$\begin{aligned}\lambda[\max\{\beta,1\} - \max\{\beta,(1+t)^e\}] &= \lambda[\beta-\beta] = 0,\\ \max\{\beta,1\} - \max\{\beta,(1+\lambda t)^e\} &= \beta-\beta = 0.\end{aligned} \tag{10}$$

In summary, the inequality always holds for the given conditions.

To show $\max\{\beta,1\} - \max\{\beta,(1+\lambda t)^e\} = \beta-\beta = 0$ is no less than $S_2$, it suffices to show $C_{3j}(S_1) \le C_{3i}(S_2)$.

*Property 1.* If $a_i < a_j, b_i < b_j, c_i > c_j$, $t_3 > \max\{t_1 + a_j \max\{\alpha, (1+\sum_{i=1}^{k-1} a_{[i]})^{e_1}, t_2 + b_j \max\{\alpha, (1+\sum_{i=1}^{k-1} b_{[i]})^{e_2}\}\}$, and $t_1 + a_i \max\{\alpha, (1+\sum_{i=1}^{k-1} a_{[i]})^{e_1}\} + a_j \max\{\alpha, (1+\sum_{i=1}^{k-1} a_{[i]} + a_i)^{e_1}\} > \max\{t_2 + b_i \max\{\alpha, (1+\sum_{i=1}^{k-1} b_{[i]})^{e_2}\} + b_j \max\{\alpha, (1+\sum_{i=1}^{k-1} b_{[i]} + b_i)^{e_2}\}, t_3 + c_i \max\{\alpha, (1+\sum_{i=1}^{k-1} c_{[i]})^{e_3}\}\}$, then $S_1$ is no less than $S_2$.

*Proof.* For a job sequence $S$, if the job $J_j$ is assigned to the $r^{\text{th}}$ position of $S$, then according to the definition of the completion time of a job,

$$C_{3[r]}(S) = \max\left\{t_1 + a_j \max\left\{\alpha, \left(1+\sum_{i=1}^{r-1} a_{[i]}\right)^{e_1}\right\}, t_2 + b_j \max\left\{\alpha, \left(1+\sum_{i=1}^{r-1} b_{[i]}\right)^{e_2}\right\}, t_3\right\} + c_j \max\left\{\alpha, \left(1+\sum_{i=1}^{r-1} c_{[i]}\right)^{e_3}\right\}. \tag{11}$$

To simplify the proof, let $TA = (1+\sum_{i=1}^{k-1} a_{[i]})^{e_1}$, $TB = (1+\sum_{i=1}^{k-1} b_{[i]})^{e_2}$, and $TC = (1+\sum_{i=1}^{k-1} c_{[i]})^{e_3}$. Then,

$$\begin{aligned}C_{3j}(S_1) = \max\{&t_1 + a_i \max\{\alpha, (TA)^{e_1}\} + a_j \max\{\alpha, (TA+a_i)^{e_1}\}, t_2 + b_i \max\{\alpha, (TB)^{e_2}\} + b_j \max\{\alpha, (TB+b_i)^{e_2}\},\\ &\max\{t_3, t_1 + a_i \max\{\alpha, (TA)^{e_1}\}, b_i \max\{\alpha, (TB)^{e_2}\}\} + c_i \max\{\alpha, (TC)^{e_3}\}\} + c_j \max\{\alpha, (TC+c_i)^{e_3}\},\\ C_{3i}(S_2) = \max\{&t_1 + a_j \max\{\alpha, (TA)^{e_1}\} + a_i \max\{\alpha, (TA+a_j)^{e_1}\}, t_2 + b_j \max\{\alpha, (TB)^{e_2}\} + b_i \max\{\alpha, (TB+b_j)^{e_2}\},\\ &\max\{t_3, t_1 + a_j \max\{\alpha, (TA)^{e_1}\}, b_j \max\{\alpha, (TB)^{e_2}\}\} + c_j \max\{\alpha, (TC)^{e_3}\}\} + c_i \max\{\alpha, (TC+c_j)^{e_3}\}.\end{aligned} \tag{12}$$

Using the given conditions $a_i < a_j, b_i < b_j$, and $t_3 > \max\{t_1 + a_j \max\{\alpha, (TA)^{e_1}, t_2 + b_j \max\{\alpha, (TB)^{e_2}\}\}\}$, one has

$$\begin{aligned}C_{3j}(S_1) = \max\{&t_1 + a_i \max\{\alpha, (TA)^{e_1}\} + a_j \max\{\alpha, (TA+a_i)^{e_1}\}, t_2 + b_i \max\{\alpha, (TB)^{e_2}\} + b_j \max\{\alpha, (TB+b_i)^{e_2}\},\\ &t_3 + c_i \max\{\alpha, (TC)^{e_3}\}\} + c_j \max\{\alpha, (TC+c_i)^{e_3}\},\\ C_{3i}(S_2) = \max\{&t_1 + a_j \max\{\alpha, (TA)^{e_1}\} + a_i \max\{\alpha, (TA+a_j)^{e_1}\}, t_2 + b_j \max\{\alpha, (TB)^{e_2}\} + b_i \max\{\alpha, (TB+b_j)^{e_2}\},\\ &t_3 + c_j \max\{\alpha, (TC)^{e_3}\}\} + c_i \max\{\alpha, (TC+c_j)^{e_3}\}.\end{aligned} \tag{13}$$

Applying the given condition,

$$t_1 + a_i \max\{\alpha,\ (TA)^{e_1}\} + a_j \max\{\alpha,\ (TA + a_i)^{e_1}\} > \max\Big\{t_2 + b_i \max\{\alpha,\ (TB)^{e_2}\} + b_j \max\{\alpha,\ (TB + b_i)^{e_2}\},\ t_3 + c_i \max\{\alpha,\ (TC)^{e_3}\}\Big\}, \tag{14}$$

and then,

$$C_{3j}(S_1) = t_1 + a_i \max\{\alpha,\ (TA)^{e_1}\} + a_j \max\{\alpha,\ (TA + a_i)^{e_1}\} + c_i \max\Big\{\alpha, \big(TC + c_j\big)^{e_3}\Big\}, \tag{15}$$

$$C_{3i}(S_2) = t_1 + a_j \max\{\alpha,\ (TA)^{e_1}\} + a_i \max\Big\{\alpha, \big(TA + a_j\big)^{e_1}\Big\} + c_i \max\Big\{\alpha, \big(TC + c_j\big)^{e_3}\Big\}. \tag{16}$$

Equation (16) holds by showing $t_1 + a_j \max\{\alpha, (TA)^{e_1}\} + a_i \max\Big\{\alpha, (TA + a_j)^{e_1}\Big\} > [t_1 + a_i \max\{\alpha, (TA)^{e_1}\} + a_j \max\{\alpha, (TA + a_i)^{e_1}\}]$.

$$\begin{aligned}
&a_j \max\{\alpha,\ (TA)^{e_1}\} + a_i \max\Big\{\alpha,\ \big(TA + a_j\big)^{e_1}\Big\} - a_i \max\{\alpha,\ (TA)^{e_1}\} - a_j \max\{\alpha,\ (TA + a_i)^{e_1}\} \\
&= a_j\big[\max\{\alpha,\ (TA)^{e_1}\} - \max\{\alpha,\ (TA + a_i)^{e_1}\}\big] - a_i\big[\max\{\alpha,\ (TA)^{e_1}\} - \max\{\alpha,\ (TA + a_i)^{e_1}\}\big], \\
&= a_i (TA)^{e_1} \cdot \left\{\frac{a_j}{a_i}\left[\max\left\{\frac{\alpha}{(TA)^{e_1}}, 1\right\} - \max\left\{\frac{\alpha}{(TA)^{e_1}}, \left(1 + \frac{a_i}{(TA)^{e_1}}\right)^{e_1}\right\}\right] - \left[\max\left\{\frac{\alpha}{(TA)^{e_1}}, 1\right\}\right.\right. \\
&\quad \left.\left. -\max\left\{\frac{\alpha}{(TA)^{e_1}}, \left(1 + \frac{a_j}{a_i}\frac{a_i}{(TA)^{e_1}}\right)^{e_1}\right\}\right]\right\}, \\
&= a_i (TA)^{e_1} \cdot \{\lambda\big[\max\{\beta, 1\} - \max\{\beta,\ (1 + t)^{e_1}\}\big] - \big[\max\{\beta, 1\} - \max\{\beta, (1 + \lambda t)^{e_1}\}\big]\},
\end{aligned} \tag{17}$$

where $\lambda = (a_j/a_i)\ (>1)$, $\beta = (\alpha/(TA)^{e_1})\ (>0)$, and $t = (a_i/(TA)^{e_1})\ (>0)$. Applying Lemma 4, $t_1 + a_j \max\{\alpha,\ (TA)^{e_1}\} + a_i \max\Big\{\alpha,\ (TA + a_j)^{e_1}\Big\} > [t_1 + a_i \max\{\alpha,\ (TA)^{e_1}\} + a_j \max\{\alpha,\ (TA + a_i)^{e_1}\}]$ follows and, thus, completes the claim.

Finally, one needs to show that $C_{3i}(S_2) - C_{3j}(S_1) \geq 0$ to fulfill the proof. From equations (15) and (16), one obtains, applying $a_i < a_j, c_i > c_j$ and the fact $(TA + a_j)^{e_1} < (TA + a_i)^{e_1}$ and $(TC + c_j)^{e_3} > (TC + c_i)^{e_3}$, that all three terms of $C_{3i}(S_2) - C_{3j}(S_1)$ are greater than or equal to 0, i.e., $C_{3i}(S_2) - C_{3j}(S_1) \geq 0$.

The details of the proofs of Property 2–4 are omitted because they can be obtained in a similar way to Property 1. □

*Property 2.* If $a_i < a_j, b_i < b_j, c_i > c_j$,

$t_3 > \max\Big\{t_1 + a_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1},\ t_2 + b_j \max\{\alpha, (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\}\Big\}\Big\}$, and $t_2 + b_i \max\{\alpha,\ (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\} + b_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} b_{[i]} + b_i)^{a_2}\Big\} > \max\{t_1 + a_i \max\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1}\} + a_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]} + a_i)^{a_1}\Big\}, t_3 + c_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} c_{[i]})^{a_3}\Big\}\}$, then $S_1$ is no less than $S_2$.

*Property 3.* If $a_i < a_j, b_i < b_j, c_i > c_j$,

$t_1 + a_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1}\Big\} > \max\Big\{t_3, t_2 + b_j \max\Big\{\alpha, (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\Big\}\Big\}$, and $a_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1}\Big\} + c_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} c_{[i]})^{a_3}\Big\} > a_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1}\Big\} + c_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} c_{[i]})^{a_3}\Big\}$, then $S_1$ is no less than $S_2$.

*Property 4.* If $a_i < a_j, b_i < b_j, c_i > c_j$,

$t_2 + b_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\Big\} > \max\Big\{t_3, t_1 + a_j \max\{\alpha, (1 + \sum_{i=1}^{k-1} a_{[i]})^{a_1}\}\Big\}$, and $b_j \max\{\alpha,\ (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\} + c_j \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} c_{[i]})^{a_3}\Big\} > b_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} b_{[i]})^{a_2}\Big\} + c_i \max\Big\{\alpha,\ (1 + \sum_{i=1}^{k-1} c_{[i]})^{a_3}\Big\}$, then $S_1$ is no less than $S_2$.

## 4. Heuristics and Metaheuristics

This section validates the performance levels of nine algorithms, namely, JS_max, JS_min, JS_mean, JS_maxpi, JS_minpi, JS_meanpi, SA_max, SA_min, and SA_mean. Note that all algorithms were developed in FORTRAN and executed on a computer with a 3.00-GHz Intel® Core™i9-10980XE CPU and 16-GB RAM. Our experiments are divided into two parts. The first part is dedicated to the small size instances, and the second is devoted to the large size instances. For the two cases, we randomly generated the normal processing times from uniform distribution U (1, 100) on M1, M2, and M3 by fixing an individual seed for each case. The factor $\beta$ would be set at 0.2, 0.4, and 0.6. The learning effect would be set at −0.1, −0.1, and −0.2, and the computational results for each small and large size instances are shown. This section describes the considered two-stage assembly scheduling problem.

*4.1. Small Size Number of Jobs.* The parameters of the small size number of jobs are selected as the number of work pieces ($n$) at 10, number of machines ($m$) of 3, and the final temperature (Tf) is fixed to $10^{-8}$, $\beta$ is 0.2, $\alpha_1$ is −0.1, $\alpha_2$ is −0.1, and $\alpha_3$ is −0.2. The selections of other parameters are shown in Table 1.

As a result of parameter selection, the starting temperature ($T_i$) is $10^{-3}$, cooling coefficient is 0.95, and the number of tests ($N_r$) of Johnson's algorithm is 10.

Next, the parameter selection process is explained in detail.

We select the parameter starting temperature ($T_i$). The range is from $10^{-4}$ to $10^{4}$ at 10 times the interval, and the cooling coefficient ($c_f$) is fixed at 0.95. The number of tests of Johnson's algorithm ($N_r$) is 5, and there are also generated data and parameters defined before selecting parameters.

By observing Figure 1, it is obvious that only the initial temperature ($T_i$) is changed. In this range, it can be observed that the AEP is relatively low at $10^{-3}$, so the initial temperature ($T_i$) is $10^{-3}$.

After fixing the starting temperature ($T_i$) to $10^{-3}$, we select the parameter cooling coefficient ($c_f$), ranging from 0.05 to 0.95 with 0.05 intervals, and fix the number of tests ($N_r$) of the Johnson's algorithm as 5. There are also generated data and parameters defined before selecting the parameters.

By selecting the parameter of the cooling coefficient ($c_f$), the overall decline in its value is obvious. It becomes flat after reaching 0.80, so we choose a lower value in this part. The cooling coefficient ($c_f$) is fixed at 0.95. After fixing the starting temperature ($T_i$) to $10^{-3}$ and the cooling coefficient (Cf) to 0.95 (please see Figure 2), we select the parameter Johnson's algorithm test times ($N_r$) with a range of 1 to 20 and the interval of 1 and see the difference. The generated data and parameters are fixed and defined before selecting the parameters.

When selecting the parameter Johnson's algorithm test times ($N_r$), we can see from Figure 3 that although it fluctuates up and down, its AEP value is actually very low, so we choose a lower value in this part. Johnson's algorithm test times ($N_r$) is 10. It shows when the number of work pieces of the B&B algorithm is 12. The results of finding the best solution in a few cases cannot be found, so only a few values (in the Fs column) are recorded. From the viewpoint of the number of nodes and CPU time, the difference in the number of jobs can also be clearly seen.

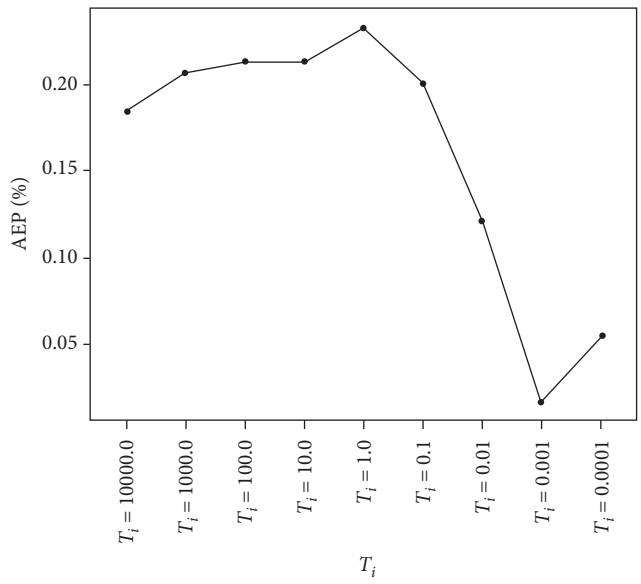



Figure 1: Parameter selection starting temperature ($T_i$) change.

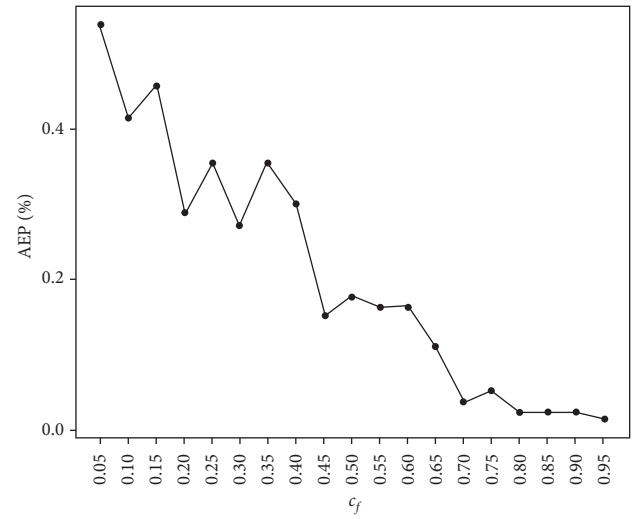



Figure 2: Parameter selection cooling coefficient ($c_f$) change.

Table 2 uses the $\beta$ values of 0.2, 0.4, and 0.6 to distinguish the table. It can be seen regardless of whether the number of work pieces is 8, 10, or 12, the smaller the $\beta$ value, the greater the number of nodes. Also, there is no best solution for the CPU time.

Table 3 is a table distinguished by three different situations of $\alpha_1, \alpha_2, \alpha_3$. It can be observed that regardless of whether the number of work pieces is 8, 10, or 12, the

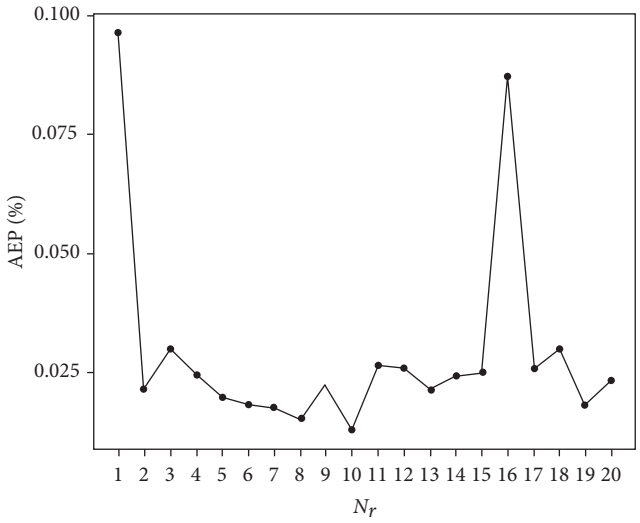



Figure 3: Parameter selection of Johnson's algorithm test times ($N_r$) change.

Table 2: Different $\beta$ for the mean of nodes and CPU time.

| Parameter | | Node | CPU time |
|---|---|---|---|
| $N$ | $\beta$ | Mean | Mean |
| 8 | 0.2 | 9871 | 0.11 |
| | 0.4 | 4474 | 0.05 |
| | 0.6 | 2825 | 0.03 |
| 10 | 0.2 | 751491 | 10.33 |
| | 0.4 | 262445 | 3.89 |
| | 0.6 | 196723 | 3.13 |
| 12 | 0.2 | 51984213 | 978.76 |
| | 0.4 | 27751022 | 585.76 |
| | 0.6 | 16943700 | 327.17 |

Table 3: Different $\beta$ for the mean of nodes and CPU time.

| $n$ | Parameter | | | Node | CPU time |
|---|---|---|---|---|---|
| | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | Mean | Mean |
| 8 | −0.1 | −0.1 | −0.1 | 4448 | 0.06 |
| | −0.1 | −0.1 | −0.2 | 7860 | 0.08 |
| | −0.2 | −0.2 | −0.1 | 4861 | 0.05 |
| 10 | −0.1 | −0.1 | −0.1 | 281133 | 3.71 |
| | −0.1 | −0.1 | −0.2 | 608118 | 8.73 |
| | −0.2 | −0.2 | −0.1 | 321109 | 4.91 |
| 12 | −0.1 | −0.1 | −0.1 | 22327360 | 467.04 |
| | −0.1 | −0.1 | −0.2 | 41700559 | 775.61 |
| | −0.2 | −0.2 | −0.1 | 32651016 | 649.04 |

number of nodes and CPU time for finding the best solution for the second case are all more than the third case, and the third case is more than the first case.

The simulation result for the small size number of jobs from the total mean at the bottom shows that the SA-related algorithm performs better in the small size number of jobs.

Table 4 has $\beta$ values of 0.2, 0.4, and 0.6, and no obvious features are noted.

Table 5 is divided into three different situations of $\alpha_1, \alpha_2, \alpha_3$. Clearly, in the case of various numbers of jobs, the AEP of the second case is better than the AEP of the first case. The AEP for each case is better than the AEP for the third case.

Table 6 compares nine algorithms in total. There are three changes in the number of jobs ($n$) and $\beta$ and 2 changes in $\alpha_1, \alpha_2, \alpha_3$.

Before comparing algorithms, normality verification is carried out on the data. If the data is normal, you can use the average of the algorithm to compare pairwise.

The residuals are not in line with the normal state, so they cannot be compared using means. We instead use the Kruskal–Wallis test to observe whether the medians of the data are the same (please see Table 7). If they are not the same, we can use the median of the algorithm to make pairwise comparisons and observe the calculations for the difference in the median between methods.

The null hypothesis is that the median of each group is the same, where the $P$ value <0.05, so the null hypothesis is rejected, meaning the median of each group is not the same and multiple comparative analysis can be performed to show the difference between each group (please see Table 8).

Figure 4 is a box diagram of the nine algorithms scored by nonparametric statistics. Clearly, the results of the SA-related algorithms under the small size number of jobs are superior.

Table 9 shows the results of the pairwise comparison. We take the significance level of 0.05 as the standard. The JS_max algorithm and the JS_mean algorithm are grouped into one group, the JS_mean algorithm is divided into a group with the JS_maxpi algorithm, and the JS_maxpi algorithm is divided into a group with the JS_meanpi algorithm. Finally, the SA-related algorithms are all divided into a group. From this, it can be seen that the SA algorithm is a relatively better group for the small size number of jobs, followed by the JS_meanpi Algorithm, JS_maxpi algorithm, JS_mean algorithm, JS_max algorithm, and finally, the JS_minpi algorithm and JS_min algorithm. There are two key points clearly observed in the grouping. First, the JS algorithm with pi is better. Second, the result obtained by mean is better than max, and max is better than min.

*4.2. Large Size Number of Jobs.* The parameters of the large size number of jobs are selected when the number of work pieces ($n$) is 10, the number of machines ($m$) is 3, and the final temperature ($T_f$) is fixed to $10^{-8}$, $\beta$ is 0.2, $\alpha_1$ is −0.1, $\alpha_2$ is −0.1, and $\alpha_3$ is −0.2. The selections of other parameters are shown in Table 10.

From the summary of Tables 11 and 12, the SA algorithm is the best in calculating the RPD. However, the CPU_time shows that the SA algorithm takes a relatively long time. With pi or without pi, the JS algorithm clearly shows that the solution with pi to the run out of RPD is slightly inferior to the solution without pi. The JS algorithm with pi for the CPU_time is the fastest.

Table 13 shows the residual error in the large size number of jobs is not in line with the normal state, so we also

Table 4: Simulation result for $\beta$ values.

| Parameter | | JS_max | JS_min | JS_mean | JS_maxpi | JS_minpi | JS_meanpi | SA_max | SA_min | SA_mean |
|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $\beta$ | Mean | Mean | Mean | Mean | Mean | Mean | Mean | Mean | Mean |
| | 0.2 | 2.92 | 6.69 | 1.96 | 1.81 | 4.43 | 1.15 | 0.10 | 0.09 | 0.08 |
| 8 | 0.4 | 2.87 | 7.77 | 2.50 | 1.78 | 5.51 | 1.61 | 0.07 | 0.09 | 0.05 |
| | 0.6 | 2.25 | 7.21 | 2.01 | 1.43 | 5.07 | 1.29 | 0.07 | 0.08 | 0.06 |
| | 0.2 | 1.95 | 5.94 | 1.67 | 1.21 | 4.18 | 1.12 | 0.05 | 0.06 | 0.05 |
| 10 | 0.4 | 2.09 | 6.70 | 1.60 | 1.33 | 4.95 | 1.01 | 0.06 | 0.06 | 0.06 |
| | 0.6 | 1.59 | 6.11 | 1.38 | 1.09 | 4.40 | 0.93 | 0.07 | 0.10 | 0.08 |
| | 0.2 | 1.70 | 5.99 | 1.36 | 1.04 | 4.75 | 0.88 | 0.11 | 0.08 | 0.10 |
| 12 | 0.4 | 1.59 | 5.93 | 1.40 | 0.97 | 4.59 | 0.93 | 0.05 | 0.09 | 0.08 |
| | 0.6 | 1.32 | 6.24 | 1.26 | 0.92 | 4.88 | 0.95 | 0.10 | 0.09 | 0.10 |
| Mean | | 2.03 | 6.51 | 1.68 | 1.28 | 4.75 | 1.10 | 0.08 | 0.08 | 0.07 |

Table 5: Simulation result for $\alpha$ values.

| | Parameter | | | JS_max | JS_min | JS_mean | JS_maxpi | JS_minpi | JS_meanpi | SA_max | SA_min | SA_mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | Mean | Mean | Mean | Mean | Mean | Mean | Mean | Mean | Mean |
| | −0.1 | −0.1 | −0.1 | 2.10 | 7.14 | 1.72 | 1.26 | 5.15 | 1.07 | 0.06 | 0.08 | 0.05 |
| 8 | −0.1 | −0.1 | −0.2 | 1.89 | 5.52 | 1.80 | 1.32 | 3.82 | 1.11 | 0.05 | 0.07 | 0.04 |
| | −0.2 | −0.2 | −0.1 | 4.04 | 9.01 | 2.95 | 2.43 | 6.04 | 1.88 | 0.13 | 0.11 | 0.10 |
| | −0.1 | −0.1 | −0.1 | 1.57 | 6.13 | 1.42 | 0.98 | 4.53 | 0.98 | 0.05 | 0.06 | 0.06 |
| 10 | −0.1 | −0.1 | −0.2 | 1.48 | 4.52 | 1.37 | 1.04 | 3.20 | 0.93 | 0.06 | 0.06 | 0.07 |
| | −0.2 | −0.2 | −0.1 | 2.58 | 8.10 | 1.87 | 1.60 | 5.79 | 1.15 | 0.07 | 0.10 | 0.06 |
| | −0.1 | −0.1 | −0.1 | 1.37 | 6.41 | 1.23 | 0.90 | 4.91 | 0.85 | 0.11 | 0.14 | 0.10 |
| 12 | −0.1 | −0.1 | −0.2 | 1.09 | 4.32 | 0.99 | 0.79 | 3.55 | 0.76 | 0.03 | 0.02 | 0.02 |
| | −0.2 | −0.2 | −0.1 | 2.15 | 7.43 | 1.80 | 1.24 | 5.76 | 1.15 | 0.12 | 0.09 | 0.15 |
| Mean | | | | 2.03 | 6.51 | 1.68 | 1.28 | 4.75 | 1.10 | 0.08 | 0.08 | 0.07 |

Table 6: Small size number of jobs analysis variable.

| Class level information | | |
|---|---|---|
| Class | Levels | Values |
| Algorithm | 9 | JS_max, JS_maxpi, JS_mean, JS_meanpi, JS_min, JS_minpi, SA_max, SA_mean, and SA_min |
| $N$ | 3 | 8 10 12 |
| $\beta$ | 3 | 0.2 0.4 0.6 |
| $\alpha_1$ | 2 | −0.1 −0.2 |
| $\alpha_2$ | 2 | −0.1 −0.2 |
| $\alpha_3$ | 2 | −0.1 −0.2 |

Table 7: Normality test of residual error of the GLM model in the small size number of jobs.

| Normality test | | | | |
|---|---|---|---|---|
| Test | Statistics | | $P$ value | |
| Shapiro–Wilk | $W$ | 0.936652 | Pr < $W$ | <0.0001 |
| Kolmogorov–Smirnov | $D$ | 0.086882 | Pr < $D$ | <0.0100 |
| Cramer–von Mises | $W$-Sq | 0.532874 | Pr < $W$-Sq | <0.0050 |
| Anderson–Darling | $A$-Sq | 3.734124 | PR < $A$-Sq | <0.0050 |

Table 8: Kruskal–Wallis verification of the small size number of jobs.

| Kruskal–Wallis test | | |
|---|---|---|
| Chi-square | DF | Pr > ChiSq |
| 214.5511 | 8 | <0.0001 |

use the same method as the small size number of jobs to compare the relationship between the algorithms.

In the Kruskal–Wallis test, the null hypothesis is rejected as with the small size number of jobs, meaning the median of each group is not the same and multiple comparison analysis

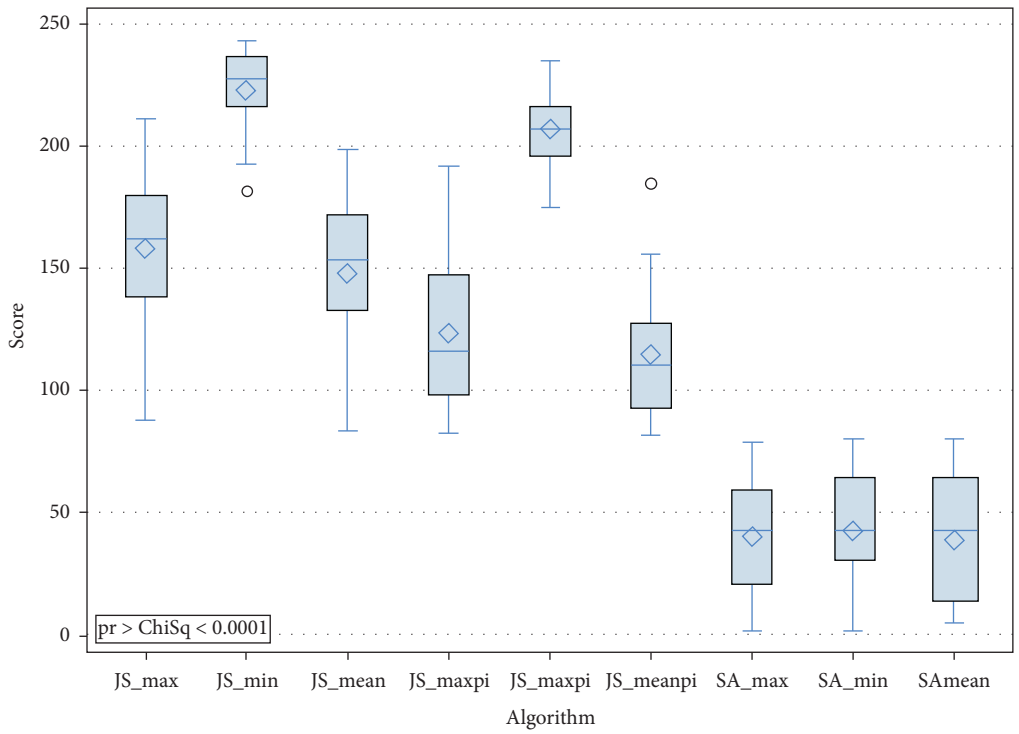



Figure 4: Scoring box of the nine algorithms.

Table 9: Pairwise comparison of nine algorithms for the small size number of jobs.

| | Small $n$ | |
|---|---|---|
| Algorithm | Statistic | $P$ value |
| JS_max vs. JS_min | 8.5999 | <0.0001 |
| JS_max vs. JS_mean | 1.8843 | 0.9218 |
| JS_max vs. JS_maxpi | 5.3095 | 0.0054 |
| JS_max vs. JS_minpi | 7.7925 | <0.0001 |
| JS_max vs. JS_meanpi | 6.9494 | <0.0001 |
| JS_max vs. SA_max | 8.9241 | <0.0001 |
| JS_max vs. SA_min | 8.9268 | <0.0001 |
| JS_max vs. SA_mean | 8.9231 | <0.0001 |
| JS_min vs. JS_mean | 8.7717 | <0.0001 |
| JS_min vs. JS_maxpi | 8.8692 | <0.0001 |
| JS_min vs. JS_minpi | 5.0645 | 0.0103 |
| JS_min vs. JS_meanpi | 8.8938 | <0.0001 |
| JS_min vs. SA_max | 8.9239 | <0.0001 |
| JS_min vs. SA_min | 8.9267 | <0.0001 |
| JS_min vs. SA_mean | 8.9229 | <0.0001 |
| JS_mean vs. JS_maxpi | 4.3435 | 0.0547 |
| JS_mean vs. JS_minpi | 8.6249 | <0.0001 |
| JS_mean vs. JS_meanpi | 5.9589 | 0.0008 |
| JS_mean vs. SA_max | 8.9246 | <0.0001 |
| JS_mean vs. SA_min | 8.9273 | <0.0001 |
| JS_mean vs. SA_mean | 8.9236 | <0.0001 |
| JS_maxpi vs. JS_minpi | 8.7224 | <0.0001 |
| JS_maxpi vs. JS_meanpi | 1.7866 | 0.9418 |
| JS_maxpi vs. SA_max | 8.9243 | <0.0001 |
| JS_maxpi vs. SA_min | 8.9270 | <0.0001 |
| JS_maxpi vs. SA_mean | 8.9233 | <0.0001 |
| JS_minpi vs. JS_meanpi | 8.8449 | <0.0001 |
| JS_minpi vs. SA_max | 8.9239 | <0.0001 |
| JS_minpi vs. SA_min | 8.9267 | <0.0001 |
| JS_minpi vs. SA_mean | 8.9229 | <0.0001 |
| JS_meanpi vs. SA_max | 8.9244 | <0.0001 |
| JS_meanpi vs. SA_min | 8.9272 | <0.0001 |
| JS_meanpi vs. SA_mean | 8.9234 | <0.0001 |
| SA_max vs. SA_min | 0.6278 | 1.0000 |
| SA_max vs. SA_mean | 0.4790 | 1.0000 |
| SA_min vs. SA_mean | 1.1795 | 0.9959 |

Table 10: Parameter selection of the large size number of jobs.

| Parameter | $n = 90$, $m = 3$ |
|---|---|
| $T_i$ | $10^{-3}$ |
| $T_f$ | $10^{-8}$ |
| $c_f$ | 0.95 |
| $N_r$ | 90 |
| $\beta$ | 0.2 |
| $\alpha_1$ | −0.1 |
| $\alpha_2$ | −0.1 |
| $\alpha_3$ | −0.2 |

can be performed to show the differences between each group (please see Table 14).

Figure 5 is a box diagram of nine algorithms scored by nonparametric statistics. The figure shows clear differences among SAs and others under the large size number of jobs.

Table 15 shows the results of the pairwise comparison. If the $P$ value value is greater than 0.05, it means the effect between the pair is not significant, but they are correlative. In

Table 11: RPD summary of the large size number of jobs.

| Parameter | | | | | JS_max | | JS_min | | JS_mean | | JS_maxpi | | JS_minpi | | JS_meanpi | | SA_max | | SA_min | | SA_mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | $\beta$ | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max |
| | | −0.1 | −0.1 | −0.1 | 0.22 | 1.61 | 2.07 | 7.92 | 0.24 | 1.76 | 0.26 | 1.70 | 2.19 | 7.92 | 0.26 | 1.76 | 0.11 | 1.56 | 0.10 | 1.45 | 0.10 | 1.00 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.17 | 0.97 | 1.73 | 6.50 | 0.21 | 2.31 | 0.21 | 1.04 | 1.83 | 7.78 | 0.23 | 2.33 | 0.08 | 0.77 | 0.10 | 1.35 | 0.06 | 0.80 |
| | | −0.2 | −0.2 | -0.1 | 0.20 | 1.73 | 1.74 | 6.41 | 0.22 | 1.91 | 0.24 | 1.97 | 1.88 | 6.50 | 0.25 | 1.94 | 0.06 | 0.60 | 0.09 | 0.83 | 0.07 | 0.68 |
| | | −0.1 | −0.1 | −0.1 | 0.19 | 1.13 | 1.84 | 7.15 | 0.22 | 1.47 | 0.23 | 1.20 | 1.95 | 7.40 | 0.25 | 1.48 | 0.08 | 1.11 | 0.06 | 1.06 | 0.05 | 0.28 |
| 60 | 0.4 | −0.1 | −0.1 | −0.2 | 0.18 | 1.11 | 1.68 | 6.18 | 0.18 | 1.42 | 0.22 | 1.14 | 1.80 | 7.11 | 0.21 | 1.44 | 0.07 | 0.67 | 0.07 | 1.91 | 0.07 | 0.65 |
| | | −0.2 | −0.2 | −0.1 | 0.15 | 0.96 | 1.83 | 6.74 | 0.16 | 1.49 | 0.19 | 1.02 | 1.95 | 7.02 | 0.19 | 1.53 | 0.07 | 0.64 | 0.08 | 1.44 | 0.07 | 1.17 |
| | | −0.1 | −0.1 | −0.1 | 0.16 | 1.34 | 2.01 | 6.79 | 0.19 | 1.36 | 0.20 | 1.38 | 2.15 | 7.30 | 0.21 | 1.39 | 0.07 | 0.53 | 0.12 | 1.90 | 0.05 | 0.34 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.18 | 1.40 | 2.09 | 6.77 | 0.20 | 1.49 | 0.23 | 1.50 | 2.22 | 8.56 | 0.22 | 1.50 | 0.09 | 0.68 | 0.11 | 1.84 | 0.06 | 0.58 |
| | | −0.2 | −0.2 | −0.1 | 0.16 | 0.95 | 1.99 | 7.17 | 0.19 | 1.18 | 0.20 | 1.02 | 2.12 | 11.54 | 0.21 | 1.19 | 0.08 | 0.51 | 0.06 | 1.06 | 0.07 | 0.57 |
| Mean | | | | | 0.18 | | 1.89 | | 0.20 | | 0.22 | | 2.01 | | 0.23 | | 0.08 | | 0.09 | | 0.07 | |
| | | −0.1 | −0.1 | −0.1 | 0.13 | 0.97 | 1.48 | 6.35 | 0.13 | 0.99 | 0.15 | 1.03 | 1.53 | 6.77 | 0.14 | 1.02 | 0.06 | 0.67 | 0.08 | 1.00 | 0.05 | 0.41 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.12 | 1.02 | 1.10 | 5.01 | 0.12 | 1.03 | 0.14 | 1.04 | 1.15 | 5.02 | 0.14 | 1.04 | 0.06 | 0.46 | 0.07 | 0.98 | 0.07 | 0.77 |
| | | −0.2 | −0.2 | −0.1 | 0.14 | 1.01 | 1.40 | 6.47 | 0.16 | 1.18 | 0.17 | 1.09 | 1.44 | 6.48 | 0.18 | 1.21 | 0.08 | 0.68 | 0.10 | 1.61 | 0.07 | 0.63 |
| | | −0.1 | −0.1 | −0.1 | 0.14 | 0.88 | 1.39 | 4.97 | 0.15 | 0.87 | 0.16 | 0.92 | 1.45 | 5.01 | 0.17 | 0.88 | 0.07 | 0.71 | 0.08 | 1.01 | 0.08 | 0.78 |
| 90 | 0.4 | −0.1 | −0.1 | −0.2 | 0.14 | 0.96 | 1.35 | 5.11 | 0.14 | 1.03 | 0.16 | 0.98 | 1.40 | 5.34 | 0.15 | 1.07 | 0.07 | 0.88 | 0.07 | 0.86 | 0.05 | 0.17 |
| | | −0.2 | −0.2 | −0.1 | 0.11 | 0.89 | 1.51 | 5.61 | 0.11 | 0.85 | 0.13 | 0.90 | 1.56 | 5.61 | 0.13 | 0.87 | 0.05 | 0.47 | 0.07 | 0.89 | 0.05 | 0.66 |
| | | −0.1 | −0.1 | −0.1 | 0.10 | 0.66 | 1.41 | 4.87 | 0.12 | 0.97 | 0.13 | 0.73 | 1.46 | 4.87 | 0.14 | 0.99 | 0.06 | 0.40 | 0.10 | 0.69 | 0.06 | 0.61 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.16 | 0.98 | 1.52 | 4.88 | 0.17 | 1.22 | 0.18 | 0.99 | 1.56 | 4.89 | 0.18 | 1.24 | 0.06 | 0.65 | 0.08 | 0.87 | 0.08 | 0.93 |
| | | −0.2 | −0.2 | −0.1 | 0.14 | 0.87 | 1.44 | 5.23 | 0.17 | 1.26 | 0.17 | 0.88 | 1.50 | 5.52 | 0.19 | 1.30 | 0.07 | 0.76 | 0.09 | 0.90 | 0.08 | 0.78 |
| Mean | | | | | 0.13 | | 1.40 | | 0.14 | | 0.15 | | 1.45 | | 0.16 | | 0.07 | | 0.08 | | 0.07 | |
| | | −0.1 | −0.1 | −0.1 | 0.11 | 0.60 | 1.09 | 3.84 | 0.12 | 0.58 | 0.12 | 0.62 | 1.12 | 3.95 | 0.13 | 0.63 | 0.06 | 0.50 | 0.09 | 0.88 | 0.06 | 0.56 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.10 | 0.70 | 1.12 | 3.58 | 0.10 | 0.75 | 0.11 | 0.71 | 1.16 | 3.71 | 0.11 | 0.76 | 0.04 | 0.51 | 0.10 | 0.61 | 0.05 | 0.58 |
| | | −0.2 | −0.2 | −0.1 | 0.07 | 0.49 | 1.14 | 3.66 | 0.08 | 0.50 | 0.09 | 0.53 | 1.18 | 4.27 | 0.09 | 0.51 | 0.05 | 0.48 | 0.07 | 0.76 | 0.05 | 0.49 |
| | | −0.1 | −0.1 | −0.1 | 0.07 | 0.61 | 1.11 | 4.31 | 0.08 | 0.58 | 0.09 | 0.65 | 1.15 | 4.40 | 0.09 | 0.59 | 0.04 | 0.39 | 0.07 | 0.85 | 0.05 | 0.40 |
| 120 | 0.4 | −0.1 | −0.1 | −0.2 | 0.09 | 0.71 | 1.15 | 4.41 | 0.10 | 1.06 | 0.11 | 0.79 | 1.18 | 4.41 | 0.11 | 1.08 | 0.06 | 0.76 | 0.09 | 0.88 | 0.05 | 0.42 |
| | | −0.2 | −0.2 | −0.1 | 0.12 | 0.70 | 1.13 | 3.86 | 0.14 | 0.93 | 0.13 | 0.73 | 1.17 | 3.87 | 0.15 | 0.94 | 0.07 | 0.67 | 0.10 | 1.01 | 0.07 | 0.68 |
| | | −0.1 | −0.1 | −0.1 | 0.08 | 0.61 | 1.01 | 3.61 | 0.10 | 0.88 | 0.10 | 0.62 | 1.06 | 3.61 | 0.11 | 0.88 | 0.05 | 0.47 | 0.08 | 1.18 | 0.05 | 0.25 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.09 | 0.71 | 1.04 | 3.82 | 0.10 | 0.67 | 0.11 | 0.74 | 1.06 | 3.83 | 0.11 | 0.68 | 0.04 | 0.64 | 0.10 | 0.75 | 0.04 | 0.23 |
| | | −0.2 | −0.2 | −0.1 | 0.09 | 0.64 | 1.15 | 3.54 | 0.10 | 0.57 | 0.11 | 0.67 | 1.18 | 3.55 | 0.11 | 0.59 | 0.06 | 0.62 | 0.08 | 0.61 | 0.05 | 0.39 |
| Mean | | | | | 0.09 | | 1.10 | | 0.10 | | 0.11 | | 1.14 | | 0.11 | | 0.05 | | 0.09 | | 0.05 | |
| Total mean | | | | | 0.13 | | 1.46 | | 0.15 | | 0.16 | | 1.53 | | 0.16 | | 0.07 | | 0.09 | | 0.06 | |

Table 12: CPU_time summary of the large size number of jobs.

| Parameter | | | | | JS_max | | JS_min | | JS_mean | | JS_maxpi | | JS_minpi | | JS_meanpi | | SA_max | | SA_min | | SA_mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\beta$ | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max | Mean | Max |
| | | −0.1 | −0.1 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.22 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.22 |
| | | −0.2 | −0.2 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.22 | 0.19 | 0.20 |
| | | −0.1 | −0.1 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.22 | 0.19 | 0.20 |
| 60 | 0.4 | −0.1 | −0.1 | −0.2 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.20 |
| | | −0.2 | −0.2 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.20 |
| | | −0.1 | −0.1 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.22 | 0.19 | 0.20 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.20 |
| | | −0.2 | −0.2 | −0.1 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.02 | 0.19 | 0.20 | 0.19 | 0.20 | 0.19 | 0.20 |
| Mean | | | | | 0.02 | | 0.02 | | 0.02 | | 0.00 | | 0.00 | | 0.00 | | 0.19 | | 0.19 | | 0.19 | |
| | | −0.1 | −0.1 | −0.1 | 0.06 | 0.08 | 0.06 | 0.08 | 0.06 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.28 | 0.30 | 0.28 | 0.30 | 0.28 | 0.30 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.06 | 0.06 | 0.06 | 0.08 | 0.06 | 0.06 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.00 | 0.28 | 0.30 | 0.29 | 0.31 | 0.28 | 0.30 |
| | | −0.2 | 0.2 | −0.1 | 0.06 | 0.08 | 0.06 | 0.06 | 0.06 | 0.08 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.28 | 0.30 | 0.28 | 0.30 | 0.28 | 0.30 |
| | | −0.1 | −0.1 | −0.1 | 0.06 | 0.06 | 0.06 | 0.08 | 0.06 | 0.06 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.00 | 0.28 | 0.30 | 0.29 | 0.31 | 0.28 | 0.30 |
| 90 | 0.4 | −0.1 | −0.1 | −0.2 | 0.06 | 0.06 | 0.06 | 0.08 | 0.06 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.28 | 0.30 | 0.28 | 0.30 | 0.28 | 0.30 |
| | | −0.2 | −0.2 | −0.1 | 0.06 | 0.06 | 0.06 | 0.08 | 0.06 | 0.08 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.28 | 0.30 | 0.29 | 0.31 | 0.28 | 0.30 |
| | | −0.1 | −0.1 | −0.1 | 0.06 | 0.08 | 0.06 | 0.06 | 0.06 | 0.06 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.02 | 0.28 | 0.30 | 0.28 | 0.30 | 0.28 | 0.31 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.08 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.02 | 0.28 | 0.30 | 0.28 | 0.31 | 0.28 | 0.30 |
| | | −0.2 | −0.2 | −0.1 | 0.06 | 0.08 | 0.06 | 0.08 | 0.06 | 0.08 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.28 | 0.30 | 0.28 | 0.33 | 0.28 | 0.30 |
| Mean | | | | | 0.06 | | 0.06 | | 0.06 | | 0.00 | | 0.00 | | 0.00 | | 0.28 | | 0.28 | | 0.28 | |
| | | −0.1 | −0.1 | −0.1 | 0.13 | 0.14 | 0.13 | 0.14 | 0.13 | 0.14 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 | 0.38 | 0.39 | 0.38 | 0.41 | 0.38 | 0.41 |
| | 0.2 | −0.1 | −0.1 | −0.2 | 0.13 | 0.14 | 0.13 | 0.14 | 0.13 | 0.14 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.38 | 0.39 | 0.38 | 0.41 | 0.38 | 0.39 |
| | | −0.2 | −0.2 | −0.1 | 0.13 | 0.14 | 0.13 | 0.14 | 0.13 | 0.14 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 | 0.38 | 0.41 | 0.38 | 0.39 | 0.38 | 0.39 |
| | | −0.1 | −0.1 | −0.1 | 0.13 | 0.14 | 0.13 | 0.14 | 0.13 | 0.14 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.38 | 0.39 | 0.38 | 0.41 | 0.38 | 0.41 |
| 120 | 0.4 | −0.1 | −0.1 | −0.2 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.02 | 0.39 | 0.41 | 0.39 | 0.42 | 0.39 | 0.42 |
| | | −0.2 | −0.2 | −0.1 | 0.14 | 0.16 | 0.14 | 0.14 | 0.14 | 0.14 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.02 | 0.39 | 0.41 | 0.39 | 0.42 | 0.39 | 0.41 |
| | | −0.1 | −0.1 | −0.1 | 0.14 | 0.16 | 0.14 | 0.14 | 0.14 | 0.14 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.02 | 0.39 | 0.41 | 0.39 | 0.39 | 0.39 | 0.41 |
| | 0.6 | −0.1 | −0.1 | −0.2 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.16 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.00 | 0.39 | 0.41 | 0.39 | 0.44 | 0.39 | 0.42 |
| | | −0.2 | −0.2 | −0.1 | 0.14 | 0.16 | 0.14 | 0.16 | 0.14 | 0.16 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.39 | 0.42 | 0.39 | 0.42 | 0.39 | 0.41 |
| Mean | | | | | 0.14 | | 0.13 | | 0.13 | | 0.00 | | 0.00 | | 0.00 | | 0.38 | | 0.38 | | 0.38 | |
| Total mean | | | | | 0.07 | | 0.07 | | 0.07 | | 0.00 | | 0.00 | | 0.00 | | 0.28 | | 0.29 | | 0.29 | |

Table 13: Normality test of residual error of the GLM model with the large size number of jobs.

| Normality test | | | | |
|---|---|---|---|---|
| Hypothesis test | Statistics | | *P* value | |
| Shapiro–Wilk | W | 0.916964 | Pr < W | <0.0001 |
| Kolmogorov–Smirnov | D | 0.120337 | Pr < D | <0.0100 |
| Cramer–von Mises | W-Sq | 0.862822 | Pr < W-Sq | <0.0050 |
| Anderson–Darling | A-Sq | 5.519845 | PR < A-Sq | <0.0050 |

Table 14: Kruskal–Wallis test of the small size number of jobs.

| Kruskal–Wallis test | | |
|---|---|---|
| Chi-square | DF | Pr > ChiSq |
| 204.1873 | 8 | <0.0001 |

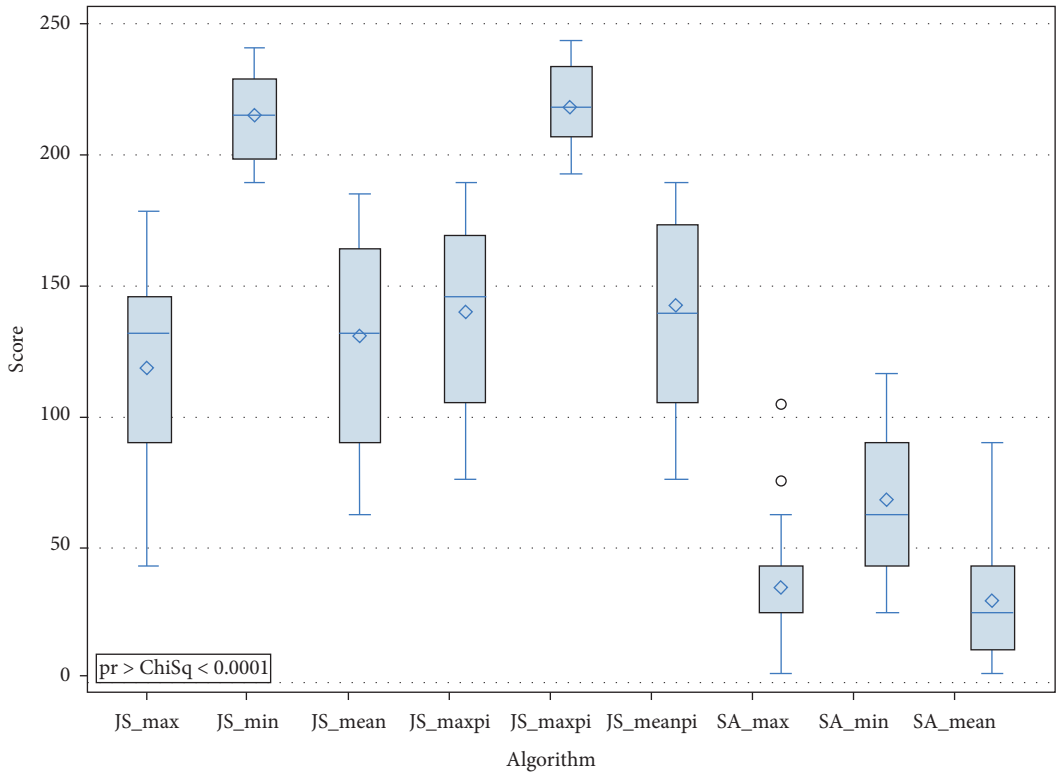



Figure 5: Scoring box of nine algorithms.

Table 15: Pairwise comparison of nine algorithms for the large size number of jobs.

| Big *n* | | |
|---|---|---|
| Algorithm | Statistic | *P* value |
| JS_max vs. JS_min | 8.9219 | <0.0001 |
| JS_max JS_mean | 1.5462 | 0.9754 |
| JS_max JS_maxpi | 2.6857 | 0.6144 |
| JS_max JS_minpi | 8.9231 | <0.0001 |
| JS_max JS_meanpi | 3.0313 | 0.4434 |
| JS_max SA_max | 8.2129 | <0.0001 |
| JS_max vs. SA_min | 6.3166 | 0.0003 |
| JS_max SA_mean | 8.4541 | <0.0001 |
| JS_min JS_mean | 8.9231 | <0.0001 |

Table 15: Continued.

| Big $n$ | | |
|---|---|---|
| Algorithm | Statistic | $P$ value |
| JS_min JS_maxpi | 8.9212 | <0.0001 |
| JS_min JS_minpi | 1.4563 | 0.9831 |
| JS_min JS_meanpi | 8.9238 | <0.0001 |
| JS_min vs. SA_max | 8.9450 | <0.0001 |
| JS_min vs. SA_min | 8.9414 | <0.0001 |
| JS_min SA_mean | 8.9677 | <0.0001 |
| JS_mean JS_maxpi | 1.2996 | 0.9920 |
| JS_mean JS_minpi | 8.9243 | <0.0001 |
| JS_mean JS_meanpi | 1.7044 | 0.9556 |
| JS_mean SA_max | 8.6226 | <0.0001 |
| JS_mean SA_min | 7.4503 | <0.0001 |
| JS_mean SA_mean | 8.7932 | <0.0001 |
| JS_maxpi JS_minpi | 8.9224 | <0.0001 |
| JS_maxpi JS_meanpi | 0.4543 | 1.0000 |
| JS_maxpi SA_ma | 8.8029 | <0.0001 |
| JS_maxpi SA_min | 8.1199 | <0.0001 |
| JS_maxpi SA_mean | 8.9094 | <0.0001 |
| JS_minpi JS_meanpi | 8.9250 | <0.0001 |
| JS_minpi SA_max | 8.9462 | <0.0001 |
| JS_minpi SA_min | 8.9426 | <0.0001 |
| JS_minpi SA_mean | 8.9689 | <0.0001 |
| JS_meanpi SA_max | 8.8186 | <0.0001 |
| JS_meanpi SA_min | 8.2291 | <0.0001 |
| JS_meanpi SA_mean | 8.9242 | <0.0001 |
| SA_max SA_min | 5.8616 | 0.0011 |
| SA_max SA_mean | 1.4043 | 0.9866 |
| SA_min SA_mean | 6.7381 | <0.0001 |

the pairwise comparison, JS_max and JS_mean are in the same group, JS_max and JS_maxpi are in the same group, JS_max and JS_meanpi are in the same group, JS_min and JS_minpi are in the same group, JS_mean and JS_meanpi are in the same group, JS_maxpi and JS_meanpi are in the same group, and SA_max and the SA_mean are in the same group. The results of the SA algorithm and JS algorithms are also significant, so they would not be divided into the same group. The RPD summary of the large size number of jobs shows the solution of the SA algorithm is comparatively better. The SA algorithm is better than the JS algorithm in this case.

## 5. Conclusions

This study focuses on two-stage three-machine flow shop assembly problems mixed with a controllable number and sum-of-processing times-based learning effect, in which job processing time is considered to be a function of the control of the truncation parameter and learning based on the sum of the processing time. We derive several dominance rules, lemmas, and lower bounds applied in the branch-and-bound method. On the other hand, three simulated annealing algorithms are proposed for finding approximate solutions. Computational results show that the SA algorithm is relatively better for the small size number of jobs. In the large size number of jobs, the solution of the SA algorithm is comparatively better. Both in the small and large size number of jobs situation, the SA algorithm is better than the JS algorithm in this study. For future research, it would be interesting to develop more powerful dominance rules and a sharper lower bound on the optimal solution for medium-size instances.

## Data Availability

All data are available on request.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## References

[1] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, third edition, 2008.

[2] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[3] W.-H. Kuo and D.-L. Yang, "Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect," *European Journal of Operational Research*, vol. 174, no. 2, pp. 1184–1190, 2006.

[4] A. Azzouz, M. Ennigrou, and L. Ben Said, "Scheduling problems under learning effects: classification and cartography," *International Journal of Production Research*, vol. 56, no. 4, pp. 1642–1661, 2018.

[5] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[6] Y. Yin, T. C. E. Cheng, and C.-C. Wu, "Scheduling with time-depending processing times," *Mathematical Problems in Engineering*, vol. 2015, Article ID 367585, 2 pages, 2015.

[7] C. Kuolamas and G. J. Kyparisis, "Single-machine and two-machine flowshop scheduling with general learning functions," *European Journal of Operational Research*, vol. 178, no. 2, pp. 402–407, 2007.

[8] P. Chen, C.-C. Wu, and W.-C. Lee, "A bi-criteria two-machine flowshop scheduling problem with a learning effect," *Journal of the Operational Research Society*, vol. 57, no. 9, pp. 1113–1125, 2006.

[9] M. C. Isler, B. Toklu, and V. Celik, "Scheduling in a two-machine flow-shop for earliness/tardiness under learning effect," *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 9-12, pp. 1129–1137, 2012.

[10] W. Lee and C.-C. Wu, "Minimizing total completion time in a two-machine flowshop with a learning effect," *International Journal of Production Economics*, vol. 88, no. 1, pp. 85–93, 2004.

[11] J.-B. Wang and Z.-Q. Xia, "Flow-shop scheduling with a learning effect," *Journal of the Operational Research Society*, vol. 56, no. 11, pp. 1325–1330, 2005.

[12] C.-C. Wu, S.-C. Liu, T. C. E. Cheng, Y. Cheng, S.-Y. Liu, and W.-C. Lin, "Re-entrant flowshop scheduling with learning considerations to minimize the makespan," *Iranian Journal of Science and Technology, Transactions A: Science*, vol. 42, no. 2, pp. 727–744, 2018a.

[13] C.-C. Wu, D.-J. Wang, S.-R. Cheng, I.-H. Chung, and W.-C. Lin, "A two-stage three-machine assembly scheduling problem with a position-based learning effect," *International Journal of Production Research*, vol. 56, no. 9, pp. 3064–3079, 2018b.

[14] C.-C. Wu, J.-Y. Chen, W.-C. Lin, K. Lai, S.-C. Liu, and P.-W. Yu, "A two-stage three-machine assembly flow shop scheduling with learning consideration to minimize the flowtime by six hybrids of particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 41, pp. 97–110, 2018c.

[15] W.-H. Wu, Y. Yin, T. C. E. Cheng et al., "A combined approach for two-agent scheduling with sum-of-processing-times-based learning effect," *Journal of the Operational Research Society*, vol. 68, no. 2, pp. 111–120, 2017.

[16] Y. Zou, W.-C. Wang, J.-Y. Lin et al., "Two-stage three-machine assembly scheduling problem with sum-of-processing-times-based learning effect," *Soft Computing*, vol. 24, no. 7, pp. 5445–5462, 2020.

[17] C.-C. Wu, J.-Y. Chen, W.-C. Lin, K. Lai, D. Bai, and S.-Y. Lai, "A two-stage three-machine assembly scheduling flowshop problem with both two-agent and learning phenomenon," *Computers & Industrial Engineering*, vol. 130, pp. 485–499, 2019a.

[18] C.-C. Wu, D. Bai, A. Azzouz et al., "A branch-and-bound algorithm and four metaheuristics for minimizing total completion time for atwo-stage assembly flow-shop scheduling problem with learning consideration," *Engineering Optimization*, vol. 52, no. 6, pp. 1009–1103, 2020.

[19] S.-C. Liu, J. Duan, W.-C. Lin et al., "A branch-and-bound algorithm for two-agent scheduling with learning effect and late work criterion," *Asia-Pacific Journal of Operational Research*, vol. 35, no. 5, pp. 1–24, 2018.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 9260479, 7 pages
https://doi.org/10.1155/2020/9260479

*Research Article*

# Due-Window Assignment and Resource Allocation Scheduling with Truncated Learning Effect and Position-Dependent Weights

**Shan-Shan Lin**

*School of Business and Management, Fujian Jiangxia University, Fuzhou 350108, China*

Correspondence should be addressed to Shan-Shan Lin; 1257290990@qq.com

Received 19 July 2020; Revised 17 August 2020; Accepted 15 September 2020; Published 8 October 2020

Academic Editor: Chin-Chia Wu



This paper studies single-machine due-window assignment scheduling problems with truncated learning effect and resource allocation simultaneously. Linear and convex resource allocation functions under common due-window (CONW) assignment are considered. The goal is to find the optimal due-window starting (finishing) time, resource allocations and job sequence that minimize a weighted sum function of earliness and tardiness, due window starting time, due window size, and total resource consumption cost, where the weight is position-dependent weight. Optimality properties and polynomial time algorithms are proposed to solve these problems.

## 1. Introduction

Scheduling models and problems with learning effects (see Biskup [1]; Lu et al. [2]; Azzouz et al. [3]; Wang et al. [4]) and/or resource allocations (see Shabtay and Steiner [5]; Yang et al. [6]) have become popular topics for scheduling researchers in recent years. Scheduling with learning effects and resource allocations simultaneously was introduced by Wang et al. [7], who focused on single-machine scheduling problems. Lu et al. [8] studied single-machine due-date assignment scheduling with learning effects and resource allocations. They proved that several problems can be solved in polynomial time. Wang and Wang [9] and Li et al. [10] considered common and slack due-window assignment problems with learning effects and resource allocations. Wang and Wang [11] considered single-machine scheduling problems with learning effects and convex resource allocation function. For the scheduling criterion (the total resource compression criterion) minimization subject to the constraint that the total resource compression criterion (the scheduling criterion) is less than or equal to a fixed constant, they proved that the problems can be solved in polynomial time. Wang et al. [12] and Liu and Jiang [13] considered due-date assignment scheduling with job-dependent learning effects and resource allocation. Liu and Jiang [14] considered flow shop due-date assignment scheduling with resource allocation and learning effect. Shi and Wang [15] considered flow shop due-window assignment scheduling with resource allocation and learning effect.

In recent years, many researchers focused on the study of scheduling with due-window, where a time interval is assumed, such that jobs completed within this interval are not penalized (Janiak et al. [16] and Wang et al. [17]). Wang et al. [18] considered the single-machine due-window scheduling problems with position-dependent weights. For the weighted sum of earliness and tardiness, due window starting time, and due window size, where the weight only dependent on its position in a sequence (i.e., a position-dependent weight), they proved that the problems can be solved in polynomial time. In this study, we continue the work of Wang et al. [18], i.e., we consider the due-window assignment scheduling problems with learning effect and resource allocation in the single-machine environment. The goal is to find the optimal due-window starting (finishing) time, resource allocations, and job sequence such that a sum of scheduling cost (including weighted sum function of

earliness and tardiness, due window starting time, due window size, where the weight is position-dependent weight) and total resource consumption cost is minimized. The contributions of this paper are given as follows. (1) The structural properties of single-machine scheduling problems are derived. (2) For the linear resource allocation, we proved that the sum of scheduling cost and total resource consumption cost can be solved in polynomial time. For the convex resource allocation, three versions of scheduling cost and total resource consumption cost can be solved in polynomial time respectively. (3) It is further extended the model to the case with slack due-window (SLKW) assignment model.

The rest of the article is organized as follows: In Section 2, we introduce the problem. In Sections 3 and 4, we provide some properties to optimally solve these problems under linear and convex resource allocation. In Section 5, we conclude the paper.

## 2. Problem Formulation

We study a scheduling problem consisting of a set of $n$ independent jobs $N = \{J_1, J_2, \ldots, J_n\}$ that need to be processed on a single machine. For the linear resource allocation, the actual processing time of job $J_j$ is

$$P_j^A = \widetilde{p}_j \max\{r^{\alpha_j}, \delta\} - \beta_j u_j, \tag{1}$$

where $\widetilde{p}_j$ is the basic processing time of job $J_j$ (i.e., the processing time without any resource allocation and truncated learning effect), $\alpha_j \le 0$ is the job-dependent learning rate (Mosheiov and Sidney [19]) of job $J_j$, $0 < \delta < 1$ is a truncation parameter (Wang et al. [20]), $\beta_j$ is the compression rate of job $J_j$, and $u_j$is the amount of resource allocated to job $J_j$ and satisfies $0 \le u_j \le \overline{u}_j \le ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/\beta_j)$.

For the convex resource allocation, the actual processing time of job $J_j$ is

$$P_j^A = \left(\frac{\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}}{u_j}\right)^{\eta}, \tag{2}$$

where $\eta > 0$ is a constant, i.e., $P_j^A$ is a convex decreasing function of resource $u_j$.

Let $[d_1, d_2]$be the common due-window for all jobs, where $d_1 \ge 0\,(d_2, d_1 \le d_2)$ denotes the starting (finishing) time of the common due window. The length of the due-window is $D = d_2 - d_1$. Both $d_1$ and $d_2$are decision variables in this paper. The goal of this paper is to find jointly the optimal due-window location, the optimal resource allocation and sequence $\pi$ such that the following objective function is minimized:

$$Z(d_1, d_2, \pi) = \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^{n} v_{[j]} u_{[j]}, \tag{3}$$

where $[j]$ denotes the job scheduled in $j$th position, $w_j\,(j = 0, 1, 2, \ldots, n, n+1)$ denotes a position- dependent weight, $L_{[j]}$ is the earliness-tardiness of job $J_{[j]}$, and

$$L_{[j]} = \begin{cases} d_1 - C_{[j]}, & \text{for} \quad d_1 > C_{[j]}, \\ 0, & \text{for} \quad d_1 \le C_{[j]} \le d_2, \\ C_{[j]} - d_2, & \text{for} \quad C_{[j]} > d_2, \end{cases} \tag{4}$$

where $C_{[j]}$ is the completion time of job $J_{[j]}$, $j = 1, 2, \ldots, n$. Using the three-field classification, the problem can be denoted as $1|\text{CONW}, P_j^A| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^{n} v_{[j]} u_{[j]}$, where $P_j^A \in \{\widetilde{p}_j \max\{r^{\alpha_j}, \delta\} - \beta_j u_j, ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^{\eta}\}$ (Graham et al. [21]), where CONW denotes the common due-window assignment. Wang et al. [18] considered single-machine scheduling problems with CONW and slack due-window (SLKW) assignments problems $1|\text{CONW}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$ and $1|\text{SLKW}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 q' + w_{n+1} D$; for the SLKW model, $[d_j', d_j'']$ is the due-window of job $J_j$ such that $d_j' \le d_j''$, where $d_j' = P_j^A + q', d_j'' = P_j^A + q'',\ j = 1, 2, \ldots, n$ $\lim_{x \to \infty}$, $q'$ and $q''$ are decision variables and $D = q'' - q'$. Wang et al. [18] proved that these both problems can be solved in $O(n \log n)$ time, respectively.

## 3. Linear Resource Allocation

**Lemma 1** [Wang et al. [18]]. *For any given sequence $\pi$, there exists an optimal sequence in which $d_1 = C_{[k]}$ for some $k$ and $d_2 = C_{[l]}$ for some $l, l \ge k$, where $\sum_{i=0}^{k-1} w_i \le w_{n+1} \le \sum_{i=0}^{k-1} w_i$, and $\sum_{i=l+1}^{n} w_i \le w_{n+1} \le \sum_{i=l}^{n} w_i$.*

**Lemma 2.** *The objective function of the problem $1|CONW, P_j^A| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^{n} v_{[j]} u_{[j]}$ can be written as*

$$\sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^{n} v_{[j]} u_{[j]} = \sum_{j=1}^{n} \xi_j P_{[j]}^A + \sum_{j=1}^{n} v_{[j]} u_{[j]}, \tag{5}$$

*where*

$$\xi_j = \begin{cases} \sum_{h=0}^{j-1} w_h, & \text{for } j = 1, 2, \ldots, k; w_{n+1}, \quad \text{for } j = k+1, k+2, \ldots, l; \sum_{h=0}^{j-1} w_h, \quad \text{for } j = l+1, l+2, \ldots, n. \end{cases} \tag{6}$$

*Proof.* From Lemma 1, we have

$$\begin{aligned} Z(d_1,d_2,\pi) &= w_0C_{[k]} + \sum_{j=1}^{k-1} w_j\left(C_{[k]} - C_{[j]}\right) + \sum_{j=l+1}^{n} w_j\left(C_{[j]} - C_{[l]}\right) + w_{n+1}\left(C_{[l]} - C_{[k]}\right) + \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= w_0\sum_{j=1}^{k} P^A_{[j]} + \sum_{j=1}^{k-1} w_j\left(\sum_{h=j+1}^{k} P^A_{[h]}\right) + \sum_{j=l+1}^{n} w_j\left(\sum_{h=j+1}^{l} P^A_{[h]}\right) + w_{n+1}\left(\sum_{h=k+1}^{l} P^A_{[h]}\right) + \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= \sum_{j=1}^{k} P^A_{[j]}\left(\sum_{h=0}^{j-1} w_h\right) + \sum_{j=l+1}^{n} P^A_{[j]}\left(\sum_{h=j}^{n} w_h\right) + w_{n+1}\left(\sum_{j=k+1}^{l} P^A_{[j]}\right) + \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= \sum_{j=1}^{n} \xi_j P^A_{[j]} + \sum_{j=1}^{n} v_{[j]}u_{[j]}, \end{aligned} \tag{7}$$

where $\xi_j$ $(j = 1, 2, \ldots, n)$ are given by (6).

From Lemma 2, we have

$$\begin{aligned} Z(d_1,d_2,\pi) &= \sum_{j=1}^{n} w_jL_{[j]} + w_0d_1 + w_{n+1}D + \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= \sum_{j=1}^{n} \xi_j\left(\widetilde{p}_{[j]}\max\{r^{\alpha_j},\delta\} - \beta_{[j]}u_{[j]}\right) + \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= \sum_{j=1}^{n} \xi_j\widetilde{p}_{[j]}\max\{r^{\alpha_{[j]}},\delta\} + \sum_{j=1}^{n}\left(v_{[j]} - \xi_j\beta_{[j]}\right)u_{[j]}. \end{aligned} \tag{8}$$

From (8), for a given sequence, the optimal resource allocation $u^*_{[j]}$ with $v_{[j]} - \xi_j\beta_{[j]} < 0$ should be $\overline{u}_{[j]}$; otherwise, $u^*_{[j]} = 0$, i.e., the optimal resource allocation of job $J_{[j]}$ is

$$u^*_{[j]} = \begin{cases} 0, & \text{if,} \quad v_{[j]} - \xi_j\beta_{[j]} \geq 0, \\ \overline{u}_{[j]}, & \text{if,} \quad v_{[j]} - \xi_j\beta_{[j]} < 0. \end{cases} \tag{9}$$

For a given sequence, from (8), we can obtain the optimal resource allocation. In order to determine the optimal sequence, let $x_{jr} = 1$ if job $J_j$ $(j = 1, 2, \ldots, n)$ is scheduled at position $r$ $(r = 1, 2, \ldots, n)$, and $x_{jr} = 0$, otherwise. Then, the problem $1|\text{CONW}, P^A_j = \widetilde{p}_j\max\{r^{\alpha_j},\delta\} - \beta_j u|\sum_{j=1}^{n} w_jL_{[j]} + w_0d_1 + w_{n+1}D + \sum_{j=1}^{n} v_{[j]}u_{[j]}$ can be solved by the following assignment problem:

$$\text{Min} \sum_{r=1}^{n}\sum_{j=1}^{n} \Psi_{jr}x_{jr}, \tag{10}$$

$$\text{S.T.} \sum_{r=1}^{n} x_{jr} = 1, \quad j = 1, 2, \ldots, n, \tag{11}$$

$$\sum_{j=1}^{n} x_{jr} = 1, \quad r = 1, 2, \ldots, n, \tag{12}$$

$$x_{jr} = 0 \text{ or } 1, \ j = 1, 2, \ldots, n, \tag{13}$$

where

$$\Psi_{jr} = \begin{cases} \xi_r\widetilde{p}_j\max\{r^{\alpha_j},\delta\}, & \text{if } v_j - \xi_r\beta_j \geq 0, \ j, r = 1, 2, \ldots, n, \\ \xi_r\widetilde{p}_j\max\{r^{\alpha_j},\delta\} + \left(v_j - \xi_r\beta_j\right)\overline{u}_j, & \text{if } v_{[j]} - \xi_r\beta_j < 0, \ j, r = 1, 2, \ldots, n. \end{cases} \tag{14}$$

And $\xi_r$ $(r = 1, 2, \ldots, n)$ are given by (6).

Based on the above analysis, the problem $1|\text{CONW}, P^A_j = \widetilde{p}_j\max\{r^{\alpha_j},\delta\} - \beta_j u|\sum_{j=1}^{n} w_jL_{[j]} + w_0d_1 + w_{n+1}D + \sum_{j=1}^{n} v_{[j]}u_{[j]}$ can be optimally solved by the following algorithm. □

*Algorithm 1*

Step 1. Calculate the indices $k$ and $l$ according to Lemma 1.

Step 2. Calculate the values $\Psi_{jr}$ by using (14).

Step 3. Solve the assignment problem (10)–(13) to determine the optimal job sequence.

Step 4. Calculate the optimal resource allocation by (7).

Step 5. Calculate $d_1 = C_{[k]}$, $d_2 = C_{[l]}$.

**Theorem 1.** *The problem* $1|\text{CONW}, P^A_j = \widetilde{p}_j\max\{r^{\alpha_j},\delta\} - \beta_j u|\sum_{j=1}^{n} w_jL_{[j]} + w_0d_1 + w_{n+1}D + \sum_{j=1}^{n} v_{[j]}u_{[j]}$ *can be solved by Algorithm 1 in* $O(n^3)$ *time.*

*Proof.* The correctness of Algorithm 1 follows from the above analysis. The time complexity of Step 1 is $O(n)$ time, Step 2 is $O(n^2)$ time, Step 3 is $O(n^3)$ time, Step 4 is $O(n)$, and 5 is $O(n)$ time. Thus, the overall computational complexity of Algorithm 1 is $O(n^3)$.

In order to illustrate Algorithm 1 for $1|\text{CONW}, P_j^A = \widetilde{p}_j \max\{r^{\alpha_j}, \delta\} - \beta_j u| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$, we present the following example. □

*Example 1.* Data: $n = 7, \delta = 0.6$, $w_0 = 19, w_1 = 20$, $w_2 = 12$, $w_3 = 7, w_4 = 14, w_5 = 24, w_6 = 22, w_7 = 15, w_8 = 22$, and the other corresponding parameters shown in Table 1.

Solution:

Step 1. According to Lemma 1, $k = 1, l = 6$.

Step 2. From (5), $\xi_1 = 19$ , $\xi_2 = \xi_3 = \xi_4 = \xi_5 = \xi_6 = 22$, $\xi_7 = 15$, and the values $\Psi_{jr}$ are given in Table 2.

Step 3. Stemming from the assignment problem (8)–(11), the optimal job sequence is $\pi = (J_4, J_7, J_2, J_6, J_1, J_5, J_3)$.

Step 4. From (7), the optimal resource allocation is $u_4 = 1$, $u_7 = 5$, $u_2 = 3$, $u_6 = 4$, $u_1 = 5$, $u_5 = 2$, $u_3 = 0$.

Step 5. Calculate $d_1 = C_{[1]} = C_4 = 7$, $d_2 = C_{[6]} = C_5 = 22.67617$, and $\sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n w_j L_{[j]} = 886.8757$.

## 4. Convex Resource Allocation

*4.1. Problem* $1|CONW, P_j^A = (\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}/u_j)| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$. From Lemma 2 and $P_j^A = ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^\eta$, we have

$$\sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]} = \sum_{j=1}^n \xi_j \left( \frac{\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}}{u_j} \right)^\eta + \sum_{j=1}^n \nu_{[j]} u_{[j]}, \tag{15}$$

where $\xi_j$ $(j = 1, 2, \ldots, n)$ are given by (6).

By taking the first derivative of the objective given by (15) with respect to $u_{[j]}$, equating it to zero and solving it for $J_{[j]}$, we have (16).

**Lemma 3.** *For a given sequence, the optimal resource allocation of the problem* $1|CONW, P_j^A = (\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}/u_j)| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$ *is*

$$u_{[j]}^* = \left( \frac{\eta \xi_j}{\nu_{[j]}} \right)^{(1/\eta+1)} \times \left( \widetilde{p}_{[j]} \max\{ j^{\alpha_{[j]}}, \delta \} \right)^{(\eta/\eta+1)}. \tag{16}$$

*By substituting (16) into (15), we have*

$$\sum_{j=1}^n \xi_j P_{[j]}^A + \sum_{j=1}^n \nu_{[j]} u_{[j]} = \left( \eta^{(-\eta/\eta+1)} + \eta^{(1/\eta+1)} \right) \sum_{j=1}^n \left( \nu_{[j]} \widetilde{p}_{[j]} \max\{ j^{\alpha_{[j]}}, \delta \} \right)^{(\eta/\eta+1)} \left( \xi_j \right)^{(1/\eta+1)}. \tag{17}$$

Table 1: Data for Example 1.

| $J_j$ | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ | $J_7$ |
|---|---|---|---|---|---|---|---|
| $\widetilde{p}_j$ | 23 | 17 | 19 | 10 | 18 | 16 | 9 |
| $\alpha_j$ | −0.32 | −0.24 | −0.33 | −0.25 | −0.28 | −0.3 | −0.29 |
| $\beta_j$ | 2 | 3 | 1 | 3 | 4 | 2 | 1 |
| $\nu_j$ | 3 | 14 | 30 | 9 | 6 | 15 | 20 |
| $\overline{u}_j$ | 5 | 3 | 6 | 1 | 2 | 4 | 5 |

*Let*

$$\Psi_{jr} = \left( \eta^{(-\eta/\eta+1)} + \eta^{(1/\eta+1)} \right) \left( \nu_j \widetilde{p}_j \max\{r^{\alpha_j}, \delta\} \right)^{(\eta/\eta+1)} (\xi_r)^{(1/\eta+1)}, \tag{18}$$

*where* $\xi_r$ $(r = 1, 2, \ldots, n)$ *are given by (6).*

*As in Section 3, for the problem* $1|CONW, P_j^A = (\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}/u_j)| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$, *we can propose the following algorithm:*

*Algorithm 2*

Step 1. Calculate the indices $k$ and $l$ according to Lemma 1.

Step 2. Calculate the values $\Psi_{jr}$ by using (18).

Step 3. Solve the assignment problem (10)–(13) to determine the optimal job sequence.

Step 4. Calculate the optimal resource allocation by (16).

Step 5. Calculate $d_1 = C_{[k]}$, $d_2 = C_{[l]}$.

**Theorem 2.** *Algorithm 2 solves the problem* $1|CONW, P_j^A = (\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}/u_j)| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$ *in* $O(n^3)$ *time.*

*In order to illustrate Algorithm 2 for* $1|CONW, P_j^A = (\widetilde{p}_j \max\{r^{\alpha_j}, \delta\}/u_j)| \sum_{j=1}^n w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^n \nu_{[j]} u_{[j]}$, *we present the following example.*

*Example 2.* Consider $n = 7, \delta = 0.6$, $\eta = 2$, $w_0 = 9, w_1 = 8$, $w_2 = 12, w_3 = 7, w_4 = 14, w_5 = 24, w_6 = 5, w_7 = 15, w_8 = 22$, and the other corresponding parameters shown in Table 3.

Solution:

Step 1. According to Lemma 1, $k = 2, l = 5$.

Step 2. From (5), $\xi_1 = 9$, $\xi_2 = 17$, $\xi_3 = \xi_4 = \xi_5 = 22$, $\xi_6 = 20, \xi_7 = 15$, and the values $\Psi_{jr}$ are given in Table 4.

Step 3. Stemming from the assignment problem (8)–(11), the optimal job sequence is $\pi = (J_2, J_1, J_3, J_7, J_6, J_5, J_4)$.

Step 4. From (14), the optimal resource allocation is $u_2 = 10.91533$, $u_1 = 10.71178$, $u_3 = 11.53352$, $u_7 = 5.841858, u_6 = 9.501238, u_5 = 9.251873, u_4 = 5.013141$.

Table 2: Values $\Psi_{jr}$ for Example 1.

| | $(j/r)$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| | 1 | 262.0000 | 200.3414 | 151.0178 | 119.7068 | **98.60000** | 98.60000 | 72.00000 |
| | 2 | 194.0000 | 160.6827 | **131.3178** | 112.1496 | 98.16679 | 87.28498 | 66.85148 |
| | 3 | 361.0000 | 332.5343 | 290.8883 | 264.5431 | 250.8000 | 250.8000 | **171.0000** |
| $\Psi_{jr=}$ | 4 | **142.0000** | 127.9972 | 110.1639 | 98.56349 | 90.12287 | 83.56748 | 56.21822 |
| | 5 | 202.0000 | 162.1420 | 127.1396 | 104.6077 | 88.33854 | **75.77993** | 54.00000 |
| | 6 | 212.0000 | 169.9128 | 137.1665 | **116.2334** | 101.1959 | 95.20000 | 84.00000 |
| | 7 | 171.0000 | **151.9446** | 133.9793 | 122.4548 | 114.1549 | 108.8000 | 81.00000 |

Bold numbers are the optimal solution.

Table 3: Data for Example 2.

| $J_j$ | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ | $J_7$ |
|---|---|---|---|---|---|---|---|
| $\widetilde{p}_j$ | 13 | 17 | 12 | 10 | 18 | 16 | 9 |
| $\alpha_j$ | −0.32 | −0.24 | −0.33 | −0.25 | −0.28 | −0.3 | −0.29 |
| $v_j$ | 3 | 4 | 2 | 9 | 6 | 5 | 8 |

Table 4: Values $\Psi_{jr}$ for Example 2.

| | $(j/r)$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| | 1 | 45.20903 | **48.20302** | 48.17622 | 45.30844 | 43.32309 | 41.96834 | 38.13077 |
| | 2 | **65.49197** | 72.45895 | 74.00176 | 70.67272 | 68.19401 | 64.16226 | 56.87507 |
| | 3 | 32.70817 | 34.71350 | **34.60055** | 32.47854 | 31.34372 | 30.36358 | 27.58714 |
| $\Psi_{jr=}$ | 4 | 78.94848 | 86.94426 | 88.55579 | 84.40999 | 81.32839 | 76.42716 | **67.67740** |
| | 5 | 89.15204 | 96.82953 | 97.82774 | 92.71289 | 88.93040 | **83.26686** | 75.19374 |
| | 6 | 72.98642 | 78.54253 | 78.92439 | 74.51153 | **71.25929** | 67.75459 | 61.55913 |
| | 7 | 68.03574 | 73.55408 | 74.11175 | **70.10229** | 67.14231 | 63.15878 | 57.38356 |

Bold numbers are the optimal solution.

Step 5. Calculate $d_1 = C_{[2]} = C_1 = 3.370785$, $d_2 = C_{[5]} = C_6 = 6.0368777$, and $\sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D + \sum_{j=1}^{n} v_{[j]} u_{[j]} = 440.6014$.

*4.2. Problem* $1|CONW, P_j^A = ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^{\eta}, \sum_{j=1}^{n} v_{[j]} u_{[j]} \le \overline{U}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$. In this section, we aim to minimize the following cost function $\sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$ subject to $\sum_{j=1}^{n} v_{[j]} u_{[j]} \le \overline{U}$, $1|\text{CONW}, P_j^A = ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^{\eta}, \sum_{j=1}^{n} v_{[j]} u_{[j]} \le \overline{U}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$ where $\overline{U}$ is a limitation on the total resource consumption cost. Obviously, in an optimal solution for the problem $1|\text{CONW}, P_j^A = ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^{\eta}, \sum_{j=1}^{n} v_{[j]} u_{[j]} \le \overline{U}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$ the constraint will be satisfied as $\sum_{j=1}^{n} v_{[j]} u_{[j]} = \overline{U}$.

**Lemma 4.** *For a given sequence, the optimal resource allocation of the problem* $1|CONW, P_j^A = ((\widetilde{p}_j \max\{r^{\alpha_j}, \delta\})/u_j)^{\eta}, \sum_{j=1}^{n} v_{[j]} u_{[j]} \le \overline{U}| \sum_{j=1}^{n} w_j L_{[j]} + w_0 d_1 + w_{n+1} D$ *is*

$$u_{[j]}^{*} = \frac{\left(\xi_j\right)^{(1/\eta+1)} \left(v_{[j]}\right)^{(-1/\eta+1)} \left(\widetilde{p}_{[j]} \max\left\{j^{\alpha_{[j]}}, \delta\right\}\right)^{(\eta/\eta+1)}}{\sum_{j=1}^{n} \left(\xi_j\right)^{(1/\eta+1)} \left(\widetilde{p}_{[j]} \max\left\{j^{\alpha_{[j]}}, \delta\right\}\right)^{(\eta/\eta+1)}} \times \overline{U}, \quad j = 1, 2, \ldots, n, \tag{19}$$

*where* $\xi_j$ $(j = 1, 2, \ldots, n)$ *are given by (6).*

*Proof.* For a given sequence $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, the Lagrange function is

$$L(u, \lambda) = \sum_{j=1}^{n} \xi_j P_{[j]}^A + \lambda \left( \sum_{j=1}^{n} v_{[j]} u_{[j]} - \overline{U} \right) = \sum_{j=1}^{n} \xi_j \left( \frac{\widetilde{p}_{[j]} \max\left\{j^{\alpha_{[j]}}, \delta\right\}}{u_{[j]}} \right)^{\eta} + \lambda \left( \sum_{j=1}^{n} v_{[j]} u_{[j]} - \overline{U} \right), \tag{20}$$

where $\lambda$ is the Lagrangian multiplier. Deriving (20) with respect to $u_{[j]}$ and $\lambda$, we have

$$\frac{\partial L(u, \lambda)}{\partial u_{[j]}} = \lambda v_{[j]} - \eta \xi_j \times \frac{\left(\widetilde{p}_j \max\left\{j^{\alpha_{[j]}}, \delta\right\}\right)^{\eta}}{\left(u_{[j]}\right)^{\eta+1}} = 0. \tag{21}$$

It follows that

$$u_{[j]}^{*} = \frac{\left(\eta \xi_j \left(\widetilde{p}_{[j]} \max\left\{j^{\alpha_{[j]}}, \delta\right\}\right)^{\eta}\right)^{(1/\eta+1)}}{\left(\lambda v_{[j]}\right)^{(1/\eta+1)}}. \tag{22}$$

From $\sum_{j=1}^{n} v_{[j]}u_{[j]} = \overline{U}$, we have

$$\lambda^{(1/\eta+1)} = \frac{\sum\left(\eta\xi_j\right)^{(1/\eta+1)}\left(\widetilde{p}_{[j]}v_{[j]}\max\{j^{\alpha_{[j]}},\delta\}\right)^{(\eta/\eta+1)}}{\overline{U}}. \quad (23)$$

Finally, inserting (23) into (22), we have (19).

By substituting the values $u_{[j]}^{*}$ given in (19) into $\sum_{j=1}^{n}\xi_j P_{[j]}^{A} = \sum_{j=1}^{n}\xi_j((\widetilde{p}_{[j]}\max\{j^{\alpha_{[j]}},\delta\}/u_{[j]}))^{\eta}$, we have

$$\sum_{j=1}^{n}\xi_j P_{[j]}^{A} = \overline{U}^{-\eta}\left(\sum_{j=1}^{n}\left(v_{[j]}\widetilde{p}_{[j]}\max\{j^{\alpha_{[j]}},\delta\}\right)^{(\eta/\eta+1)}\left(\xi_j\right)^{(1/\eta+1)}\right)^{\eta+1}. \quad (24)$$

Similarly to Section 4.1, we have the following. □

**Theorem 3.** *Problem* $1|CONW, P_j^A = (((\widetilde{p}_j\max\{r^{\alpha_j},\delta\})/u_j))^{\eta}, \sum_{j=1}^{n}v_{[j]}u_{[j]} \le \overline{U}|\sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D$ *can be solved in* $O(n^3)$ *time.*

*4.3. Problem* $1|CONW, P_j^A = ((\widetilde{p}_j\max\{r^{\alpha_j},\delta\})/u_j)^{\eta}, \sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D \le \overline{V}|\sum_{j=1}^{n}v_{[j]}u_{[j]}$. In this section, the inverse version' of $1|\mathrm{CONW}, P_j^A = ((\widetilde{p}_j\max\{r^{\alpha_j},\delta\}/u_j))^{\eta}$, $\sum_{j=1}^{n}v_{[j]}u_{[j]} \le \overline{U}|\sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D$ will be considered, i.e., the problem of minimizing $\sum_{j=1}^{n}v_{[j]}u_{[j]}$ subject to $\sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D \le \overline{V}$, where $\overline{V}$ is a given real number.

Similarly to Section 4.2, we have.

**Lemma 5.** *For a given sequence, the optimal resource allocation of the problem* $1|CONW, P_j^A = ((\widetilde{p}_j\max\{r^{\alpha_j},\delta\})/u_j)^{\eta}, \sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D \le \overline{V}|\sum_{j=1}^{n}v_{[j]}u_{[j]}$ *is*

$$u_{[j]}^{*} = \overline{V}^{-(1/\eta)}\left(\xi_j\right)^{(1/\eta+1)}\left(v_{[j]}\right)^{(-1/\eta+1)}\left(\widetilde{p}_{[j]}\max\{j^{\alpha_{[j]}},\delta\}\right)^{(\eta/\eta+1)}\left(\sum_{j=1}^{n}\left(\xi_j\right)^{(1/\eta+1)}\left(\widetilde{p}_{[j]}v_{[j]}\max\{j^{\alpha_{[j]}},\delta\}\right)^{(\eta/\eta+1)}\right)^{(1/\eta)}, \quad (25)$$

*where* $\xi_j$ $(j = 1, 2, \ldots, n)$ *are given by (6).*

*By substituting the values* $u_{[j]}^{*}$ *given in (25) into* $\sum_{j=1}^{n}v_{[j]}u_{[j]}$, *we have*

$$\sum_{j=1}^{n}v_{[j]}u_{[j]} = \overline{V}^{-(1/\eta)}\left(\sum_{j=1}^{n}\left(v_{[j]}\widetilde{p}_{[j]}\max\{j^{\alpha_{[j]}},\delta\}\right)^{(\eta/\eta+1)}\left(\xi_j\right)^{(1/\eta+1)}\right)^{(1/\eta)+1}. \quad (26)$$

*Similarly to Section 4.2, we have.*

**Theorem 4.** *Problem* $1|CONW, P_j^A = ((\widetilde{p}_j\max\{r^{\alpha_j},\delta\})/u_j)^{\eta}, \sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D \le \overline{V}|\sum_{j=1}^{n}v_{[j]}u_{[j]}$ *can be solved in* $O(n^3)$ *time.*

*Remark.* Obviously, the CONW model can be extended to the slack due-window (SLKW) assignment model. The objective function $\sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D$ can be replaced by

$$\sum_{j=1}^{n}w_jL_{[j]} + w_0q' + w_{n+1}D, \quad (27)$$

where

$$L_{[j]} = \begin{cases} d'_{[j]} - C_{[j]}, & \text{for} \quad d'_{[j]} > C_{[j]}, \\ 0, & \text{for} \quad d_1 \le C_{[j]} \le d_2, \\ C_{[j]} - d''_{[j]}, & \text{for} \quad C_{[j]} > d''_{[j]}, \end{cases} \quad (28)$$

and $D = d''_j - d'_j = q'' - q'$.

## 5. Conclusions

This paper considered the single-machine due-window assignment scheduling problems with learning effect and resource allocation. For the linear and convex resource allocations, we showed that some different models are polynomially solvable, respectively. Future research may focus on the flow shop scheduling problems with learning effect and resource allocation or study the Pareto-optimal solutions with respect to the criterion $\sum_{j=1}^{n}w_jL_{[j]} + w_0d_1 + w_{n+1}D$ and the resource compression cost $\sum_{j=1}^{n}v_{[j]}u_{[j]}$.

## Data Availability

No data were used in the study.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## References

[1] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[2] Y.-Y. Lu, F. Teng, and Z.-X. Feng, "Scheduling jobs with truncated exponential sum-of-logarithm-processing-times based and position-based learning effects," *Asia-Pacific*

*Journal of Operational Research*, vol. 32, no. 3, pp. 1–17, Article ID 1550026, 2015.

[3] A. Azzouz, M. Ennigrou, and L. Ben Said, "Scheduling problems under learning effects: Classification and cartography," *International Journal of Production Research*, vol. 56, no. 4, pp. 1642–1661, 2018.

[4] J.-B. Wang, F. Liu, and J.-J. Wang, "Research on m -machine flow shop scheduling with truncated learning effects," *International Transactions in Operational Research*, vol. 26, no. 3, pp. 1135–1151, 2019.

[5] D. Shabtay and G. Steiner, "A survey of scheduling with controllable processing times," *Discrete Applied Mathematics*, vol. 155, no. 13, pp. 1643–1666, 2007.

[6] D.-L. Yang, C.-J. Lai, and S.-J. Yang, "Scheduling problems with multiple due windows assignment and controllable processing times on a single machine," *International Journal of Production Economics*, vol. 150, pp. 96–103, 2014.

[7] D. Wang, M.-Z. Wang, and J.-B. Wang, "Single-machine scheduling with learning effect and resource-dependent processing times," *Computers & Industrial Engineering*, vol. 59, no. 3, pp. 458–462, 2010.

[8] Y.-Y. Lu, G. Li, Y.-B. Wu, and P. Ji, "Optimal due-date assignment problem with learning effect and resource-dependent processing times," *Optimization Letters*, vol. 8, no. 1, pp. 113–127, 2014.

[9] J.-B. Wang and M.-Z. Wang, "Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 5, pp. 1–15, Article ID 1450036, 2014.

[10] G. Li, M.-L. Luo, W.-J. Zhang, and X.-Y. Wang, "Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation," *International Journal of Production Research*, vol. 53, no. 4, pp. 1228–1241, 2015.

[11] J.-B. Wang and J.-J. Wang, "Research on scheduling with job-dependent learning effect and convex resource-dependent processing times," *International Journal of Production Research*, vol. 53, no. 19, pp. 5826–5836, 2015.

[12] J.-B. Wang, X.-N. Geng, L. Liu, J.-J. Wang, and Y.-Y. Lu, "Single machine CON/SLK due date assignment scheduling with controllable processing time and job-dependent learning effects," *The Computer Journal*, vol. 61, no. 9, pp. 1329–1337, 2018.

[13] W. Liu and C. Jiang, "Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation," *Engineering Optimization*, vol. 52, no. 1, pp. 74–89, 2020.

[14] W.-W. Liu and C. Jiang, "Flow shop resource allocation scheduling with due date assignment, learning effect and position-dependent weights," *Asia-Pacific Journal of Operational Research*, vol. 37, no. 3, pp. 1–27, Article ID 2050014, 2020.

[15] H.-B. Shi and J.-B. Wang, "Research on common due window assignment flowshop scheduling with learning effect and resource allocation," *Engineering Optimization*, vol. 52, no. 4, pp. 669–686, 2020.

[16] A. Janiak, W. A. Janiak, T. Krysiak, and T. Kwiatkowski, "A survey on scheduling problems with due windows," *European Journal of Operational Research*, vol. 242, no. 2, pp. 347–357, 2015.

[17] L.-Y. Wang, D.-Y. Lv, B. Zhang, W.-W. Liu, and J.-B. Wang, "Optimization for different due-window assignment scheduling with position-dependent weights," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–7, Article ID 9746538, 2020.

[18] J.-B. Wang, B. Zhang, L. Li, D. Bai, and Y.-B. Feng, "Due-window assignment scheduling problems with position-dependent weights on a single machine," *Engineering Optimization*, vol. 52, no. 2, pp. 185–193, 2020.

[19] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.

[20] X.-R. Wang, J.-B. Wang, J. Jin, and P. Ji, "Single machine scheduling with truncated job-dependent learning effect," *Optimization Letters*, vol. 8, no. 2, pp. 669–677, 2014.

[21] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 5241710, 5 pages
https://doi.org/10.1155/2020/5241710

*Research Article*

# A Base on Fuzzy Theory to Supplier Evaluation and Selection Optimization

**Chun-Tsai Lin**

*Department of Food & Beverage Management, Cheng Shiu University, Kaohsiung, Taiwan*

Correspondence should be addressed to Chun-Tsai Lin; capital789@yahoo.com.tw

Received 27 July 2020; Revised 9 September 2020; Accepted 23 September 2020; Published 5 October 2020

Academic Editor: Yunqiang Yin



Supplier evaluation and selection is critical for schools to save costs. Supplier evaluation indexes must be developed before evaluation. In addition, how to select the most appropriate supplier and make the best procurement decision are also important topics. This study selected three most important supplier evaluation and selection criteria and established a supplier evaluation model through fuzzy logic inference using MATLAB. This method can overcome the defects in some methodologies and improve organizational capability and competitiveness. This model can also be used to address practical supplier evaluation and selection matters. Modelling took several steps, including fuzzification, fuzzy rule base, fuzzy inference engine, and defuzzification. Then, a supplier evaluation model was established based on the fuzzy rules obtained from the abovementioned operations. This model can provide a basis for school purchasers or managers in executing supplier selection policies in the future.

## 1. Introduction

With increasingly intense market competition and growing consumer awareness over recent years, consumers are provided with more and more opportunities to select the products they need. Mass, simple, and standard production cannot effectively satisfy ever-changing consumer demands any more. Manufacturers must be able to efficiently provide diverse products to satisfy consumer demands and maintain their presence against fierce competition. Schools also represent a group of consumers who frequently procure large amounts of various products. Therefore, supplier selection not only affects a school's procurement cost but also affects its operation quality. Supplier selection is a key step in the procurement process. A school that does not engage an appropriate supplier to provide low-cost and high-quality products cannot attract students, which will invisibly increase costs. This study, based on the data acquired from a supplier database and literature review, used fuzzy logic to propose an evaluation method which addressed such problems. Using a germane supplier selection and evaluation model to cut school procurement cost, it is also the key for schools to increase school competitiveness. Therefore, a supplier evaluation model provides an important basis for school procurement decisions [1, 2].

## 2. Fuzzy Theory Method

Fuzzy theory is a famous paper "Fuzzy Sets (Fuzzy Sets), which was published by Professor LA Zadeh of the University of California, Berkeley, in 1965, in the Information and Control (Information and Control) academic journal)." [3]. He believes that traditional control theory places too much emphasis on precision to master complex systems. If you use the concepts of fuzzy sets and continuous membership functions (Membership Function) to control, you won't have this problem. In 1973, he published a paper introducing the concept of semantic variables and proposed the use of fuzzy logic (Fuzzy If-Then Rule) [4] to formulate human knowledge and establish the basis of fuzzy control. In 1978, the first fuzzy controller appeared to verify the feasibility of fuzzy theory [5]. However, the scholars at that time insisted on the traditional set theory, so when the fuzzy theory was put forward, it received many criticisms, but it developed rapidly in the following two decades and combined with other different theories to become a new branch

of mathematics, and the related applications were also exponential. Growth, such as the application of fuzzy logic systems [6], image recognition [7], time series prediction [8], intelligent robots [9], and its scope, is quite extensive (Figure 1). The basic structure of the fuzzy system is shown in Figures 2 and 3. The main functional blocks include: (1) fuzzification, (2) fuzzy rule base, (3) fuzzy inference engine, and (4) defuzzification.

*2.1. Fuzzification and Attribution Function.* Fuzzy theory uses the attribute function of the range in the interval [0, 1] to represent the fuzzy set on the range $U$.Common fuzzy attribution functions include trapezoidal and triangular attribution functions.

(1) Triangular attribute function, as shown in Figure 2. The mathematical expression of the triangle attribution function is

$$\mu_A(x) = \begin{cases} \dfrac{x-k}{m-k}, & k \le x \le n, \\ \dfrac{r-x}{r-m}, & m < x \le r, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

(2) Trapezoidal attribute function, as shown in Figure 3. The mathematical expression of the trapezoidal attribute function is

$$\mu_A(x) = \begin{cases} \dfrac{x-k}{a-k}, & k \le x \le a, \\ 1, & a \le x \le b, \\ \dfrac{r-x}{r-b}, & b < x \le r, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

*2.2. Fuzzy Rule Base.* Fuzzy Rule Base is a classification rule of fuzzy theory. The design method is expressed in the form of the If-Then column formula, and the fuzzy control algorithm is shown in the following formula:

$$\begin{aligned} &\text{If } \langle x_1 \rangle \text{ is } \langle A_1 \rangle \text{ and} \ldots \text{and } \langle x_i \rangle \text{ is } \langle A_i \rangle \quad i = 1, 2, \ldots, n, \\ &\text{Then } y \text{ is } \langle \text{action}_i \rangle, \end{aligned} \tag{3}$$

where $n$ is the number of control rules, $x$ is the input parameter, $A$ is the condition, and action is the corresponding action based on the previous conditions. Generally speaking, there are three main methods for generating rule bases:

(1) Based on expert knowledge and experience: a fuzzy rule library is designed based on expert consultation with experience in this research field or reference books with reference value.

(2) Response according to the input of the controlled object: according to the input and output reaction actions of the controlled object, it can be summarized as the method of system identification.

(3) Through self-learning: after collecting some measurement data, the data are measured by a specific training algorithm, and fuzzy rules are extracted from it, for example, Genetic Algorithm (GA) and Artificial Neural Network (ANN).

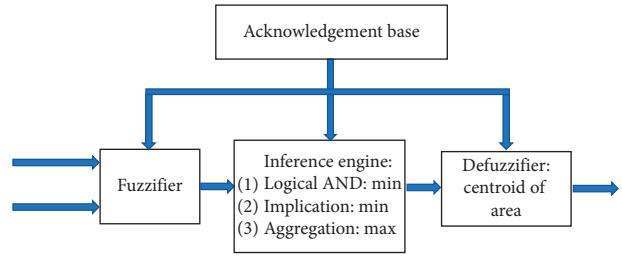



Figure 1: Fuzzy inference.

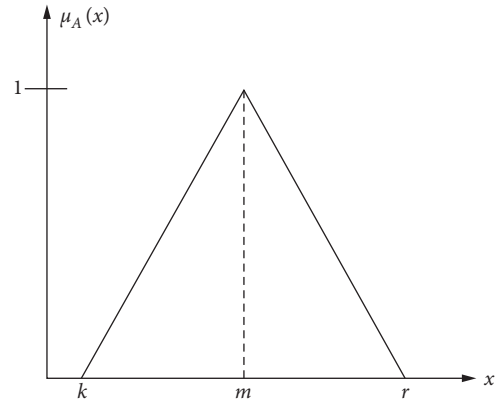



Figure 2: Triangle attribute function.

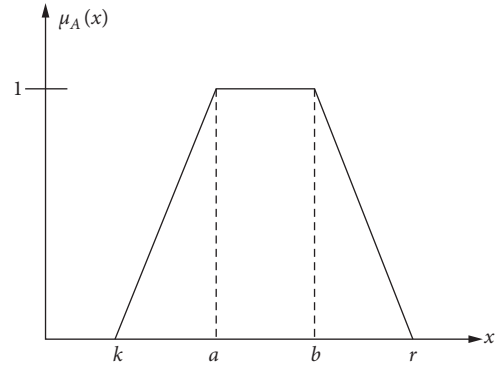



Figure 3: Trapezoidal attribution function.

*2.3. Fuzzy Inference Engine.* The fuzzy inference engine is the core of the fuzzy system. It simulates human thinking and decision-making modes through approximate inference or

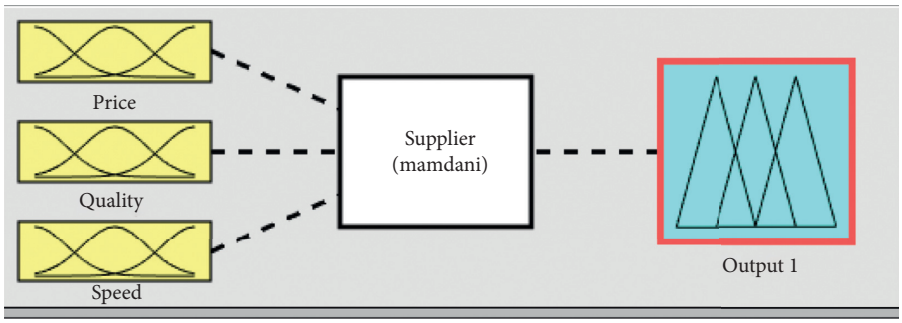



Figure 4: Fuzzy inference system.

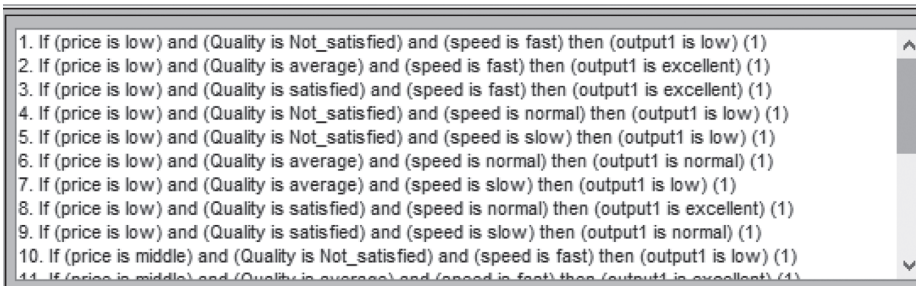



Figure 5: Fuzzy rules.

fuzzy inference to achieve the main purpose of problem solving. In this research, the maximum-minimum composite operation in fuzzy inference is used. The formula is as follows:

Maximum-minimum composite operation:

$$\mu_{C'}(z) = \max_{x}\left[\min\left(\mu_A(x), \mu_{A\longrightarrow B}(x, y)\right)\right]. \quad (4)$$

*2.4. Defuzzification.* After the parameters are fuzzified and fuzzy inference engine, clear values must be solved through defuzzification. The process of converting the conclusions generated after fuzzy inference into clear values is called "defuzzification." The legal meaning is to calculate the area center point of the overlapping area, and the set value of the position of the center of gravity point is the fuzzy output value as shown in the following formula:

$$y^* = \frac{\left[\sum_{i=1}^{N}\mu_C(y_i)\cdot y_i\right]}{\sum_{i=1}^{N}\mu_C(y_i)}. \quad (5)$$

Among them, $N$ is the total number of fuzzy rules, $y_i$ is the inferred result of the $i$-th fuzzy rule, $\mu_C(y_i)$ represents the attribution value of the $i$-th rule fuzzy set.

## 3. Simulation Results

This research is based on the supplier data established over the years as a database. Three variables are selected including price, delivery speed, and quality variables. Based on this, a set of fuzzy rules is compiled, and the design steps are as follows:

(1) Input and output language variables are created

(2) Relevant terms are established, and attribution functions are defined

(3) Fuzzy rules and inferences are given

(4) Deblurring

This study designed and constructed a supplier evaluation model based on this rule.

*3.1. Creating Input and Output Language Variables.* This research uses fuzzy logic inference tools to establish a supplier system and uses fuzzy algorithms to make judgments. Before making judgments, first, the input and output parameters are confirmed, and price, delivery date, and quality are choosen as input parameters.

*3.2. Establishing Related Terms and Attribution Functions.* In the fuzzy judgment, the relevant terms are corresponding to the input language variables, and the input and output settings of each parameter are as follows:

We enter the parameters of price and quality in three steps of low, medium, and high and the delivery speed in three steps of fast, normal, and delayed. After establishing the relevant terminology, an attribution function must be defined for each term. The input and output functions of this study use triangular attribution functions. The various central values of the attribution function and the size of its range will affect the fuzzy value of the input variable, which is good for fuzzy judgment design. The influence is very large, and the suitable value range designed in this study is shown in Figures 4 and 5.

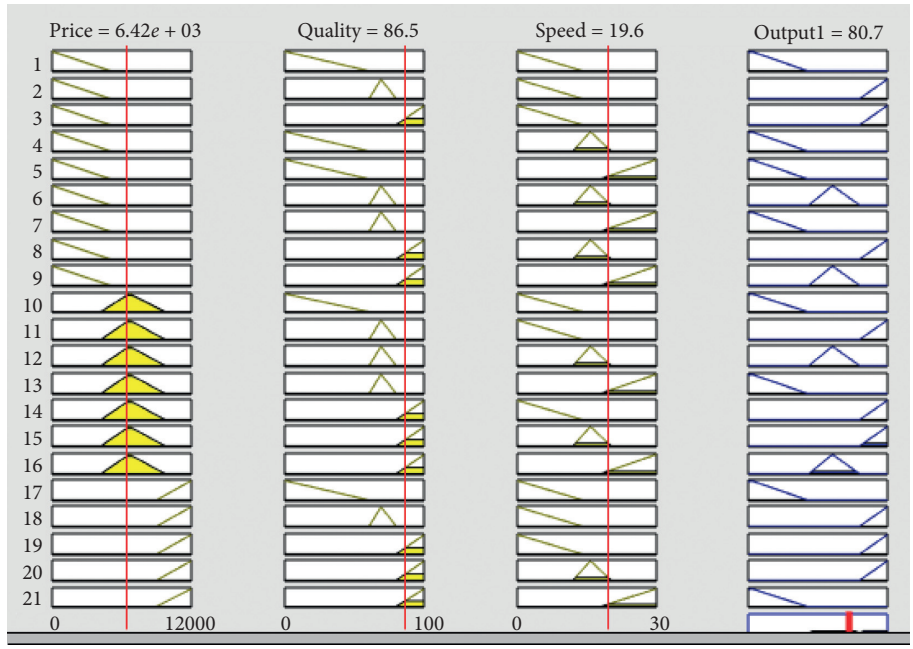



Figure 6: Fuzzy inference.

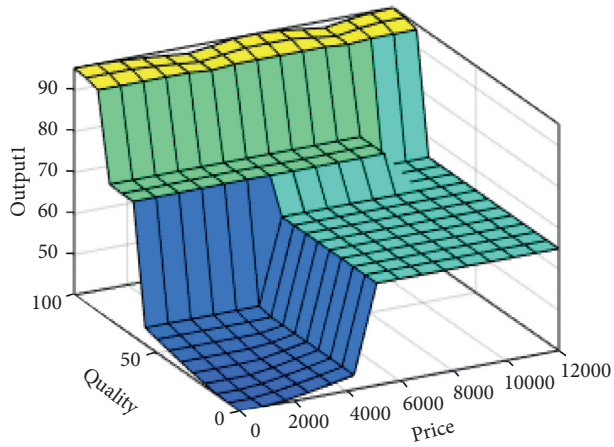



Figure 7: Surface plot.

## 4. Conclusions

This study acquired data from a database containing years of supplier data and carried out fuzzy inference and defuzzification using MATLAB. The centroid method was used for defuzzification. As inferenced by it, 27 rules were triggered in total. As shown in the figure, when the price is above NT$ 5000, quality is above medium and high levels, delivery is efficient within 20 days, and 80.7% of suppliers are reliable, as shown in Figures 6 and 7.

Supplier selection is the first step to establish a supplier system. Selection of appropriate suppliers is helpful to school procurement evaluation. In contrast, if inappropriate or uncooperative suppliers are selected, schools (and even their operations) will be negatively affected. Therefore, cautions must be exercised in selecting suppliers. The fuzzy multiple criteria method was used for analyzing and validating the efficacy and feasibility of the model.

## Data Availability

After the article is published and accepted, data sharing and use can be allowed.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by Cheng Shiu University.

## References

[1] A. Brun and A. P. Staudacher, "Negotiation-driven supply-Chain Co-ordination small and medium enterprises," in *Proceedings of the ECAI 2000 Workshop "13-gent Technologies and their Application Scenarios in Logistics,"* I. J. Timm, Ed., pp. 55–60, Berlin, Germany, July 2000.

[2] M. Wouters, J. C. Anderson, and F. Wynstra, "The adoption of total cost of ownership for sourcing decisions-a structural equations analysis," *Accounting, Organizations and Society*, vol. 30, no. 2, pp. 167–191, 2005.

[3] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[4] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[5] W. Li-Xin, *A Course in Fuzzy Systems and Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1997.

[6] T. Munakata and Y. Jani, "Fuzzy systems: an overview," *Communications of the ACM*, vol. 37, pp. 69–96, 1994.

[7] J. C. Bezdek, *Pattern Recognition with Fuzzy Oβjective Function Algorithms*, Springer Science & Business Media, Berlin, Germany, 2013.

[8] A. Pouraβdollah, C. Wagner, J. H. Aladi, and J. M. Gariβaldi, "Improved uncertainty capture for nonsingleton fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1513–1524, 2016.

[9] Y. P. Kondratenko, O. V. Kozlov, O. S. Gerasin, and Y. M. Zaporozhets, "Synthesis and research of neuro-fuzzy o*β*server of clamping force for mo*β*ile ro*β*ot automatic control system," in *Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, pp. 90–95, Lviv, Ukraine, August 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 2510413, 15 pages
https://doi.org/10.1155/2020/2510413

*Research Article*

# Integrated Inventory-Transportation Scheduling with Sustainability-Dependent Demand under Carbon Emission Policies

**Zhongming Tang,[1] Xingxing Liu,[2] Ying Wang,[1] and Da Ma[1]**

[1]*School of Management, Hubei University of Education, Wuhan 430205, China*
[2]*School of Safety Science and Emergency Management, Wuhan University of Technology, Wuhan 430070, China*

Correspondence should be addressed to Da Ma; mada0618@163.com

Received 14 July 2020; Revised 20 August 2020; Accepted 31 August 2020; Published 26 September 2020

Academic Editor: Chin-Chia Wu



Carbon policies and consumer environmental consciousness are effective motivators that drive enterprises to adopt sustainability technology. To provide enterprises insights into sustainable investment and inventory-transportation decision-making and governments insights into policy-making, this study investigates integrated inventory-transportation scheduling considering consumer environmental consciousness and sustainability technology under carbon cap, tax, and cap-and-trade policies. We first examined sustainability that extends the economic order quantity (EOQ) models, simultaneously taking into account the comprehensive emission model, consumer environmental consciousness, and carbon policies. We then optimized the sustainability level and EOQ using the simulation method. Furthermore, we performed a regression analysis on the carbon policy effects on sustainability level, profit, and emission. Moreover, using the regression models, we estimated and discussed the optimal policy parameters from the perspective of social welfare maximization. The results indicate that the carbon cap-and-trade policy is superior to carbon cap and tax policies. Under carbon cap and tax policies, the tougher the carbon policy, the higher the sustainability level and the lower the profit and carbon emission. Meanwhile, under the carbon cap-and-trade policy, the carbon trading price is the decisive factor that affects the sustainability level, enterprises' profit, and carbon emission; the carbon cap has a positive regulatory effect on profit.

## 1. Introduction

In recent years, global warming and climate change have created increasing awareness of environmental issues among people [1]. Moreover, carbon emission is regarded as the main contributor to global warming and climate change [2]. Due to the frequent occurrences of natural disasters, many countries and regional organizations have passed carbon emission regulations to prevent enterprises from excessively discharging emissions into the air [3]. In general, governments often adopt three policies, namely, carbon cap, tax, and cap-and-trade policies, to reduce carbon emission [4]. For example, the US Congress carries out a carbon cap policy [5]. Meanwhile, the carbon tax policy is adopted in Denmark, Japan, Ireland, and Finland, and the cap-and-trade policy is adopted in Norway, Switzerland, Sweden, Italy, Slovenia, UK, USA, Canada, and China [3, 6]. At the same time, the public environmental consciousness and social responsibility are increasing with more frequent occurrence of extreme weather events [7], and sustainable consumption is becoming more and more popular all over the world. The idea of carbon label is used to identify the sustainability level of a product. For instance, Walmart has requested its 100,000 suppliers to complete the carbon footprint verification and labeled their products with colors according to the carbon footprints [8]. The sustainability level of the product enables environmental-conscious consumers to select products with the smallest carbon footprints. The higher the consumer environmental consciousness is, the more the customers are willing to accept sustainable

products even with a higher price [9, 10]. The European Commission surveys show that 83% of the Europeans express concern about the carbon emission of products when buying them [11]. The Environmental Protection Ministry of China has launched a pilot project to arouse the public's environmental consciousness by attaching carbon labels and certifying sustainable products [9]. Enhanced environmental awareness means familiarity with the public with the ideas of carbon reduction and sustainable development [3].

In this context, the dual effects of governments and the public cause more stressful and challenging enterprise operations. With the increasing consumer environmental awareness, sustainable technology is an effective way for enterprises to achieve a competitive and commercial advantage [8] and enhance environmental sustainability [12]. Sustainable investment can cut down the carbon emission in the supply chain [13] and improve environmental sustainability and enterprises' competitiveness in the long run [14]. However, enterprises should invest in sustainable operations when adopting sustainable technology, which may lead to a change in their cost structure [8]. Hence, determining an appropriate level of sustainability is a primary concern for enterprises. To meet governments' carbon regulations, enterprises have to adjust their operations' objective from economy to environment. By nature, emission from supply chain operations exceeds 20% of the total global emission [5], and production, inventory, and transportation activities contribute to sustainability problems in enterprises' operations [15]. In particular, inventory activity is responsible for 11% of the carbon emission from the logistics sector. For example, by optimizing inventory operations, Hewlett–Packard decreased its carbon emission from 26.1 tonnes to 18.3 tonnes in 2010 [4]. Moreover, transportation accounts for around 5% of the world's carbon emission [16], which is considered one of the principal sources of carbon emission [15]. The carbon emission generated from the inventory and transportation process is mainly determined by inventory control decisions and transportation scheduling [17]. The interaction of trade-off between inventory and transportation indicates that their integrated optimization is needed to reduce costs and carbon emission [18]. When enterprises adopt sustainable technology, market demand and cost structure will change according to the product's sustainability level. These changes will influence the inventory control decision and transportation scheduling, thereby influencing costs and carbon emission in logistics. In turn, the inventory-transportation scheduling will affect the marginal cost of sustainable investment, thus affecting the decision of sustainability level. Therefore, enterprises need joint decisions on sustainability level and inventory-transportation solutions to satisfy the carbon emission requirements set by the governments.

As far as our knowledge from the literature review, a few studies have attempted to optimize sustainability level and inventory-transportation scheduling simultaneously. For instance, Toptal et al. [19] and Huang et al. [3] developed EOQ models with sustainable investment. However, they did not consider consumer environmental awareness; that is, the demand in their models was constant and did not vary with the sustainability level. In this study, inventory-transportation models that consider consumer environmental awareness and sustainable investment under carbon cap, tax, and cap-and-trade policies were investigated. Furthermore, a comprehensive fuel consumption function is integrated into these models to calculate the costs and emissions in transportation, which can improve reliability and applicability in optimization [20]. We focus on investigating the following questions:

(1) How do enterprises decide sustainability level and inventory-transportation scheduling simultaneously under carbon cap, tax, and cap-and-trade policies?

(2) How do carbon cap, tax, and cap-and-trade policies affect enterprises' sustainability level, profit, and emission?

(3) How do governments set policy parameters to harmonize economic and environmental objectives?

The rest of this paper is organized as follows. The relevant literature is presented in Section 2. Model assumptions and notations are proposed in Section 3. The optimization models of the integrated sustainability level and inventory-transportation problem under carbon cap, tax, and cap-and-trade policies are explored and solved using the simulation solution method in Section 4. Then, the results of simulations and the effect of carbon policies on sustainability level and enterprises' performance are presented in Section 5. Policy parameters set by governments are estimated and discussed in Section 6. Finally, a conclusion is presented in Section 7.

## 2. Literature Review

Integrated inventory-transportation model incorporates inventory and transportation decisions simultaneously because of their trade-off. Earlier models aimed to maximize the overall total costs of inventory and transportation [21] without taking into account carbon emission. The classical economic order quantity (EOQ) was the first integrated inventory-transportation model introduced by Ford W. Harris in 1913. Since then, increasing numbers of scholars have extended the integrated inventory-transportation model in several ways. In integrated inventory-transportation models, they analyzed different demand functions, such as constant [3, 19, 22], random [20, 23], linear [24], quadratic [25], and time-varying demands [1]. The logistics network has been expanded from a single-single type to single-many [26], many-single [27], and many-many forms [20, 23]. For the decision level, the existing research mainly focuses on the tactical level problem [20, 23, 28–30], with few researchers considering strategic level problem [31]. In terms of the product, the existing research can be divided into the following categories: single product [5] and multiple products [20, 32]. Some researchers also consider other characters in the integrated inventory-transportation model. For example, Alım and Beullens [33] integrated a flexible delivery option into the inventory-transportation model for an online sales firm. Meanwhile,

Gautam et al. [34] jointly optimize the number of shipments and quantities of orders with defect management.

Following the sustainable development, the amount of literature on the integrated inventory-transportation problem considering environmental factors has increased rapidly in recent years. Benjaafar et al. [18] examined a simplified inventory model to explore the impacts of operations decisions on carbon emission. Meanwhile, Soysal et al. [23] explored an inventory routing problem with a comprehensive emissions model in transportation. The results indicated that horizontal collaboration decreases the costs and emissions in logistics. Biuki et al. [35] integrated the economic, ecological, and societal aspects into a location-inventory routing model. Moreover, Bouchery, et al. [16] presented an EOQ model that considers vehicle capacities to provide sufficient conditions that ensure a decrease in costs and carbon emission. The joint decision on inventory and transportation under carbon policies is a hot research topic. Considering carbon tax policy, Wang et al. [30] developed an inventory-transportation model in refined oil logistics. Xu et al. [36] constructed nonlinear models to optimize inventory and transportation strategy for perishable items under carbon tax and carbon cap-and-trade regulations. Meanwhile, Micheli and Mantella [20] extended the model of Soysal et al. [23] with a heterogeneous fleet under carbon cap, carbon tax, and carbon cap-and-trade regulations; they ignored the emissions associated with inventory. Moreover, Tang et al. [17] examined the effect of controlling carbon emission in inventory-transportation management with stochastic demand. They analyzed three carbon regulations, namely, carbon tax, cap-and-trade, and carbon offset. Under the cap-and-trade scheme, Hua et al. [37] extended the classical EOQ model with carbon emission and proved that the optimal order quantity is between the classical EOQ model and the model that minimizes carbon emission. Furthermore, Konur and Schaefer [5] investigated the EOQ model with less-than-truckload and truckload transportation under carbon tax and cap, cap-and-trade, and cap-and-offset regulations. Chen et al. [38] used EOQ models under various environmental regulations to illustrate the conditions where emissions may be reduced and the relative reduction in emissions is greater than the relative increase in cost. Meanwhile, Liao and Deng [39] extended an EOQ model with uncertain demand under carbon tax regulation, and the result showed that increasing carbon tax will decrease profit margins and alter the optimal order decisions. Finally, Rabta [29] presented an EOQ model in a circular economy and proposed various relationships (linear and nonlinear) between the circularity level and demand, and cost and selling price.

In the context of a sustainable economy, consumer environmental awareness is integrated into a supply chain optimization model. Yu et al. [40] developed an optimization model under oligopolistic competition, and their results show that manufacturers could promote a product's sustainability level due to an increase in consumer environmental awareness. Moreover, to analyze the effect of carbon tax price on carbon emission, Hovelaque and Bironneau [28] explored an EOQ model with demand dependent on price and carbon emission in production and logistics activities. Meanwhile, Zhang et al. [41] investigated the effect of consumer environmental awareness on channel coordination and order quantities. Cheng et al. [8] integrated carbon-labeling scheme into game-theoretic models between a manufacturer and a retailer to investigate the impact of consumer environmental awareness on supply chain performance. Consumer environmental awareness increases market demand or sale price, which drives manufacturers to adopt clean technology and thus increase the sustainability level of product. Hence, integrating clean technology or sustainable investment into supply chain activities has also become a research hotspot. Drake et al. [42] addressed the technology choice problem under carbon tax and cap-and-trade policies. Their results revealed that the expected profit under the cap-and-trade policy is greater than that under carbon tax policy. Meanwhile, Tao and Xu [43] examined an EOQ model to investigate the effect of consumer environmental awareness on optimal order quantity, emission level, and total costs under carbon tax and carbon cap-and-trade regulations. Toptal et al. [19] investigated an EOQ model considering green technology under carbon tax, cap, and cap-and-trade policies. The results revealed that green technology can simultaneously reduce costs and carbon emissions. Moreover, by extending the model, Huang et al. [3] developed green technology to determine the green investment amount, delivery quantity, and optimal production quantity under the same regulations. Dong et al. [13] integrated sustainable investment into order quantity decision with stochastic demand under the carbon cap-and-trade policy. They indicated that sustainable investment has a major impact on the performance of supply chain. Under a carbon tax regulation, Cheng et al. [8] explored a sustainable investment decision-making model using Bayesian information updating. Furthermore, to explore the effects of the government subsidy coefficient on the sustainability level and the retail price, Su et al. [44] examined a green supply chain model under different government subsidies.

This study points out three research gaps on this topic: (1) studies extending classical EOQ models by considering sustainable investment and consumer environmental awareness are scarce. Tao and Xu [43] only considered consumer environmental awareness, whereas Toptal et al. [19] and Huang et al. [3] only considered sustainable investment. Meanwhile, Dong et al. [13], Cheng et al. [10], and Su et al. [44] considered them both; however, they did not consider inventory and transportation activities in their models. (2) In those EOQ models, transportation cost and emission are assumed to have fixed values or piecewise functions, giving a less accurate estimation of transportation cost and emission. (3) When considering carbon policies, the literature focuses on the sensitivity analysis of policy parameters. There are few studies on deciding appropriate policy parameters from the government perspective.

To conclude, our study adds to the literature on the integrated inventory-transportation model by (1) extending EOQ model with sustainable investment and consumer environmental awareness simultaneously under different carbon emission regulations (i.e., cap, tax, and cap-and-

trade) and (2) employing a comprehensive fuel consumption function, on the basis of factors such as vehicle type, vehicle load, vehicle speed, and traveled distance, to compute transportation cost and emission. The explicit consideration of fuel consumption ensures a more accurate estimation of transportation cost and emission [20, 23]. This study also contributes to the literature by (3) developing regression models to analyze the effect of policy parameters on sustainability level, profit, and emission, and estimate optimal policy parameters based on maximizing social welfare. Table 1 presents the literature positioning of the present paper.

## 3. Model Assumptions and Notations

We consider an enterprise that manufactures a sustainable item and sells the product on its own salespoint. The enterprise determines the sustainability level and controls the inventory and transportation for the item. The main forces driving sustainable investment are governments' carbon policies and consumers' environmental awareness. The market demand is dependent on the sustainability level, which affects the inventory control and transportation decision. The enterprise needs joint decisions on sustainability level and EOQ to maximize its profit and satisfy carbon policies. We consider three carbon regulations: cap, tax, and cap-and-trade. The major notations used in the models are summarized in Table 2. The models are developed under the following assumptions:

(1) The enterprise assumes the basic EOQ settings: the demand is uniform and continuous; no shortage is allowed; the order lead time is known and constant; the inventory replenishment is completed instantaneously; the ordering cost per time is constant and independent of order quantity; and the holding cost is a linear function of inventory [43]. In addition, the production cost of each unit is constant and independent of the sustainability level.

(2) We assume that the sustainability level only describes the carbon emission in the manufacturing process but does not include the carbon emission in transportation and inventory. Sustainability level $S$ is a dimensionless indicator ranging from 0 to 1 [29]. The sustainability level $S$ is determined by the manufacturer and measured by equation (1), where $e_0$ is the carbon emission in production, $a$ is the base carbon emission per unit when $S = 0$, and $b$ is the minimum carbon emission when $S = 1$:

$$S = \frac{a - e_0}{a - b}. \tag{1}$$

(3) We assume that the sustainable investment cost is a quadratic function [10, 13]; that is, $\delta S^2/2$, where $\delta$ is the sustainable investment coefficient. The cost and cost growth rate of sustainable investment increase monotonously with the sustainability level.

(4) We assume a linear demand function affected by the sustainability level; that is, $D = D_0 + \rho S$, where $\rho$ is the coefficient of the sustainable effect on the increasing demand [29] and $D_0$ is the potential demand per year when $S = 0$, or $\rho = 0$.

(5) We assume a limited, capacitated, and homogeneous fleet. Moreover, to compute the cost and carbon emission in transportation, we assume comprehensive fuel consumption [20, 23]. Given a traveled distance $d$ and a vehicle speed $f$, the fuel consumption, denoted by FC, is computed by equation (2), where $\xi$ is fuel-to-air mass ratio, $k_e$ is engine friction factor, $N_e$ is engine speed, $V_e$ is engine displacement, $\varpi$ is efficiency parameter for diesel engines, $A_f$ is frontal surface area, $\varepsilon$ is vehicle drive train efficiency, $c_d$ is coefficient of aerodynamic drag, $\alpha$ is air density, $g$ is gravitational constant, $\varphi$ is road angle, $c_r$ is coefficient of rolling resistance, $\mu$ is curtweight, $\kappa$ is heating value of a typical diesel fuel, and $V$ is payload:

$$\mathrm{FC} = \xi\left(\frac{\left(k_e N_e V_e d/f\right) + \left(0.5 c_d \alpha A_f d f^2/(1000\varpi\varepsilon)\right) + dg\left(\left(\sin\varphi + c_r\cos\varphi\right)(\mu + V)/(1000\varpi\varepsilon)\right)}{(\kappa\psi)}\right). \tag{2}$$

Let $\lambda = (\xi/(\kappa\psi))$, $y = k_e N_e V_e$, $\gamma = (1/(1000\varpi\varepsilon))$, $B = 0.5 c_d \alpha A_f$, and $\sigma = g(\sin\varphi + c_r\cos\varphi)$; then $FC = \lambda((y\mathrm{d}/f) + \mathrm{d}\gamma B f^2 + \mathrm{d}\gamma\sigma(\mu + V))$. Let $\tau_1 = (\lambda \mathrm{d} y/f) + \lambda \mathrm{d}\gamma(Bf^2 + \mu\sigma)$ and $\tau_2 = \lambda \mathrm{d}\gamma\sigma$; then the vehicle transportation cost is $\vartheta k\tau_1 + \vartheta\tau_2 Q$, where $k$ is the number of vehicles, $Q$ is order quantity, and $\vartheta$ is fuel price. The carbon emission in transportation is $\theta k\tau_1 + \theta\tau_2 Q$, where $\theta$ denotes fuel conversion factor.

## 4. Optimization Model

*4.1. Base Case Model ($M_0$).* The base case model, denoted by $M_0$, represents the case where there is no carbon policy. The enterprise's goal is to maximize profits. Without any carbon policy in place, the average annual profit is given by equation (3), where $k = \lceil Q/L \rceil$. In equation (3), the first, second, third, fourth, fifth, and sixth terms are sales income, manufacturing cost, sustainable investment cost, inventory

Table 1: The literature positioning of this paper.

| Papers | Sustainable investment | Consumer environmental awareness | Comprehensive fuel consumption | Carbon emission regulation | | | Decision on policy parameters |
|---|---|---|---|---|---|---|---|
| | | | | Cap | Tax | Cap-and-trade | |
| Konur and Schaefer [5], Tang et al. | | | | √ | √ | √ | |
| Dong et al. [13] | √ | √ | | | | √ | |
| Soysal et al. [23] | | | √ | | | | |
| Micheli and Mantella [20] | | | √ | √ | √ | √ | |
| Tao and Xu [43] | | √ | | √ | √ | √ | |
| Toptal et al. [19], Huang et al. [3] | √ | | | √ | √ | √ | |
| Cheng et al. [8], Su et al. [44] | √ | √ | | | | | |
| This paper | √ | √ | √ | √ | √ | √ | √ |

Table 2: Main notations.

| Parameters | Meaning |
|---|---|
| $A$ | Fixed cost of inventory replenishment |
| $h$ | Holding cost per unit per year held in inventory |
| $c_0$ | Manufacturing cost per unit |
| $c_t$ | Transportation cost, $c_t = \vartheta k\tau_1 + \vartheta\tau_2 Q$ |
| $L$ | Vehicle capacity |
| $P$ | Sale price |
| $D_0$ | Potential demand per year when $S = 0$ |
| $\rho$ | Coefficient of the effect of sustainability on demand |
| $\delta$ | Coefficient of sustainable investment |
| $D$ | Actual demand per year, $D = D_0 + \rho S$ |
| $\widehat{A}$ | Carbon emission amount due to inventory replenishment |
| $\widehat{h}$ | Carbon emission amount due to holding a unit inventory per year |
| $\widehat{c_t}$ | Carbon emission amount due to transportation, $\widehat{c}_t = \theta k\tau_1 + \theta h\tau_2 Q$ |
| $n$ | Carbon policy index: $n = 0$ for no carbon, $n = 1$ for cap, $n = 2$ for tax, and $n = 3$ for cap-and-trade policies |
| $C_a$ | Carbon emission cap per year |
| $C_p$ | Carbon emission trading price per ton |
| $C_T$ | Carbon tax price per ton |
| $SW$ | Social welfare |
| $U_{ec}$ | Economic utility |
| $U_{en}$ | Environmental utility |
| *Decision variables* | |
| $Q$ | Order quantity |
| S | Sustainability level |
| *Functions* | |
| $E(Q,S)$ | Average annual carbon emission function |
| $Z(Q,S)$ | Average annual profit function |

replenishment cost, inventory holding cost, and transportation cost, respectively:

$$Z_0(Q,S) = P(D_0 + \rho S) - c_0(D_0 + \rho S) - \frac{\delta S^2}{2} - \frac{A(D_0 + \rho S)}{Q} - \frac{hQ}{2} - \left(\frac{k\vartheta\tau_1(D_0 + \rho S)}{Q} + \vartheta\tau_2(D_0 + \rho S)\right), \tag{3}$$

where $Z_0(Q,S)$ is a discontinuous function, and the optimal solution $(Q,S)$ cannot be directly obtained by the first-order partial derivative. We design a two-stage optimization method to solve the problem. First, we assume that the sustainability level is given by the enterprise, under which we solve the optimal order quantity problem. Second, we obtain the optimal $S$ by the simulation method. In the following analysis, the optimal order quantity problems under different carbon regulations are solved in Sections 4.1–4.4. Meanwhile, the simulation method for solving the optimal $S$ is analyzed in Section 4.5.

Given S, the optimal EOQ $Q_0^*$ that maximizes $Z_0(Q,S)$ also cannot be determined using the first-order derivative. $Z_0(Q,S)$, especially, is a piecewise continuous function such that each piece is in the form of equation (2) over a given quantity range of length $L$. Obviously, the sum of the first five terms of $Z_0(Q,S)$ is an EOQ-type convex function of $Q$ with a maximum at $q_{\text{eoq}} = \sqrt{(2A(D_0+\rho S)/h)}$. For each fixed positive integer $k$, the sixth term is a decreasing convex function of $Q$ over $(k-1)L < Q \le kL$. Based on these characteristics of $Z_0(Q,S)$, the following properties are verified [22].

*Property 1.* Let $Q_0^k = \sqrt{(2(A+k\vartheta\tau_1)(D_0+\rho S)/h)}$, $k = 1, 2, \ldots$ If $(k-1)L < Q_0^k \le kL$, $Q_0^k$ is realizable.

*Property 2.* Define $i$ to be the unique integer such that $(i-1)L < q_{\text{eoq}} \le iL$. For all $k \le i-1$, $Z_0(Q,S)$ is increasing over $(k-1)L < Q \le kL$ and $Q_0^k$ is not realizable.

*Property 3.* If $k \ge i$, then $Z_0(Q,S) \le Z_0(kL,S)$ for $Q \ge kL$.

*Property 4.* If $Q_0^i \ge iL$, then $Z_0(Q,S)$ is increasing over $(i-1)L < Q \le iL$. If $Q_0^i < iL$, then $Z_0(Q,S)$ is increasing over $(i-1)L < Q \le Q_0^i$ and decreasing over $Q_0^i < Q \le iL$.

From Properties 1–4, the optimal EOQ $Q_0^*$ is defined by the following corollary.

**Corollary 1.** *Given the sustainability level $S$,* $Q_0^* = \arg\max\{Z(\min\{Q_0^{i+1}, (i+1)L\}, S), Z(iL,S)\}$. *$Q_0^*$ can be determined by the following algorithm:*

*Step 1. Compute* $q_{eoq} = \sqrt{(2A(D_0+\rho S)/h)}$ *and* $i = \lceil q_{eoq}/L \rceil$.

*Step 2. Compute* $Q_0^i = \sqrt{(2(A+i\vartheta\tau_1)(D_0+\rho S)/h)}$. *If* $Q_0^i \ge iL$, *let* $Q_0^i = iL$.

*Step 3.Compute and compare* $Z(Q_0^i, S)$ *and* $Z_0((i-1)L, S)$. *Select the one that yields the maximum profit as the optimal* $Q_0^*$, *and stop.*

*4.2. Model under Carbon Cap Regulation ($M_1$).* Under a cap regulation, the enterprise is subject to an upper bound on the total average annual carbon emission. The enterprise's problem is to find the sustainability level of production and the optimal order quantity to maximize the average annual total profit without exceeding the emission cap $C_a$. This problem can be formulated as follows:

$$
\begin{aligned}
(M1):\ \max \quad & Z_1(Q,S) = (P - c_0 - \vartheta\tau_2)(D_0+\rho S) - \frac{\delta S^2}{2} - \frac{(A+k\vartheta\tau_1)(D_0+\rho S)}{Q} - \frac{hQ}{2} \\
\text{s.t.} \quad & E(Q,S) = \frac{(\widehat{A}+k\theta\tau_1)D_0+\rho S}{Q} + \frac{\widehat{h}Q}{2} + (a+(b-a)S+\theta\tau_2)(D_0+\rho S) \le C_a \\
& Q \ge 0.
\end{aligned}
\tag{4}
$$

$Z_1(Q,S)$ and $E(Q,S)$ are discontinuous functions and should be analyzed simultaneously to find the optimal solution to $M1$, which is denoted by $Q_1^*$. Feasible solutions exist for M1 when the minimum of $E(Q,S)$ is lager than $C_a$. Similar to $Z_1(Q,S)$, each piece of the $E(Q,S)$ function is an EOQ type. According to Corollary 1, the optimal emission order quantity $\widehat{Q}^*$ and the least emission $\widehat{E}^*$ can be determined by the following algorithm [22, 45]:

*Step 1.* Compute $\widehat{q}_{\text{eoq}} = \sqrt{(2\widehat{A}(D_0+\rho S)/\widehat{h})}$ and $i = \lceil \widehat{q}_{\text{eoq}}/L \rceil$.

*Step 2.* Compute $\widehat{Q}^i = \sqrt{(2(\widehat{A}+i\vartheta\tau_1)(D_0+\rho S)/\widehat{h})}$. If $\widehat{Q}^i \ge iL$, let $\widehat{Q}^i = iL$.

*Step 3.* Compute and compare $E(\widehat{Q}^i, S)$ and $E((i-1)L, S)$. If $E(\widehat{Q}^i, S) \le E((i-1)L, S)$, $\widehat{Q}^* = \widehat{Q}^i$, $\widehat{E}^* = E(\widehat{Q}^i, S)$; else, $\widehat{Q}^* = (i-1)L$, $\widehat{E}^* = E((i-1)L, S)$, and stop.

We assume that $C_a \ge \widehat{E}^*$; that is, feasible order quantities exist for M1. Given S, the feasible solution consists of all pairs $(Q_1, S)$ such that $q_1^i \le Q_1 \le q_2^i$, where

$$
q_1^i = \frac{C_a - (a+(b-a)S+\theta\tau_2)(D_0+\rho S) - \sqrt{(C_a - (a+(b-a)S+\theta\tau_2)(D_0+\rho S))^2 - 2\widehat{h}(\widehat{A}+i\theta\tau_1)(D_0+\rho S)}}{\widehat{h}}. \tag{5}
$$

$$
q_2^i = \frac{C_a - (a+(b-a)S+\theta\tau_2)(D_0+\rho S) + \sqrt{(C_a - (a+(b-a)S+\theta\tau_2)(D_0+\rho S))^2 - 2\widehat{h}(\widehat{A}+i\theta\tau_1)(D_0+\rho S)}}{\widehat{h}}, \tag{6}
$$

where $q_1^i$ and $q_2^i$ are the two roots of $(\widehat{A} + k\theta\tau_1)(D_0 + \rho S)/Q + \widehat{h}Q/2 + (a + (b-a)S + \theta\tau_2)(D_0 + \rho S) = C_a$. Now, consider the range $((i-1)L, iL]$. When $q_2^i \le (i-1)L$ or $q_1^i > iL$, there is no feasible solution in the range $((i-1)L, iL]$. Otherwise, the feasible range of $Q_1$ is $[Q_1^i, Q_2^i]$, where $Q_1^i = \max\{q_1^i, (i-1)L\}$ and $Q_2^i = \min\{q_2^i, iL\}$. It is easy to verify that $q_1^1 < q_1^2 < \cdots < q_1^n < q_2^n < q_2^{n-1} < \cdots < q_2^1$, and the length of feasible region is decreasing in the number of vehicles $i$. Let $m_1$ and $m_2$ be defined as the minimum and the maximum number of vehicles, respectively, such that M1 is feasible. $m_1$ and $m_2$ can be easily determined using the relation $q_1^1 < q_1^2 < \cdots < q_1^n < q_2^n < q_2^{n-1} < \cdots < q_2^1$. By the definitions of $m_1$ and $m_2$, it follows that $Q_1 \in [Q_1^{m_1}, Q_2^{m_2}]$ [5].

Recall form Property 2 that $i$ is the unique integer, such that $(i-1)L < q_{\text{eoq}} \le iL$. The following corollary summarizes the optimal solution for M1 [5].

**Corollary 2.** *Given the sustainability level $S$ and supposing that M1 is feasible,*

*(1) if $m_1 \ge i$, then $Q_1^* = Q_0^i$ if $Q_0^i \in [Q_1^{m_1}, Q_2^{m_2}]$; $Q_1^* = Q_1^{m_1}$ if $Q_0^i < Q_1^{m_1}$; and $Q_1^* = Q_2^{m_1}$ if $Q_0^i > Q_2^{m_1}$*

*(2) if $m_2 \le i-1$, then $Q_1^* = Q_2^{m_1}$, if $m_1 = m_2$; and $Q_1^* = \arg\max\{Z_1(Q_2^{m_2-1}, S), Z_1(Q_2^{m_2}, S)\}$ if $m_1 \ne m_2$*

*(3) if $m_1 \le i-1 < i \le m_2$, then $Q_1^* = \arg\max\{Z_1((i-1)L, S), Z_1(\min\{Q_0^i, Q_2^i\}, S)\}$*

*From Corollary 2, $Q_1^*$ can be determined by the following algorithm:*

*Step 1. Compute $q_{eoq} = \sqrt{(2A(D_0 + \rho S)/h)}$ and $i = \lceil q_{eoq}/L \rceil$.*

*Step 2. Compute $Q_0^i = \sqrt{(2(A + i\vartheta\tau_1)(D_0 + \rho S)/h)}$.*

*Step 3. Let $j = 1, 2, \ldots$, and set $i = j$ in equations (5) and (6) to compute $q_1^j$ and $q_2^j$. Let $m_1$ be defined as the minimum integer such that $[Q_1^j, Q_2^j] \ne \emptyset$, and let $m_2$ be defined as the maximum integer such that $[Q_1^j, Q_2^j] \ne \emptyset$.*

*Step 4. Set $Q_1^{m_1} = \max\{q_1^{m_1}, (m_1 - 1)L\}$, $Q_1^{m_2} = \max\{q_1^{m_2}, (m_2 - 1)L\}$, $Q_2^{m_1} = \min\{q_2^{m_1}, m_1 L\}$, and $Q_2^{m_2} = \min\{q_2^{m_2}, m_2 L\}$. If $m_1 \ge i$, go to Step 5. If $m_2 \le i - 1$, go to Step 6. Otherwise, go to Step 7.*

*Step 5. If $Q_1^{m_1} > Q_0^i$, $Q_1^* = Q_1^{m_1}$. If $Q_2^{m_1} < Q_0^i$, $Q_1^* = Q_2^{m_1}$. Otherwise, $Q_1^* = Q_0^i$ and stop.*

*Step 6. If $m_1 = m_2$, $Q_1^* = Q_2^{m_1}$. Otherwise, compute the profit for $Q = Q_2^{m_2-1}$ and $Q = Q_2^{m_2}$. Select the one that yields the maximum profit as the optimal $Q_1^*$, and stop.*

*Step 7. Let $Q_1' = \min\{Q_0^i, Q_2^i\}$. Compute the profit for $Q = Q_1'$ and $Q = (i-1)L$. Select the one that yields the maximum profit as the optimal $Q_1^*$, and stop.*

*4.3. Model under Carbon Tax Regulation ($M_2$).* Under a tax regulation, the enterprise pays $C_T$ monetary units in taxes for unit carbon emission. The enterprise's objective is to maximize average annual total profit without restriction on the maximum carbon emission and its problem can be formulated as follows:

$$
\begin{aligned}
(M2):\ \max \quad & Z_2(Q,S) = (P - c_0 - \vartheta\tau_2)(D_0 + \rho S) - \frac{\delta S^2}{2} - \frac{(A + k\vartheta\tau_1)(D_0 + \rho S)}{Q} - \frac{hQ}{2} - C_T E(Q,S) \\
\text{s.t.} \quad & E(Q,S) = \frac{(\widehat{A} + k\theta\tau_1)(D_0 + \rho S)}{Q} + \frac{\widehat{h}Q}{2} + (a + (b-a)S + \theta\tau_2)(D_0 + \rho S) \\
& Q \ge 0.
\end{aligned} \tag{7}
$$

Substituting $E(Q,S)$ into $Z_2(Q,S)$, then we have

$$
\begin{aligned}
Z_2(Q,S) = {} & (P - c_0 - \tau_2(\vartheta + C_T\theta) \\
& - C_T(a + (b-a)S))(D_0 + \rho S) - \frac{\delta S^2}{2} \\
& - \frac{(A + C_T\widehat{A} + k\tau_1(\vartheta + C_T\theta))(D_0 + \rho S)}{Q} \\
& - \frac{(h + C_T\widehat{h})Q}{2}.
\end{aligned} \tag{8}
$$

Equation (8) indicates that $Z_2(Q,S)$ has the same function structure as $Z_0(Q,S)$. Hence, the following corollary is obtained from Properties 1–4 to find the optimal EOQ, which is denoted by $Q_2^*$.

**Corollary 3.** *Given the sustainability level $S$, $Q_2^* = \arg\max\{Z(\min\{Q_2^i, iL\}, S), Z((i-1)L, S)\}$. $Q_2^*$ can be determined by the following algorithm:*

*Step 1. Compute $q_{2eoq} = \sqrt{((2(A + C_T\widehat{A})(D_0 + \rho S))/(h + C_T\widehat{h}))}$ and $i = \lceil q_{2eoq}/L \rceil$.*

*Step 2. Compute $Q_2^i = \sqrt{((2(A + C_T\widehat{A} + i(\vartheta + C_T\theta)\tau_1)(D_0 + \rho S))/(h + C_T\widehat{h}))}$. If $Q_2^i \ge iL$, let $Q_2^i = iL$.*

*Step 3. Compute and compare $Z_2(Q_2^i, S)$ and $Z_2((i-1)L, S)$. Select the one that yields the maximum profit as the optimal $Q_2^*$, and stop.*

*4.4. Model under Carbon Cap-and-Trade Regulation* ($M_3$). Under a cap-and-trade regulation, the enterprise is subject to an emission cap on the total carbon emission. If the carbon emission is more than or lower than the cap, the enterprise can buy carbon emission permits or sell extracarbon emission with a carbon emission trading system [19]. Supply and demand are assumed to be always available for buying and selling carbon emissions. The enterprise's problem of deciding the optimal order quantity is formulated as follows:

$$(M3):\ \max \quad Z_3(Q,S) = (P - c_0 - \vartheta\tau_2)(D_0 + \rho S) - \frac{\delta S^2}{2} - \frac{(A + k\vartheta\tau_1)(D_0 + \rho S)}{Q} - \frac{hQ}{2} - C_p x. \tag{9}$$

$$\text{s.t.} \quad E(Q,S) = \frac{(\widehat{A} + k\theta\tau_1)(D_0 + \rho S)}{Q} + \frac{\widehat{h}Q}{2} + (a + (b-a)S + \theta\tau_2)(D_0 + \rho S). \tag{10}$$

$$x = E(Q,S) - C_a. \tag{11}$$

$$Q \geq 0. \tag{12}$$

The objective function $Z_3(Q,S)$ is translated into equation (13) by substituting $E(Q,S)$ and $x$ into equation (9):

$$\begin{aligned} Z_3(Q,S) = {} & \left(P - c_0 - \tau_2\left(\vartheta + C_p\theta\right)\right. \\ & \left. - C_p(a + (b-a)S)\right)(D_0 + \rho S) + C_p C_a - \frac{\delta S^2}{2} \\ & - \frac{\left(A + C_p\widehat{A} + k\tau_1\left(\vartheta + C_p\theta\right)\right)(D_0 + \rho S)}{Q} \\ & - \frac{\left(h + C_p\widehat{h}\right)Q}{2}. \end{aligned} \tag{13}$$

Note that $Z_3(Q,S)$ follows a similar functional form with $Z_0(Q,S)$ and $Z_2(Q,S)$. From Properties 1–4, the optimal EOQ under carbon cap-and-trade regulation, denoted by $Q_3^*$, can be determined by the following corollary.

**Corollary 4.** *Given the sustainability level $S$, $Q_3^* = \arg\max\{Z(\min\{Q_3^i, iL\}, S), Z((i-1)L, S)\}$. $Q_3^*$ can be determined by the following algorithm:*

*Step 1. Compute* $q_{3eoq} = \sqrt{((2(A + C_p\widehat{A})(D_0 + \rho S))/(h + C_p\widehat{h}))}$ *and* $i = (q_{3eoq}/L)$.

*Step 2. Compute* $Q_3^i = \sqrt{((2(A + C_p\widehat{A} + i(\vartheta + C_p\theta)\tau_1)(D_0 + \rho S))/(h + C_p\widehat{h}))}$.

*If $Q_3^i \geq iL$, let $Q_3^i = iL$.*

*Step 3. Compute and compare $Z_3(Q_3^i, S)$ and $Z_3((i-1)L, S)$. Select the one that yields the maximum profit as the optimal $Q_3^*$, and stop.*

*4.5. Simulation Method for Optimizing Sustainability Level.* Corollaries 1–4 show that the optimal order quantity is dependent on the sustainability level $S$ at any carbon policy case. The optimal sustainability level $S$ cannot be deduced directly. A simulation method is used to seek for the optimal sustainability level $S$ by transforming the continuous variable S into a discrete variable. Considering $0 \leq S \leq 1$, and setting the optimization precision error to 0.001, 1001 feasible solutions exist for $S$, that is, 0, 0.001, 0.002, . . ., 0.999, 1. At any carbon policy case, the simulation process consists of nine steps, as shown in Figure 1.

*Step 1.* Set $j = 0$, $ZZ = 0$, $SS = 0$.

*Step 2.* Let $S(j) = j * 0.001$.

*Step 3.* Given $S(j)$, compute the EOQ using the algorithms of Corollaries 1–4 and obtain $Q(j)$.

*Step 4.* For the solution pair $(Q(j), S(j))$, compute the profit $Z(j)$ using $Z(Q,S)$.

*Step 5.* Compare $Z(j)$ and $ZZ$. If $Z(j) \geq ZZ$, go to Step 6. Otherwise, go to Step 7.

*Step 6.* Let $SS = S(j)$, $ZZ = Z(j)$.

*Step 7.* Let $j = j + 1$.

*Step 8.* If $j > 1000$, go to Step 9. Otherwise, go back to Step 2.

*Step 9.* Assign $SS$ to the optimal $S$, and stop.

Let us take the following as an example. Set $A$ =\$150, $h$ = \$3, $\widehat{A}$ = 200 kg, $\widehat{h}$ = 5 kg, $P$ = \$0.5/kg, $c_0$ = \$0.2/kg, $a$ = 0.4 kg, $b$ = 0.1 kg, $\vartheta$ = \$1.5/l, $\theta$ = 2.63 kg/l, $D_0$ = 250,000 kg, $\rho$ = 50,000 kg, $\delta$ = \$70,000, $f$ = 80 km/h, and $d$ = 500 km. Based on the model of Micheli and Mantella [20], the vehicle parameters are reported as follows: $\xi = 1$, $k_e$ = (0.25 kj/rev/l), $N_e$ = 38.3 rev/s, $V_e$ = 4.5 l, $\varpi = 0.45$, $A_f = 7\,\text{m}^2$, $\varepsilon = 0.45$, $c_d = 0.6$, $\alpha = 1.2041\,\text{kg/m}^3$, 9.81 m/s$^2$, $\varphi = 0$, $c_r = 0.01$, $\mu$ = 3,500 kg, $\kappa = 44$, and $L$ = 4,000 kg. Carbon policy parameters are assumed as follows: the carbon tax price and carbon trading price are \$200/ton, and the cap is 80% of allowed emissions with respect to the base case when the sustainable investment is not considered. Based on the simulation optimization process shown in Figure 1, the simulation calculations under

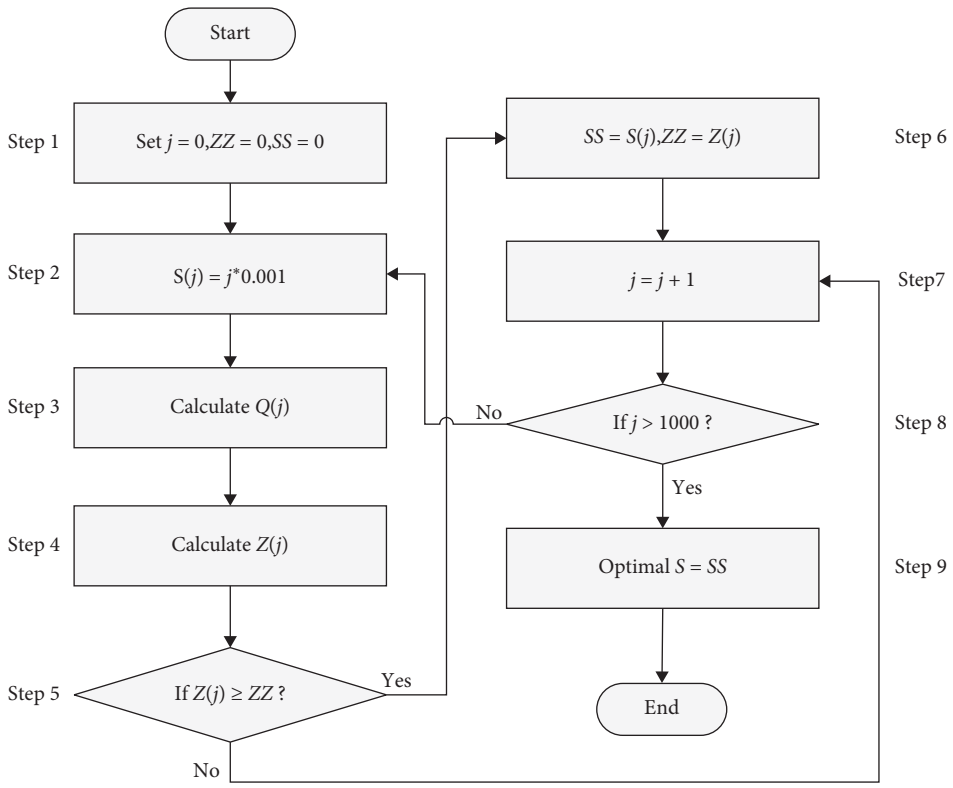



Figure 1: Simulation optimization process.

each carbon policy were programmed separately in MATLAB. The optimal sustainability level $S$, EOQ $Q$, profit $Z$, and carbon emission $E$ are shown in Table 3. The computation time is less than 0.5 s in all cases, implying the feasibility of the simulation optimization method.

Table 3: Simulation computation results of the example.

| Carbon policy | No policy | Cap | Tax | Cap-and-trade |
|---|---|---|---|---|
| $S$ | 0.139 | 0.281 | 0.316 | 0.331 |
| $Q$(kg) | 4,000 | 4,000 | 8,000 | 8,000 |
| $Z$(\$) | 46,971 | 43,348 | 13,362 | 44,816 |
| $E$(kg) | 170,767 | 130,413 | 139,371 | 131,840 |

## 5. Simulation Analysis

In this section, we focus on the simulation analysis to explore the effects of carbon emission regulations on sustainability level, profit, and carbon emission. In all of the problem instances, we assume that $d \sim U[100, 500]$ km and $f \sim U[20, 80]$ km/h, where $U[m, n]$ denotes a uniform distribution with bounds $m$ and $n$. The same vehicle parameters are adopted as in Section 4.5 [20]. The fuel price $\vartheta$ is assumed to be $U\$[1, 2]$/l. The emission conversion factor $\theta$ is set equal to 2.63 kg/l. The cost and carbon emission parameters are randomly generated assuming that $P \sim U\$[0.5, 0.6]$/kg, $c_0 \sim U\$[0.2, 0.3]$/kg, $a \sim U[0.4, 0.6]$ kg, $b \sim U[0.1, 0.2]$ kg. $h \sim U\$[1, 5]$/kg, $D_0 \sim U[100, 500]$ tonne, $A \sim U\$[50, 250]$, $h \sim U\$[2, 8]$, and $\widehat{A} \sim U[50, 300]$ kg (similar values are used by Konur and Schaefer [5]). The sustainable parameters $\rho$ and $\delta$ are assumed to be $U[0.1D_0, 0.3D_0]$ kg and $U\$[50000, 100000]$, respectively. All the above parameters randomly generate 1,000 different parameters and then generate 1,000 different situations through random combination. Based on the previous model solution method and the above data, the simulation method programmed in MATLAB is used to solve the optimal sustainability level, profit, and carbon emission under the three policies, namely, carbon cap, tax, and cap-and-trade.

*5.1. Effects of Carbon Cap under Carbon Cap Regulation.* Under the carbon cap policy, we use the model $M_0$ to calculate the carbon emission when the profit is maximized. Use this carbon emission as a benchmark, and set a carbon cap policy between 50% and 100% of this benchmark; that is, set the policy parameter to a carbon cap ratio. Let the carbon cap ratio be $C_{\text{cap}}$, and set 51 different $C_{\text{cap}}$ values, which are evenly generated from 0.5 to 1 using the LINSPACE function of MATLAB. At each $C_{\text{cap}}$, we analyze all 1,000 problem instances and, respectively, recorded their average values in terms of optimal sustainability level, carbon emission, and profit. The independent variable is $C_{\text{cap}}$, whereas the dependent variables are sustainability level, profit, and carbon emission. Three regression models are constructed as shown in equations (14)–(16), and the regression results are shown in Table 4:

$$S_1 = \alpha_{11} + \beta_{11} C_{\text{cap}} + \varepsilon. \quad (14)$$

$$E_1 = \alpha_{12} + \beta_{12} C_{\text{cap}} + \varepsilon. \quad (15)$$

$$Z_1 = \alpha_{13} + \beta_{13} C_{\text{cap}} + \varepsilon. \quad (16)$$

The regression results show that the sustainability level negatively correlates with the carbon cap ratio, whereas

carbon emission and profit positively correlate with the carbon cap ratio. The smaller the carbon cap ratio, the smaller the carbon cap. To meet the carbon cap requirements, enterprises must further increase the sustainability level. Increasing the sustainability level reduces carbon emission, whereas increasing the cost of sustainable investment reduces the profits. The simulation results show that when the carbon cap ratio increases by 1%, the sustainability level decreases by 0.01672, the carbon emission increases by 1,318.623 kg, and the profit increases by $200.797.

*5.2. Effects of Carbon Tax Price under Carbon Tax Regulation.* To analyze the effects of carbon tax price, we set 101 different $C_T$ values that are evenly generated from 0 to 1 using LINSPACE function of MATLAB. At each $C_T$, we analyze all 1,000 problem instances and, respectively, recorded their average values in terms of optimal sustainability level, carbon emission, and profit. Regression analyses are carried out to analyze the effects of carbon tax price $C_T$ on sustainability level, profit, and carbon emission. By using the data generated from simulation results under carbon tax regulation, we developed three regression models as shown in equations (17)–(19). Then, we present the regression results in Table 5:

$$S_2 = \alpha_{21} + \beta_{21} C_T + \varepsilon. \tag{17}$$

$$E_2 = \alpha_{22} + \beta_{22} C_T + \varepsilon. \tag{18}$$

$$Z_2 = \alpha_{23} + \beta_{23} C_T + \varepsilon. \tag{19}$$

The regression results show that the sustainability level significantly positively correlates with carbon tax price, whereas carbon emission and profit significantly negatively correlate with carbon tax price. When carbon tax price increases by $1/ton, the sustainability level will increase by 0.000714, but carbon emission and profit will decrease by 69.228 kg and $121.570, respectively. As the carbon tax raises the sustainability level, the cost of sustainable investment also increases. Although the increase in carbon tax prices can reduce carbon emission, it does not necessarily reduce the amount of carbon taxes. The economic income brought by the sustainability level cannot offset the increase of sustainable investment cost and carbon tax amount, causing the profit to decline.

*5.3. Effects of Carbon Cap and Trading Price under Carbon Cap-and-Trade Regulation.* Under the carbon cap-and-trade policy, we set carbon cap ratio to $C_{\text{cap}} \sim U[0.5, 1]$ (i.e., the same as the carbon cap policy) and carbon trading price to $C_p \sim U[0, 1]$. The carbon cap ratio and the carbon trading price randomly generate 1,001 random policy combinations in MATLAB. These 1,001 different policies are applied to 1,000 different situations, and the operations under different policies are simulated. By performing regression analysis on the simulation results, the regression model is obtained, as shown in equations (20)–(22). Moreover, the regression results are shown in Table 6:

$$S_3 = \alpha_{31} + \beta_{31} C_p + \varepsilon. \tag{20}$$

$$E_3 = \alpha_{32} + \beta_{32} C_p + \varepsilon. \tag{21}$$

$$Z_3 = \alpha_{33} + \beta_{33} C_p + \beta_{34} \left(C_p * C_{\text{cap}}\right) + \varepsilon. \tag{22}$$

The results show that $\beta_{31} > 0$, which indicates that the sustainability level is positively related to the carbon trading price. Meanwhile, the coefficient $\beta_{32} < 0$ indicates that carbon emission is inversely related to carbon trading price. Model (22) indicates that the relationship between profit and carbon trading price is affected by the carbon cap ratio; in particular, it has a positive regulatory effect on the relationship between carbon trading prices and profits. When $C_{\text{cap}}$ is greater (less) than 0.616, carbon trading price positively (negatively) correlates with profit. When $C_{\text{cap}} = 1$ and carbon trading price increases $1/tonne, the profit would increase by $76.762. By contrast, when $C_{\text{cap}} = 0.5$ and carbon trading price increases by $1/tonne, profits would decrease by $23.078. When $C_{\text{cap}} = 0.616$, the carbon trading price had no effect on profit. At this critical point, the carbon trading volume is 0, so no matter what the carbon trading price is, the carbon trading value (turnover) is still 0. Therefore, the carbon trading price will not have an impact on profit.

Models (20) and (21) indicate that no matter what the carbon price is, when carbon trading price increases by $1/tonne, the sustainability level will increase by 0.000715, whereas carbon emission will decrease by 69.466 kg. Comparing the results under the carbon tax, we find that carbon trading price has a similar impact on the sustainability level and carbon emission as carbon tax price. When $C_{\text{cap}} = 0$, carbon trading price positively correlates with profit. At this point, when the carbon trading price increases by $1/tonne, the profit decreases by $122.920. Considering the error caused by random factors in simulation, the carbon trading price and the carbon tax price also have the same effect on profit when $C_{\text{cap}} = 0$,. The above analysis results show that, under carbon cap-and-trade policy, carbon trading price plays a decisive role in sustainability level and carbon emission, which are dependent on carbon trading price and irrelevant to the carbon cap ratio. However, the carbon cap ratio will affect the enterprises' profit. The larger the carbon cap ratio, the more carbon emission enterprises can emit freely, which should be more beneficial to them. Therefore, when the carbon cap is 0, the effect of carbon trading policy is similar to that of the carbon tax policy [20]. Therefore, enterprises would choose to operate under the cap-and-trade policy involving incentives rather than under the carbon tax policy involving penalties when the carbon trading price is equal to the carbon tax price [3, 19].

## 6. Parameters Estimation of Carbon Policies and Discussion

The above results provide the exemplification of the effects of different carbon emission regulations on the performance of supply chain. In general, carbon policies have

Table 4: Regression results under carbon cap regulation.

| Variables | Model (14) $S$ | Model (15) $E$(kg) | Model (16) $Z_1$(\$) |
|---|---|---|---|
| $C_{CAP}$(standard error) | −1.672*** (0.031) | 131862.277*** (2044.752) | 20079.715*** (1130.034) |
| Constant (standard error)> | 1.802*** (0.021) | 49569.496*** (1562.821) | 46572.010*** (863.694) |
| Observations | 51 | 51 | 51 |
| $R$-squared | 0.987 | 0.988 | 0.866 |

*** $p < 0.01$; ** $p < 0.05$; * $p < 0.1$.

Table 5: Regression results under carbon tax regulation.

| Variables | Model (17) $S$ | Model (18) $E$(kg) | Model (19) $Z_2$(\$) |
|---|---|---|---|
| $C_T$ (\$/kg) (standard error) | 0.714*** (0.020) | −69228*** (2717.332) | −121570*** (906.902) |
| Constant (standard error) | 0.312*** (0.102) | 160212*** (1572.769) | 56842*** (524.904) |
| Observations | 101 | 101 | 101 |
| $R$-squared | 0.927 | 0.868 | 0.994 |

*** $p < 0.01$; ** $p < 0.05$; * $p < 0$.

Table 6: Regression results under carbon cap-and-trade regulation.

| Variables | Model (20) $S$ | Model (21) $E$(kg) | Model (22) $Z_3$(\$) |
|---|---|---|---|
| $C_p$(standard error) | 0.715*** (0.006) | −69466.637*** (845.203) | −122920.356*** (620.912) |
| $C_p{}^*C_{cap}$(standard error) | | | 199682.801*** (788.726) |
| Constant (standard error) | 0.324*** (0.004) | 160858.469*** (493.1061) | 58381.626*** (163.311) |
| Observations | 1001 | 1001 | 1001 |
| R-squared | 0.923 | 0.871 | 0.986 |

*** $p < 0.01$; ** $p < 0.05$; * $p < 0.1$.

positive effects on the environment and negative effects on the economy [19]. For the government, the permissive policy cannot achieve improvement of the environment, whereas the severe policy will affect the economic development. It is necessary to design appropriate carbon policy parameters from the perspective of maximizing social welfare.

Social welfare includes economic and environmental utilities. Economic utility is a positive utility. In the research environment of this paper, the economic utility under carbon cap and cap-and-trade policies is the profit of the enterprise, and that under a carbon tax policy includes the carbon tax revenue obtained by the government. Environmental utility is a negative utility, which reflects the environmental damage caused by carbon emission. We use a quadratic environmental damage function to represent environmental utility [46]. The social welfare function is expressed in equation (23). The first and second items in equation (23) are the economic and environmental utilities, respectively:

$$\mathrm{SW} = U_{\mathrm{ec}} - U_{\mathrm{en}}. \tag{23}$$

Under the carbon cap policy, $U_{\mathrm{ec}} = \alpha_{13} + \beta_{13}C_{\mathrm{CAP}}$ and $E_1 = \alpha_{12} + \beta_{12}C_{\mathrm{CAP}}$. Substituting $U_{\mathrm{ec}}$ and $U_{\mathrm{en}}$ into equation (23), we obtain the social welfare as shown in equation (24):

$$\mathrm{SW} = \alpha_{13} + \beta_{13}C_{\mathrm{cap}} - \frac{\varepsilon\left(\alpha_{12} + \beta_{12}C_{\mathrm{cap}}\right)^2}{2}. \tag{24}$$

By the first-order derivative, the optimal carbon cap ratio that maximizes social welfare under the carbon cap policy can be obtained (equation (24)):

$$C^*_{\mathrm{cap}} = \frac{\beta_{13} - \varepsilon\beta_{12}\alpha_{12}}{\varepsilon\beta_{12}^2}. \tag{25}$$

Set $\varepsilon = 10^{-6}$. Substituting the results of the regression models (14)–(16) into equation (25), we obtain the optimal carbon cap ratio $C^*_{\mathrm{cap}} = 77.8\%$. The corresponding social welfare, profits, carbon emission, and sustainability levels are shown in Table 7.

Under the carbon tax policy, $E_2 = \alpha_{22} + \beta_{22}C_T$ and $U_{\mathrm{ec}} = \alpha_{23} + \beta_{23}C_T + C_TE_2$ , where $C_TE_2$ is the carbon tax levied by governments. Equation (26) presents the social welfare under the carbon tax policy:

$$\mathrm{SW} = \alpha_{23} + \beta_{23}C_T + C_T * E_2 - \frac{\varepsilon\left(\alpha_{22} + \beta_{22}\,C_T\right)^2}{2}. \tag{26}$$

Similarly, through the first-order derivative, the optimal carbon tax price that maximizes social welfare under the carbon tax policy can be obtained from the following equation:

Table 7: Decision results under different carbon policies.

| Policy level | Parameter value | Social welfare | Profit ($) | Carbon emission (kg) | Sustainability |
|---|---|---|---|---|---|
| CAP | 77.8% | 50618 | 62194 | 152158 | 0.501 |
| TAX | 347$/tonne | 52641 | 14657 | 136190 | 0.560 |
| Cap-and-trade | 347$/tonne, 77.8% | 62308 | 69635 | 121054 | 0.573 |

$$C_T^* = \frac{\beta_{23} + \alpha_{22} - \varepsilon\beta_{22}\alpha_{22}}{\varepsilon\beta_{22}^2 - 2\beta_{22}}. \quad (27)$$

Based on the results of the regression models (17)–(19), we obtain the optimal carbon tax price $C_T^*$ = \$347/tonne. The corresponding social welfare, profit, carbon emission, and sustainability level are shown in Table 7.

Under the carbon cap-and-trade policy, $U_{ec} = \alpha_{33} + \beta_{33} C_p + \beta_{34} (C_p C_{cap})$ and $E_3 = \alpha_{32} + \beta_{32} C_p$. The social welfare under the carbon cap-and-trade policy is shown in the following equation:

$$\mathrm{SW} = \alpha_{33} + \beta_{33} C_p + \beta_{34}\left(C_p C_{cap}\right) - \frac{\varepsilon\left(\alpha_{32} + \beta_{32} C_p\right)^2}{2}. \quad (28)$$

There are two parameters, that is, $C_P$ and $C_{cap}$, under carbon cap-and-trade policy. Since carbon trading price $C_P$ is usually determined by the supply and demand of the carbon trading market, the government mainly needs to decide the $C_{cap}$ value. Taking the first derivative of SW with respect to $C_{cap}$, we obtain $(\mathrm{dSW}/\mathrm{d}C_{cap}) = \beta_{34} C_p > 0$. That is, social welfare increases monotonously with the carbon cap ratio. From this, a very interesting conclusion can be drawn: when the carbon cap ratio is 100%, that is, the maximum value, social welfare is maximized. This conclusion indicates that to maximize social welfare, the government should increase the carbon cap as much as possible because it has no restrictive effect on enterprises. Based on the conclusion of the previous regression models, we summarized the causal influence mechanism in the social welfare system under the carbon trading policy in Figure 2.

As shown in the figure, sustainability level and carbon emission are determined by carbon trading price and are not related to carbon cap. The carbon cap will only affect the profit [17]. Therefore, when the carbon trading price is fixed, regardless of the carbon cap, the sustainability level and carbon emission are also fixed, resulting in a fixed environmental utility. The greater the carbon cap, the greater the carbon emission that an enterprise can obtain for free. Moreover, the profit of the enterprise is higher, and the economic utility is greater, and, therefore, the social welfare is greater. To be precise, the maximum social welfare mentioned here actually refers to the social welfare caused by a single enterprise. From the perspective of the whole society, social welfare is actually unchanged. When conducting carbon trading, some companies increase their profits due to the sale of carbon emission rights, and, thus, there must be a decline in profits due to the purchase of carbon emission rights. Therefore, from the perspective of the entire society, carbon trading itself has no impact on economic utility and

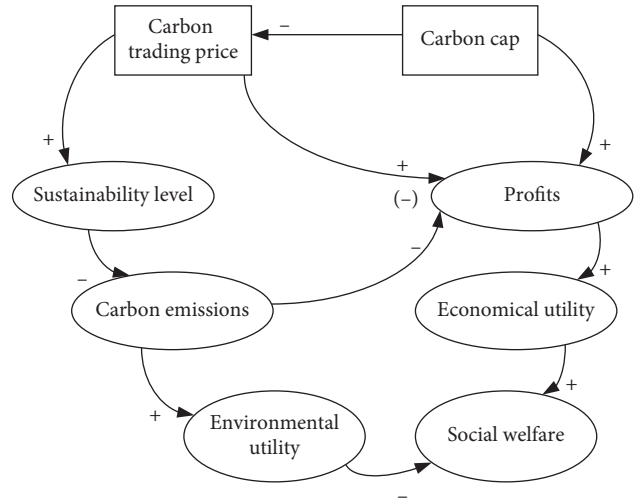



Figure 2: Causal relationships under cap-and-trade policy.

social welfare. Under the carbon cap-and-trade policy, social welfare is determined by the carbon trading price. The government can exert two aspects of regulation effect through a reasonable carbon cap. The first aspect is to adjust the relative balance of profits in different industries through carbon cap. For industries with higher profit margins, a strict carbon cap policy can be adopted. By contrast, industries with lower margins can adopt a loose carbon cap policy. The second aspect is to stabilize the carbon trading price through the carbon cap. When the carbon trading price is high, the carbon cap could be set at a large value to reduce the carbon trading price and vice versa.

To compare the results under various policies, we use the optimal parameter values under the carbon cap and the carbon tax policies. Moreover, the decision results under the carbon cap-and-trade policy are calculated (i.e., ($C_p$ = \$347/tonne, $C_{cap}$ = 77.8%)). The corresponding social welfare, profit, carbon emission, and sustainability level are shown in Table 7.

The operation results reveal that the sustainability level under the carbon cap policy is the lowest, whereas that under the carbon cap-and-trade policy is the highest. Under the carbon cap policy, the sustainability level is 0.501, and its carbon emission is the largest of the three policies. The enterprises' emission reduction goal is to control carbon emission within the carbon cap. Hence, when the carbon cap is high, the enterprises' enthusiasm for reducing emissions is not strong. Meanwhile, under the carbon tax policy, social welfare is slightly higher than the carbon cap policy. However, the enterprise's profit is only \$14,657, and the carbon tax revenue is \$47,258. This is because under the carbon tax policy, the government is both the leader and the

beneficiary of the policy, so it sets a relatively high carbon tax price. A high tax price not only can encourage enterprises to increase their sustainability level, reduce carbon tax costs, and improve the economic utility, but also can maintain social welfare at a high level. However, due to high carbon tax price, the enterprise's profit has been greatly reduced. The results shown in Table 7 indicate that the enterprise's profit under the carbon tax policy has fallen by 76.4% compared with that under the carbon cap policy. Therefore, excessively high carbon tax price will frustrate the enterprise's enthusiasm for sustainable investment [30]. If the enterprise gives up its business because of low profits, it will be a double defeat for the government and the enterprise. Therefore, if the government takes part of the carbon tax revenue to subsidize the enterprise [30], the enterprise's profit will increase, as well as their enthusiasm to achieve a win-win situation. The social welfare, enterprise's profit, and carbon emission under the carbon cap-and-trade policy are superior to those under the carbon cap policy and carbon tax policy [42]. This is because carbon cap-and-trade policy absorbs the advantages of carbon cap and carbon tax policies. To obtain greater profit, enterprises have increased their sustainability levels to reduce their carbon emission. They could even make greater profits by selling the remaining carbon emission rights. The analysis results show that a single carbon policy has certain limitations, and a composite carbon policy can consider both the economic and emission reduction goals, thereby achieving the coordination of economic and environmental utilities.

## 7. Conclusions

Integrating environmental factors into inventory-transportation problem is a hot topic in sustainable development, and carbon policies such as cap, tax, and cap-and-trade are regarded as effective ways to reduce carbon emission. However, research on the integrated inventory-transportation model simultaneously taking into account consumer environmental consciousness, sustainable investment, and carbon policies on a global scale is lacking. Under carbon cap, tax, and cap-and-trade policies, we investigated consumer environmental consciousness extended in inventory-transportation problem with demand dependent on the sustainability level of the product. First, we examined four integrated inventory-transportation models under different policy cases (i.e., no carbon policy, cap, tax, and cap-and-trade) and presented corresponding algorithms to optimize EOQ by giving sustainability level. Second, we designed a simulation method to determine the appropriate sustainability level and EOQ simultaneously. Third, we performed simulations under carbon cap, tax, and cap-and-trade policies, and we used regression models to analyze the effect of carbon policies on the sustainability level, profit, and carbon emission. Finally, by maximizing social welfare, we estimated and discussed carbon policy parameters on the basis of the regression results.

Under carbon cap policy, the carbon cap positively correlates with enterprises' profit and carbon emission but negatively correlates with the sustainability level. Meanwhile, under carbon tax policy, the carbon tax price significantly positively correlates with the sustainability level but negatively correlates with enterprises' profit and carbon emission. Moreover, under carbon cap-and-trade policy, the carbon trading price is the decisive factor that affects the sustainability level, enterprises' profit, and carbon emission. The carbon cap has no effect on the sustainability level and enterprises' carbon emission but has a positive regulatory effect on the relationship between carbon trading prices and profits. Furthermore, the carbon trading price is positively related to the sustainability level and inversely related to carbon emission. When carbon trading price is low (high), carbon trading price negatively (positively) correlates with profit. The results indicate that carbon cap-and-trade policy is superior to carbon cap and tax policies because it could absorb the advantages of the single carbon policy and achieve the coordination of economic and environmental utilities.

Under carbon cap and tax policies, policy parameters could be estimated from the perspective of social welfare maximization by using regression models. The governments tend to set a high carbon tax parameter because they benefit from the carbon tax. However, a high carbon tax parameter will markedly decrease enterprises' profit, which has a significantly negative effect on enterprises' enthusiasm for sustainable investment. Therefore, the government subsidy strategy is needed when a severe carbon tax policy is adopted. Under the carbon cap-and-trade policy, the carbon trading price has the same effect as the carbon tax price when the carbon cap is 0. The situation when carbon cap is 0 means that enterprises do not have free carbon emission permits. At this point, the same drawback as carbon tax policy will occur; that is, enterprises' profit will be decreased markedly. Hence, the governments should avoid the situation to promote the initiative of enterprises to reduce emissions in practice. Though governments cannot estimate carbon cap parameter by maximizing social welfare, they should set an appropriate carbon cap parameter according to the profit of enterprise and carbon trading price to play its regulatory role.

The novel contributions of this paper are as follows. Integrated inventory-transportation optimization models considering the sustainability of the product under carbon cap, tax, cap-and-trade policies were developed. To our knowledge, this is the first study to combine consumer environmental awareness, sustainable technology, and comprehensive emission model in inventory-transportation problem, thereby closing this gap in the literature. Furthermore, simulations were designed to optimize inventory-transportation models. Based on the simulation results, regression models are proposed to analyze the effect of carbon policy parameters on sustainability level, profit, and emission. The policy parameters are estimated and analyzed using the regression results. The proposed models could be used as a reference for enterprises needing to formulate inventory-transportation scheduling and governments intending to implement carbon policies.

Although this paper has several novel contributions, certain limitations should be considered. Future research

should address the following aspects: (1) The homogeneous vehicle was used as a delivery medium in the model. The possible extension of the model is to consider heterogeneous vehicles in future studies. (2) Our research assumes a single item. Extending the model with multiple items is also a direction worthy of further study. (3) We assume that the carbon label only includes carbon emission in production. In the future research, the carbon emission in inventory and transportation should be included in the carbon label, thereby making the optimization model more complicated and challenging. (4) The last extension worth mentioning is to expand inventory-transportation model using more complex logistics networks, such as "one-to-many" or "many-to-many", and consider varying consumer environmental consciousness in different markets.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was funded by the research project on Humanity and Social Science of the Ministry of Education in China (Grant no. 17YJAZH074) and the Key Subject Development Project of Management Science and Engineering of Hubei University of Education.

## References

[1] C. Xu, X. Liu, C. Wu, and B. Yuan, "Optimal inventory control strategies for deteriorating items with a general time-varying demand under carbon emission regulations," *Energies*, vol. 13, no. 4, p. 999, 2020.

[2] H. Yang and W. Chen, "Retailer-driven carbon emission abatement with consumer environmental awareness and carbon tax: revenue-sharing versus cost-sharing," *Omega*, vol. 78, pp. 179–191, 2018.

[3] Y.-S. Huang, C.-C. Fang, and Y.-A. Lin, "Inventory management in supply chains with consideration of Logistics, green investment and different carbon emissions policies," *Computers & Industrial Engineering*, vol. 139, p. 106207, 2020.

[4] Y.-J. Zhang, A.-D. Wang, and W. Tan, "The impact of China's carbon allowance allocation rules on the product prices and emission reduction behaviors of ETS-covered enterprises," *Energy Policy*, vol. 86, pp. 176–185, 2015.

[5] D. Konur and B. Schaefer, "Integrated inventory control and transportation decisions under carbon emissions regulations: LTL vs. TL carriers," *Transportation Research Part E: Logistics and Transportation Review*, vol. 68, pp. 14–38, 2014.

[6] U. Mishra, J.-Z. Wu, and B. Sarkar, "A sustainable production-inventory model for a controllable carbon emissions rate under shortages," *Journal of Cleaner Production*, vol. 256, p. 120268, 2020.

[7] E. Koberg and A. Longoni, "A systematic review of sustainable supply chain management in global supply chains," *Journal of Cleaner Production*, vol. 207, pp. 1084–1098, 2019.

[8] Y. Cheng, H. Sun, F. Jia, and L. Koh, "Pricing and low-carbon investment decisions in an emission dependent supply chain under a carbon labelling scheme," *Sustainability*, vol. 10, no. 4, p. 1238, 2018.

[9] S. Du, J. Zhu, H. Jiao, and W. Ye, "Game-theoretical analysis for supply chain with consumer preference to low carbon," *International Journal of Production Research*, vol. 53, no. 12, pp. 3753–3768, 2015.

[10] Y. Cheng, Y. Kuang, X. Shi, and C. Dong, "Sustainable investment in a supply chain in the big data era: an information updating approach," *Sustainability*, vol. 10, no. 2, p. 403, 2018.

[11] I. Nouira, Y. Frein, and A. B. Hadj-Alouane, "Optimization of manufacturing systems under environmental considerations for a greenness-dependent demand," *International Journal of Production Economics*, vol. 150, pp. 188–198, 2014.

[12] S. A. R. Khan, Y. Zhang, M. Anees, H. Golpîra, A. Lahmar, and D. Qianli, "Green supply chain management, economic growth and environment: a GMM based evidence," *Journal of Cleaner Production*, vol. 185, pp. 588–599, 2018.

[13] C. Dong, B. Shen, P.-S. Chow, L. Yang, and C. T. Ng, "Sustainability investment under cap-and-trade regulation," *Annals of Operations Research*, vol. 240, no. 2, pp. 509–531, 2016.

[14] P. Rao and D. Holt, "Do green supply chains lead to competitiveness and economic performance?" *International Journal of Operations & Production Management*, vol. 25, no. 9, pp. 898–916, 2005.

[15] O. Jabali, T. Van Woensel, and A. G. de Kok, "Analysis of travel times and CO2Emissions in time-dependent vehicle routing," *Production and Operations Management*, vol. 21, no. 6, pp. 1060–1074, 2012.

[16] Y. Bouchery, A. Ghaffari, Z. Jemai, and T. Tan, "Impact of coordination on costs and carbon emissions for a two-echelon serial economic order quantity problem," *European Journal of Operational Research*, vol. 260, no. 2, pp. 520–533, 2017.

[17] S. Tang, W. Wang, S. Cho, and H. Yan, "Reducing emissions in transportation and inventory management: (R, Q) Policy with considerations of carbon reduction," *European Journal of Operational Research*, vol. 269, no. 1, pp. 327–340, 2018.

[18] S. Benjaafar, Y. Li, and M. Daskin, "Carbon footprint and the management of supply chains: insights from simple models," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 99–116, 2013.

[19] A. Toptal, H. Özlü, and D. Konur, "Joint decisions on inventory replenishment and emission reduction investment under different emission regulations," *International Journal of Production Research*, vol. 52, no. 1, pp. 243–269, 2014.

[20] G. J. L. Micheli and F. Mantella, "Modelling an environmentally-extended inventory routing problem with demand uncertainty and a heterogeneous fleet under carbon control policies," *International Journal of Production Economics*, vol. 204, pp. 316–327, 2018.

[21] A. Mosca, N. Vidyarthi, and A. Satir, "Integrated transportation - inventory models: a review," *Operations Research Perspectives*, vol. 6, p. 100101, 2019.

[22] A. Toptal, S. Çetinkaya, and C.-Y. Lee, "The buyer-vendor coordination problem: modeling inbound and outbound cargo capacity and costs," *Iie Transactions*, vol. 35, no. 11, pp. 987–1002, 2003.

[23] M. Soysal, J. M. Bloemhof-Ruwaard, R. Haijema, and J. G. A. J. van der Vorst, "Modeling a green inventory routing problem for perishable products with horizontal collaboration," *Computers & Operations Research*, vol. 89, pp. 168–182, 2018.

[24] S. Bose, A. Goswami, and K. S. Chaudhuri, "An EOQ model for deteriorating items with linear time-dependent demand rate and shortages under inflation and time discounting," *The Journal of the Operational Research Society*, vol. 46, no. 6, pp. 771–782, 1995.

[25] R. R. Chowdhury, S. Ghosh, and K. Chaudhuri, "An optimal inventory replenishment policy for a perishable item with time quadratic demand and partial backlogging with shortages in all cycles," *International Journal of Applied and Computational Mathematics*, vol. 3, pp. 1001–1017, 2017.

[26] L. Bertazzi, A. Bosco, and D. Laganà, "Managing stochastic demand in an Inventory Routing Problem with transportation procurement," *Omega*, vol. 56, pp. 112–121, 2015.

[27] S. Sindhuchao, H. E. Romeijn, E. Akçali, and R. Boondiskulchok, "An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity," *Journal of Global Optimization*, vol. 32, no. 1, pp. 93–118, 2005.

[28] V. Hovelaque and L. Bironneau, "The carbon-constrained EOQ model with carbon emission dependent demand," *International Journal of Production Economics*, vol. 164, pp. 285–291, 2015.

[29] B. Rabta, "An Economic Order Quantity inventory model for a product with a circular economy indicator," *Computers & Industrial Engineering*, vol. 140, Article ID 106215, 2020.

[30] S. Wang, F. Tao, and Y. Shi, "Optimization of inventory routing problem in refined oil logistics with the perspective of carbon tax," *Energies*, vol. 11, no. 6, p. 1437, 2018.

[31] D. Zhang, Q. Zhan, Y. Chen, and S. Li, "Joint optimization of logistics infrastructure investments and subsidies in a regional logistics network with CO2 emission reduction targets," *Transportation Research Part D: Transport and Environment*, vol. 60, pp. 174–190, 2018.

[32] Z. Rafie-Majd, S. H. R. Pasandideh, and B. Naderi, "Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm," *Computers & Chemical Engineering*, vol. 109, pp. 9–22, 2018.

[33] M. Alım and P. Beullens, "Joint inventory and distribution strategy for online sales with a flexible delivery option," *International Journal of Production Economics*, vol. 222, Article ID 107487, 2020.

[34] P. Gautam, A. Kishore, A. Khanna, and C. K. Jaggi, "Strategic defect management for a sustainable green supply chain," *Journal of Cleaner Production*, vol. 233, pp. 226–241, 2019.

[35] M. Biuki, A. Kazemi, and A. Alinezhad, "An integrated location-routing-inventory model for sustainable design of a perishable products supply chain network," *Journal of Cleaner Production*, vol. 260, Article ID 120842, 2020.

[36] J. Xu, X. Cui, Y. Chen, and X. Zhang, "Joint transportation and inventory strategy for perishable items with weibull distribution under carbon emission regulations," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7589647, 11 pages, 2018.

[37] G. Hua, T. C. E. Cheng, and S. Wang, "Managing carbon footprints in inventory management," *International Journal of Production Economics*, vol. 132, no. 2, pp. 178–185, 2011.

[38] X. Chen, S. Benjaafar, and A. Elomri, "The carbon-constrained EOQ," *Operations Research Letters*, vol. 41, no. 2, pp. 172–179, 2013.

[39] H. Liao and Q. Deng, "A carbon-constrained EOQ model with uncertain demand for remanufactured products," *Journal of Cleaner Production*, vol. 199, pp. 334–347, 2018.

[40] Y. Yu, X. Han, and G. Hu, "Optimal production for manufacturers considering consumer environmental awareness and green subsidies," *International Journal of Production Economics*, vol. 182, pp. 397–408, 2016.

[41] L. Zhang, J. Wang, and J. You, "Consumer environmental awareness and channel coordination with two substitutable products," *European Journal of Operational Research*, vol. 241, no. 1, pp. 63–73, 2015.

[42] D. F. Drake, P. R. Kleindorfer, and L. N. Van Wassenhove, "Technology choice and capacity portfolios under emissions regulation," *Production and Operations Management*, vol. 25, no. 6, pp. 1006–1025, 2016.

[43] Z. Tao and J. Xu, "Carbon-regulated EOQ models with consumers' low-carbon awareness," *Sustainability*, vol. 11, no. 4, p. 1004, 2019.

[44] C. Su, X. Liu, and W. Du, "Green supply chain decisions considering consumers' low-carbon awareness under different government subsidies," *Sustainability*, vol. 12, no. 6, p. 2281, 2020.

[45] H. L. Lee and M. J. Rosenblatt, "A generalized quantity discount pricing model to increase supplier's profits," *Management Science*, vol. 32, no. 9, pp. 1177–1185, 1986.

[46] T. A. Weber and K. Neuhoff, "Carbon markets and technological innovation," *Journal of Environmental Economics and Management*, vol. 60, no. 2, pp. 115–132, 2010.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 7959508, 12 pages
https://doi.org/10.1155/2020/7959508

*Research Article*

# A Two-Stage Method for Improving the Prediction Accuracy of Complex Traits by Incorporating Genotype by Environment Interactions in *Brassica napus*

**Sican Xiong**,[1,2] **Meng Wang**,[3] **Jun Zou**,[3] **Jinling Meng**,[3] **and Yanyan Liu**[2]

[1]*School of Science, East China University of Technology, Nanchang, Jiangxi 330013, China*
[2]*School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China*
[3]*National Key Laboratory of Crop Genetic Improvement, Huazhong Agricultural University, Wuhan, Hubei 430070, China*

Correspondence should be addressed to Sican Xiong; xsc060@126.com

Received 4 July 2020; Revised 23 July 2020; Accepted 24 July 2020; Published 14 August 2020

Academic Editor: Chin-Chia Wu



Improving the prediction accuracy of a complex trait of interest is key to performing genomic selection (GS) for crop breeding. For the complex trait measured in multiple environments, this paper proposes a two-stage method to solve a linear model that jointly models the genetic effects and the genotype × environment interaction ($G \times E$) effects. In the first stage, the least absolute shrinkage and selection operator (LASSO) penalized method was utilized to identify quantitative trait loci (QTL). Then, the ordinary least squares (OLS) approach was used in the second stage to reestimate the QTL effects. As a case study, this approach was used to improve the prediction accuracies of flowering time (FT), oil content (OC), and seed yield per plant (SY) in *Brassica napus* (*B. napus*). The results showed that the $G \times E$ effects reduced the mean squared error (MSE) significantly. Numerous QTL were environment-specific and presented minor effects. On average, the two-stage method, named OLS post-LASSO, offers the highest prediction accuracies (correlations are 0.8789, 0.9045, and 0.5507 for FT, OC, and SY, respectively). It was followed by the marker × environment interaction ($M \times E$) genomic best linear unbiased prediction (GBLUP) model (correlations are 0.8347, 0.8205, and 0.4005 for FT, OC, and SY, respectively), the LASSO method (correlations are 0.7583, 0.7755, and 0.2718 for FT, OC, and SY, respectively), and the stratified GBLUP model (correlations are 0.6789, 0.6361, and 0.2860 for FT, OC, and SY, respectively). The two-stage method showed an obvious improvement in the prediction accuracy, and this study will provide methods and reference to improve GS of breeding.

## 1. Introduction

In the last three decades, the development of molecular marker technology has provided numerous molecular markers for the most important species [1]. Regarding the use of molecular markers in the selection of a genetic trait, marker-assisted selection (MAS) [2] became a valuable tool in animal and plant breeding in the 1990s and works well for traits with a simple genetic architecture. However, MAS is not suitable for complex traits controlled by multiple genes, many of which have minor effects. Therefore, genomic selection (GS) as an advanced form of MAS was first propounded by Meuwissen et al. [3]. Instead of using a subset of significant markers for selection in MAS, GS uses whole genome-wide markers to predict the genome-estimated breeding values (GEBVs) of selected individuals and thus avoids biased marker effects estimates due to the detection process in MAS. However, with high-density molecular markers, the number of markers ($p$) can vastly exceed the sample size ($n$), which is referred to as a "large $p$ small $n$" problem. Thus, it is impossible to obtain the estimates of markers effects via a linear model by OLS [4].

To deal with the "large $p$ small $n$" problem, one can impose some constraints on the linear model, resulting in penalized estimation methods, such as ridge regression (RR) [5] and LASSO [6]. RR performs parameter shrinkage only, while LASSO offers both parameter shrinkage and variable selection simultaneously. Both RR and LASSO can generate

parsimonious models in the presence of a large number of predictors. The predictor size selected by LASSO is generally less than the sample size ($n$) [7], so applying OLS to the model selected by LASSO is feasible. The post-model-selection estimator was called "OLS post-LASSO" in the study of Belloni and Chernozhukov [8] and it possesses the advantage of a smaller bias than LASSO. The debiasing in OLS post-LASSO often improves the prediction error of the model [9], and this two-stage process is also known as the relaxed LASSO [10].

Bayesian methods are also applied to fit this "large $p$ small $n$" problem in GS [11]. In Bayesian inference, the marker effects are considered as random instead of fixed, and the mixed-effects model is often adopted to describe the phenotypic variation. By specifying different priors for the random marker effects, many different models, including BLUP (best linear unbiased prediction) [12], BayesA, BayesB [3], BayesC [13], and Bayesian LASSO [14], have been proposed in GS (see de los Campos et al. [11] for an overview). BLUP is a statistical procedure used to estimate the random effects and is easily obtained by solving the well-known Henderson's mixed model equations (MME) [15]. Thus, BLUP and its extensions, including rrBLUP (ridge regression BLUP) [16,17] and GBLUP [18], have become the most widely used methods in GS. Many software packages for those methods, for example, rrBLUP [17] and BGLR (Bayesian Generalized Linear Regression) [19], are freely available online. Recently, the GBLUP model has been extended to multienvironment data. For example, Lopez-Cruz et al. [20] proposed a $M \times E$ GBLUP model to accommodate the $G \times E$, and they also compared the $M \times E$ GBLUP model with the stratified (within the environment) GBLUP model. The results showed that the prediction accuracy of the $M \times E$ GBLUP model was substantially greater than the stratified GBLUP model. The significant increase in the prediction accuracy of using multienvironment models compared with single-environment analysis has been confirmed in many crops, such as maize [21] and rice [22].

*B. napus* is one of the most important oil crops worldwide. To better understand the genetic control of important agronomic traits in *B. napus*, a doubled haploid (DH) population, named TNDH population, which was derived from the F1 cross between European cultivar Tapidor and Chinese cultivar Ningyou7, was developed [23]. After several years and trial locations, the phenotypic data were collected from a multienvironment, and the TNDH population has been adopted as reference resources by the OREGIN (Oilseed Rape Genetic Improvement Network) management team. Based on the TNDH population, many QTL for the complex traits have been detected (see Shi et al. [24], etc.). Recently, a high-density genetic map of the TNDH population with a total of 2041 molecular markers was constructed [25]. Using this high-density genetic map, the genomic prediction accuracy of the FT trait in *B. napus* was evaluated via eight existing models by Li et al. [26]. However, the authors did not incorporate the $G \times E$ effects into their study. As previously noted, the $G \times E$ effects play a very important role in explaining the variation of the complex traits. Accumulating studies showed that incorporating $G \times E$ effects into the GS model could substantially increase the prediction accuracy of the complex trait. Therefore, in this study, based on the representative TNDH population, we will evaluate the performance of a two-stage approach via a linear model that jointly models the genetic effects and $G \times E$ effects. The objective of the present study is to improve the prediction accuracy of complex traits for *B. napus*. In contrast to the most commonly used $G \times E$ GS models, such as the $M \times E$ GBLUP model of Lopez-Cruz et al. [20], we treat the marker effects as fixed instead of random. We assume that the LASSO method can be used to identify the main effect and environment-specific effect QTL. Based on the identified QTL, a parsimonious linear model can be established and the OLS method is used to reestimate the QTL's effects. The performances of this two-stage approach named OLS post-LASSO and other comparison methods, including LASSO, $M \times E$ GBLUP model, and the stratified GBLUP model of Lopez-Cruz et al. [20], are evaluated in terms of prediction accuracy for FT, OC, and SY.

## 2. Materials and Methods

*2.1. Genotypic and Phenotypic Data.* A published dataset for the TNDH population was used in this study (see Luo et al. [27] for details). The TNDH population was derived from an $F_1$ progeny of a cross between a European winter cultivar "Tapidor" and a Chinese semiwinter cultivar "Ningyou7" [23]. The population comprises 182 DH lines grown at five different sites (Wuhan, Jiangling, Daye, Hangzhou, and Dali) in China for over five years (2002–2007). Combining the harvest year and the location, a total number of ten environments (year-location combinations) are available and are coded as "S3," "S4," "S5," "S6," "S7," "E7," "N3," "N4," "N6," and "N7," separately. There are a total of 2041 molecular markers for each DH line genotyped ("A" and "B" were denoted for "Tapidor" and "Ningyou7", respectively) by the *Brassica* 60K Illumina Infinium SNP array, and a total of 22 traits, including SY, OC, and FT, are collected from all the ten environments. Details of phenotypic and genotypic data and how the TNDH population was developed can be found in Luo et al. [27]. These 182 TNDH lines, the 2041 markers, and the phenotypic data for three complex traits (SY, OC, and FT) across all the ten environments were used in the present study.

*2.2. Two-Stage Approach.* Suppose that there are several $n$ population lines (individuals) cultivated in a total of $m$ environments, $y_{ki}$ is individual $i$'s trait value collected from environment $k$ ($i = 1, \ldots, n$, $k = 1, \ldots, m$). Since those $n$ individuals cultivated in different environments share the same genotypes, we use $x_{ij}$ to denote the genotype of individual $i$ at locus $j$ ($j = 1, \ldots, p$). $x_{ij} = 1$ and 0, respectively, represent A and B genotypes, where $p$ is the number of markers. For jointly modeling the genetic effect and the $G \times E$ effect, a $G \times E$ linear model by regressing phenotypes on markers across all the multiple environments can be described as follows:

$$y_{ki} = \mu + \mu_k + \sum_{j=1}^{p} x_{ij}\left(\beta_j + \alpha_{kj}\right) + \epsilon_{ki} \quad (i = 1, \ldots, n; k = 1, \ldots, m), \tag{1}$$

where $\mu$ is the overall mean (the intercept term), which is stable across the environments, and $\mu_k$ is the environment effect ($E$) that may vary by environment, $\beta_j$is the main effect across all the environments ($Q$), $\alpha_{kj}$is the environment-specific effect or equivalently the interaction effect between the $j^{th}$ locus and the $k^{th}$ environment ($Q \times E$), and $\epsilon_{ki} \sim N(0, \sigma^2)$ is the residual error. If some $\beta_j s$ or $\alpha_{kj} s$ are not equal to zero, we considered that there exist the main effects or the $Q \times E$ effects.

Model (1) is very similar to the "$M \times E$ GBLUP model" of Lopez-Cruz et al. [20]. In $M \times E$ GBLUP model, both the main effects and the environment-specific effects are treated as random effects. Also, the $M \times E$ GBLUP model does not include the overall mean and can be expressed as follows:

$$y_{ki} = \mu_k + \sum_{j=1}^{p} x_{ij}\left(\beta_j + \alpha_{kj}\right) + \epsilon_{ki} \quad (i = 1, \ldots, n, k = 1, \ldots, m), \tag{2}$$

where $\alpha_{kj} s$ are called $M \times E$ effects by Lopez-Cruz et al. [20]. Furthermore, ignoring the $M \times E$ effect and analyzing data separately in each environment, model (1) was reduced to as the "stratified GBLUP model" by Lopez-Cruz et al. [20]. The stratified GBLUP model can be expressed as follows:

$$y_{ki} = \mu_k + \sum_{j=1}^{p} x_{ij}\beta_{kj} + \epsilon_{ki} \quad (i = 1, \ldots, n, k = 1, \ldots, m), \tag{3}$$

where $\beta_{kj}$is the effect of the $j^{\text{th}}$ marker on the $k^{\text{th}}$ environment.

In the study, marker effects, including the main effect and the environment-specific effect, are considered as fixed instead of random. Considering that the number of parameters in model (1) is often larger than the sample size in GS, a two-stage approach, that is, OLS post-LASSO, was used to solve this problem, and the approach can be described as follows.

In the first stage, the LASSO method was used to select markers that have nonzero effects or equivalently detect QTL from the markers as pointed out by Zhang et al. [28]. In matrix form, model (1) can be expressed as

$$y = \mu 1_{mn} + 1_n \otimes I_m \mu + x \otimes 1_m \beta + x \otimes I_m \alpha + \varepsilon, \tag{4}$$

where $\otimes$ is the Kronecker product of two matrices and $1_{mn}$, $1_n$, and $1_m$ are the vectors of ones of order $mn$, $n$, and $m$, respectively. $I_m$ is the identity matrix of order $m$, $y = (y_{11}, \ldots, y_{1n}, \ldots, y_{m1}, \ldots, y_{mn})^T$ are the phenotypic values across environments, $x = (x_{ij})_{n \times p}$ is the genotype matrix, $\mu = (\mu_1, \ldots, \mu_m)$ are the environmental effects, $\beta = (\beta_1, \cdots, \beta_p)^T$ are the main marker effects, $\alpha = (\alpha_{11}, \ldots, \alpha_{1p}, \ldots, \alpha_{m1}, \ldots, \alpha_{mp})^T$ are the environment-specific marker effects, and $\epsilon = (\epsilon_{11}, \ldots, \epsilon_{1n}, \ldots, \epsilon_{m1}, \ldots, \epsilon_{mn})^T$is the residual error.

Let $Z = 1_n \otimes I_m$, $X = (x \otimes 1_m, x \otimes I_m)$, $\theta = (\beta^T, \alpha^T)^T$, and $K = (m+1)\, p$ is the number of columns in matrix $X$; then $Z$ is an $mn \times m$ matrix, $X$ is an $mn \times K$ matrix, and $\theta$ is a $K$-dimensional column vector. Matrices $Z$ and $X$ can be partitioned by columns, such that $\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_m)$ and $X = (X_1, \ldots, X_K)$, and vector $\theta$ can be written as $\theta = (\theta_1, \cdots, \theta_K)^T$. Then, model (4) can be expressed as

$$y = \mu 1_{mn} + \sum_{k=1}^{m} Z_k \mu_k + \sum_{l=1}^{K} X_l \theta_l + \varepsilon. \tag{5}$$

Equation (5) is the standard form of a linear model. Considering that the number of predictors ($K + m$) in model (5) is often much larger than the sample size ($mn$) in the context of $G \times E$, we use the LASSO method to solve the model first. The LASSO estimator of model (5) can be obtained by minimizing

$$\frac{1}{2}\left\| y - \left( \mu 1_{mn} + \sum_{k=1}^{m} Z_k \mu_k + \sum_{l=1}^{K} X_l \theta_l \right) \right\|^2 + \lambda \sum_{l=1}^{K} |\theta_l|, \tag{6}$$

where $\| \cdot \|$ denotes the $l_2$-norm and $\lambda \geq 0$is a tuning parameter. The tuning parameter can be selected by $k$-fold cross-validation, for example, using 10-fold cross-validation. As noted by many studies, such as Zou and Hastie [7], the number of nonzero effects selected by LASSO is generally less than the sample size, that is, $mn$ for model (6). Therefore, after LASSO, OLS regression using the selected QTL is feasible and possesses some advantages, especially reducing the shrinkage bias of LASSO [8].

Thus, in the second stage, we adopt the OLS to reestimate the QTL effects. This two-stage method was referred to as OLS post-LASSO by Belloni and Chernozhukov [8].

*2.3. Full Data Analysis.* To estimate the marker effects and show $G \times E$ relevance and structure in the dataset, model (5) was first fitted to the full dataset using OLS post-LASSO. As noted previously, the markers with nonzero main effects and environment-specific effects by LASSO in the first stage were reported as QTL. Based on the selected QTL, the OLS method was used to reestimate its values. The reestimated values from OLS are reported as the final estimated effects of QTL. We use the $t$-test to test whether the corresponding reestimated effect of each QTL is equal to zero or not. If the $p$ value of the $t$-test is less than 0.05, the corresponding QTL is reported as significant QTL. Otherwise, it is reported as a nonsignificant QTL. For the nonsignificant QTL, the corresponding effects are not significantly different from zero or, equivalently, the corresponding effects are ignorable. Meanwhile, the OLS approach can produce the corresponding standard errors (S.E.) of the estimate for the parameters. Based on the standard errors of the estimate for the QTL's effect, we can construct the 95% confidence intervals for the estimate of QTL's effect, including the main effects and the environment-specific

effects. The 95% confidence intervals are calculated by the estimated effects plus or minus 1.96 times the standard errors.

For the linear regression, R-squared is often reported as the measure that represents the proportion of the variation in the dependent variable explained by the model. However, R-squared always does not decrease as more predictors are added to the model. Thus, R-squared cannot be used to measure the contributions of each predictor. The adjusted R-squared does not increase as more predictors are added; thus it is chosen as the measure to evaluate the contribution of each component of the model. However, we cannot interpret the adjusted R-squared the same as R-squared. Note that the adjusted R-squared is equal to the percentage of the decrement of the MSE from the null model that only contains the intercept term to the alternative model that contains both the intercept term and other components of the model. Therefore, we calculate the MSE of both the null model and the alternative model. Meanwhile, we calculate the decrement and the percentage of decrement between them. To better understand the contributions of the three components of the model, that is, $E$, Q, and $Q \times E$, five alternative models incorporating the three components of the model and its combinations, $E + Q$ and $E + Q + Q \times E$, are evaluated in the article. Those alternative models with higher values of the adjusted R-squared, that is, higher percentages of the decrement of the MSE from the null model to the corresponding alternative model, are the better models. The corresponding components included in the better models would play an important role in prediction.

*2.4. Randomly Splitting the Data for Assessing Prediction Accuracy.* For comparison, the existing "$M \times E$ GBLUP model" and "stratified GBLUP model" mentioned above are chosen as comparison methods. Meanwhile, the $n = 182$ TNDH lines across all the ten environments (i.e., $m = 10$) are chosen as a working example for assessing the prediction accuracy of the two-stage method and the comparison methods. Based on the 182 TNDH lines, for each complex trait, that is, SY, OC, and FT, we merge the phenotypic datasets as a long vector just as described in the methodology from all the ten environments into one dataset. After merging, the sample size is enlarged to 1820 ($= nm = 182 \times 10$). Then, for each merged dataset, we randomly partition it into training and testing datasets at a proportion of 2 : 1. This random partition is repeated 100 times, resulting in a total of 100 random training datasets and the corresponding 100 random testing datasets. The marker effects are estimated on each training dataset across environments using LASSO, OLS post-LASSO, and $M \times E$ GBLUP model and within each environment using stratified GBLUP model. The GEBVs are computed in the corresponding testing dataset across environments using estimated LASSO, OLS post-LASSO, and $M \times E$ GBLUP model and within each environment using the estimated stratified GBLUP model. Then, we calculate the correlation between the GEBVs and the observed phenotypes for each trait within each environment. Taking the average across 100 replicated partitions, we obtain the average correlation and report it as prediction accuracy within each environment. Meanwhile, the standard deviation (SD) of the sampling distribution of the prediction accuracy among 100 replicated partitions is also reported to indicate the deviation of the accuracy.

*2.5. Software.* The minimizing problem of equation (6) can be efficiently solved by Least Angle Regression [29] in *R* software [30] using "lars" package or Alternating Direction Method of Multipliers (ADMM) [31] algorithms in MATLAB software using "lasso" function, which is used in the present study. The $M \times E$ GBLUP model (2) and stratified GBLUP model (3) are implemented using the *R* package BGLR [19].

## 3. Results

*3.1. Marker Effects.* The number of detected QTL and the frequency analysis of significant or nonsignificant QTL are reported in Table 1. From Table 1, we can see that the total number of QTL with main effects varied across traits. There are a total of 46, 77, and 26 QTL with main effects for FT, OC, and SY, respectively, and there are also a total of 231, 237, and 146 QTL with environment-specific effects for FT, OC, and SY, respectively. For main marker effects, fewer of them have significantly nonzero effects, and the percentages of significant QTL are 39.13%, 32.47%, and 42.31% for FT, OC, and SY, respectively. Equivalently, most identified main effect QTL by LASSO have small or ignorable effects, and the percentages of nonsignificant QTL are 60.87%, 67.53%, and 57.69% for FT, OC, and SY, respectively. For environment-specific marker effects, 15.58%, 16.03%, and 6.85% for FT, OC, and SY, respectively, have effects significantly different from zero. Thus, most identified environment-specific effects QTL by LASSO have small or ignorable effects.

Figures 1–3 show the point estimates and 95% confidence interval (95% CI) of marker main and environment-specific effects along the chromosomes. The vertical green confidence intervals that overlap the horizontal line of zero contain the value of zero; thus, the corresponding marker effects are nonsignificant. The vertical blue confidence intervals that represent the corresponding marker effects are statistically significant under the significance level of 0.05. As noted from these figures, most of the main and environment-specific marker effects are small and not significantly different from zero. Figures 1(c)–3(c) show the standard error (SE) of environment-specific marker effects for the same detected environment-specific QTL. The positive values (blue lines) of S.E. indicate that the corresponding QTL have environment-specific effects in multiple environments. Figures 1–3 show that few environment-specific effects QTL interact with multiple environments, and most of them display their effects in only one environment.

Table 1: The total number of QTL detected by LASSO and the frequency of significant and nonsignificant QTL by OLS post-LASSO (significance level $\alpha$ =0.05).

| Trait | Effect | Total | Significant | Nonsignificant |
|---|---|---|---|---|
| FT | Main | 46 | 18 | 28 |
| | Environment-specific | 231 | 36 | 195 |
| OC | Main | 77 | 25 | 52 |
| | Environment-specific | 237 | 38 | 199 |
| SY | Main | 26 | 11 | 15 |
| | Environment-specific | 146 | 10 | 136 |

FT, flowering time; OC, oil content; SY, seed yield per plant; QTL: quantitative trait loci; LASSO, least absolute shrinkage and selection operator; OLS, ordinary least squares.

*3.2. Decrement of MSE.* The decrements of the MSE from the null model that only includes the intercept term to the alternative model that includes both the intercept term and one of the components of $E$, Q, $Q \times E$, $E + Q$, and $E + Q + Q \times E$ are reported in Table 2.

From Table 2, we can see that the MSE of the null model is 198.4734 for FT, 6.2661 for OC, and 0.3503 for SY, respectively. After adding the component of the $Q \times E$ into the model, the decrement of the MSE is 189.3257 for FT, 4.6635 for OC, and 0.2143 for SY, respectively, and the corresponding percentage of the decrement, that is, the adjusted R-squared, is 95.3910% for FT, 74.4235% for OC, and 61.1769% for SY, respectively. That means $Q \times E$ plays a key role in the model. If adding the combination of $E$ and $Q$ into the null model, the corresponding percentage of the decrement of the MSE is 98.2912% for FT, 74.6738% for OC, and 62.0056% for SY, respectively, which is slightly larger than that of the alternative model that contains both intercept and $Q \times E$. The percentage of the decrement of MSE for the full model that contains all of the three components, $E + Q + Q \times E$, has the highest values of 98.6731%, 88.0751%, and 66.6784% for FT, OC, and SY, respectively. Thus, the full model is most appropriate to use to predict the complex traits.

Another interesting finding is that the percentage of the decrement of MSE for the model with combined components, such as $E + Q$ and $E + Q + Q \times E$, is not equal to the sum of the percentages of decrement of the MSE for all separated models that contain only one of them. This is because the main effects QTL and the environment-specific effects QTL are highly correlated. Correlations among them can change the percentages of the decrement of the MSE dramatically from what they would be in a separate model.

*3.3. Prediction Accuracy.* Respectively, the prediction accuracies of FT, OC, and SY are evaluated (Tables 3–5). From Tables 3–5, we can see that the highest average prediction accuracy among ten environments was obtained using the OLS post-LASSO method (average correlations are 0.8789, 0.9045, and 0.5507 for FT, OC, and SY, respectively). This two-stage method is followed by the $M \times E$ GBLUP model (average correlations are 0.8347, 0.8205, and 0.4005 for FT, OC, and SY, respectively). For FT and OC, the third performing method is the LASSO approach (average correlations are 0.7583 and 0.7755 for FT and OC, respectively). However, for SY, the stratified GBLUP method is the third performing method (average correlation is 0.2860). Thus, on average, the two-stage method always performs the best in prediction accuracy for all the three complex traits.

Also, although the performances of various methods vary in different environments, the OLS post-LASSO method is advantageous in all the 10 environments except for "N6" for FT and "S5" for SY. The OLS post-LASSO achieves its best accuracy in "S4" for FT (correlation is 0.9333), in "S7" for OC (correlation is 0.9188), and "N6" for SY (correlation is 0.7039). For FT, in the environment "N6," the $M \times E$ GBLUP model has higher prediction accuracy than the OLS post-LASSO method (correlations of 0.8512 and 0.8270 for the $M \times E$ GBLUP model and OLS post-LASSO method, respectively). For SY, in the environment "S5," the $M \times E$ GBLUP model also has higher prediction accuracy than the OLS post-LASSO method (correlations of 0.3924 and 0.2285 for the $M \times E$ GBLUP model and OLS post-LASSO method, respectively).

Generally, the LASSO method yielded lower prediction accuracies compared with the $M \times E$ GBLUP method for FT, OC, and SY, whereas the OLS post-LASSO method, which refits the model again based on the QTL identified by the LASSO method, outperforms the $M \times E$ GBLUP model. The improvement of prediction accuracy from LASSO to OLS post-LASSO is significant. For example, the average prediction accuracy of the $M \times E$ GBLUP model (correlation is 0.8347) for FT locates outside the 95% confidence interval of OLS post-LASSO ($0.8789 \pm 1.96 \times 0.0038 = [0.8715, 0.8864]$) (Table 3). In other words, the probability of the difference between the average prediction accuracies of the OLS post-LASSO and $M \times E$ GBLUP model is less than 0.05. Thus, the improvement is significant for FT, and this finding is also true for OC and SY as shown in Tables 4 and 5, respectively.

## 4. Discussion

Since GS was proposed by Meuwissen et al. [3] in 2001, numerous studies have been performed to increase the prediction accuracy of the trait of interest, and numerous approaches have been proposed for GS in different situations, especially BLUP type methods. As one of the derivations of the BLUP method, the GBLUP method has become a commonly used GS method and has shown success in many situations, such as in the presence of the $G \times E$. In this study, we established a general $G \times E$ linear model to simultaneously model the genetic effects and the $G \times E$ effects. By treating the marker effects, including the main marker effects across all environments and the environment-specific marker effects, as fixed instead of random, a two-stage method named OLS post-LASSO was used to solve the model and obtain a genomic prediction.

The OLS method was also used by Meuwissen et al. [3] but not in the context of $G \times E$. For using the OLS method in GS, the biggest effects selected by some procedure, such as single segment regression analysis performed by Meuwissen et al. [3], were included. However, this stepwise procedure

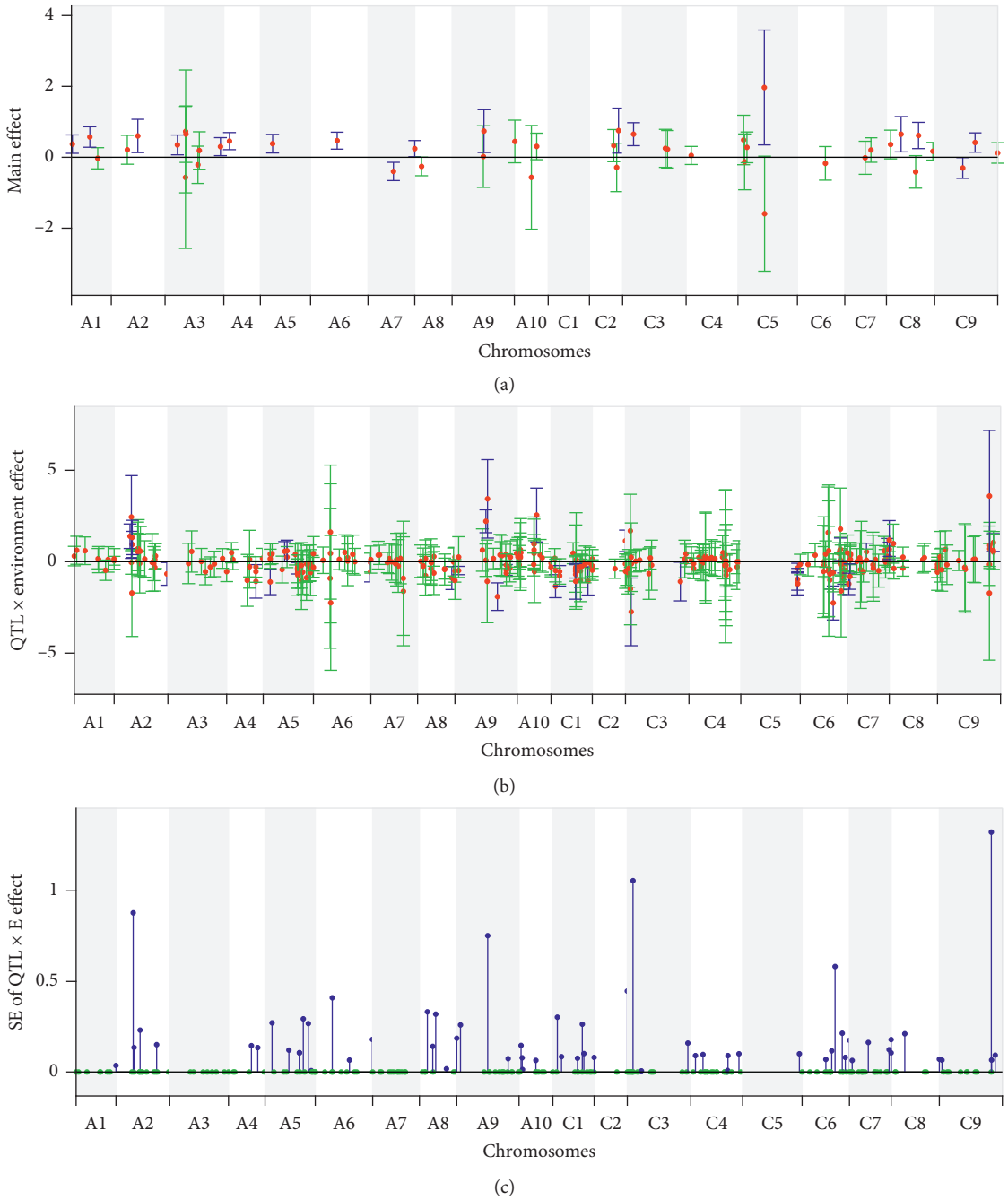



Figure 1: The point estimates and 95% confidence intervals (95% CI) for marker main and environment-specific effects and the standard error (SE) of environment-specific effects for FT. (a) Point estimates (red dots) and 95% CI for main effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (b) Point estimates (red dots) and 95% CI for environment-specific effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (c) SE for environment-specific effects (the blue and green stems represent environment-specific effects found in multiple environments or only one environment, respectively).

tends to overestimate the marker effects and cause lower prediction accuracy. This issue also exists in the context of QTL mapping using linkage disequilibrium (LD) or linkage analysis, especially in the context of $G \times E$. In the study, we adopt the LASSO method to estimate the effects of all the markers simultaneously in the first stage. As we know, the LASSO method can shrink the marker effects estimates precisely and select the biggest effects. When we have done QTL selection by LASSO beforehand, the OLS estimates are no longer unbiased in the second stage. Therefore, the two-stage method can mitigate this issue in GS, especially in the context of $G \times E$.

Table 2: The decrement and its percentages of the MSE between the null model and the alternative model.

| Trait | Type | Null model (intercept) | Alternative model (adding components as below) | | | | |
|---|---|---|---|---|---|---|---|
| | | | $E$ | $Q$ | $Q\times E$ | $E+Q$ | $E+Q+Q\times E$ |
| FT | MSE[a] | 198.4734 | 17.0725 | 196.5320 | 9.1477 | 3.3914 | 2.6336 |
| | Decrement[b] | — | 181.4009 | 1.9414 | 189.3257 | 195.0819 | 195.8397 |
| | % of decrement[c] | — | 91.3981 | 0.9781 | 95.3910 | 98.2912 | 98.6731 |
| OC | MSE | 6.2661 | 4.4863 | 4.1189 | 1.6027 | 1.5870 | 0.7472 |
| | Decrement | — | 1.7798 | 2.1473 | 4.6635 | 4.6792 | 5.5189 |
| | % of decrement | — | 28.4040 | 34.2680 | 74.4235 | 74.6738 | 88.0751 |
| SY | MSE | 0.3503 | 0.2259 | 0.3284 | 0.1360 | 0.1331 | 0.1167 |
| | Decrement | — | 0.1245 | 0.0219 | 0.2143 | 0.2172 | 0.2336 |
| | % of decrement | — | 35.5276 | 6.2476 | 61.1769 | 62.0056 | 66.6784 |

[a]Mean square error; [b]the decrement of MSE between the null model that only includes the intercept term and the alternative model that includes both intercept and the corresponding components; [c]the percentage of the decrement of the MSE between the null model and the alternative model or, equivalently, the adjusted R-squared; FT, flowering time; OC, oil content; SY, seed yield per plant; $E$, the environment effect; $Q$, the main effect of locus; Q×E, the interaction effect between the locus and the environment.

Table 3: The prediction accuracy (standard deviation, SD) of four methods for FT.

| Environment | LASSO | OLS post-LASSO | Stratified GBLUP | $M\times E$ GBLUP |
|---|---|---|---|---|
| E7 | 0.6030 (0.0077) | 0.9008 (0.0040) | 0.6906 (0.0056) | 0.7563 (0.0049) |
| N3 | 0.7871 (0.0045) | 0.8774 (0.0036) | 0.6724 (0.0064) | 0.8656 (0.0028) |
| N4 | 0.8091 (0.0040) | 0.8924 (0.0037) | 0.7032 (0.0057) | 0.8882 (0.0026) |
| N6 | 0.7864 (0.0056) | 0.8270 (0.0048) | 0.6791 (0.0057) | 0.8512 (0.0040) |
| N7 | 0.7634 (0.0063) | 0.9048 (0.0049) | 0.6429 (0.0062) | 0.8284 (0.0053) |
| S3 | 0.8124 (0.0032) | 0.8706 (0.0037) | 0.6907 (0.0052) | 0.8536 (0.0023) |
| S4 | 0.7935 (0.0039) | 0.9333 (0.0023) | 0.7497 (0.0045) | 0.8948 (0.0025) |
| S5 | 0.7563 (0.0053) | 0.7889 (0.0047) | 0.6038 (0.0076) | 0.7659 (0.0051) |
| S6 | 0.8048 (0.0042) | 0.8858 (0.0033) | 0.6916 (0.0055) | 0.8630 (0.0031) |
| S7 | 0.6675 (0.0071) | 0.9083 (0.0031) | 0.6646 (0.0062) | 0.7796 (0.0044) |
| Average | 0.7583 (0.0052) | 0.8789 (0.0038) | 0.6789 (0.0058) | 0.8347 (0.0037) |

FT, flowering time; LASSO, least absolute shrinkage and selection operator; OLS, ordinary least squares; GBLUP, genomic best linear unbiased prediction; $M\times E$, marker × environment interaction.

Table 4: The prediction accuracy (standard deviation, SD) of four methods for OC.

| Environment | LASSO | OLS post-LASSO | Stratified GBLUP | $M\times E$ GBLUP |
|---|---|---|---|---|
| E7 | 0.7552 (0.0060) | 0.8665 (0.0041) | 0.5956 (0.0075) | 0.7784 (0.0057) |
| N3 | 0.8239 (0.0043) | 0.9097 (0.0030) | 0.6721 (0.0065) | 0.8514 (0.0033) |
| N4 | 0.7594 (0.0049) | 0.9153 (0.0027) | 0.6261 (0.0078) | 0.8048 (0.0045) |
| N6 | 0.8255 (0.0038) | 0.9130 (0.0025) | 0.7300 (0.0055) | 0.8729 (0.0030) |
| N7 | 0.7625 (0.0073) | 0.8866 (0.0044) | 0.5837 (0.0096) | 0.7859 (0.0065) |
| S3 | 0.8127 (0.0037) | 0.9175 (0.0021) | 0.6767 (0.0063) | 0.8336 (0.0034) |
| S4 | 0.7809 (0.0048) | 0.9114 (0.0028) | 0.6600 (0.0068) | 0.8372 (0.0042) |
| S5 | 0.7189 (0.0065) | 0.9030 (0.0029) | 0.5806 (0.0075) | 0.7990 (0.0040) |
| S6 | 0.7174 (0.0045) | 0.9037 (0.0028) | 0.5625 (0.0070) | 0.7846 (0.0035) |
| S7 | 0.7986 (0.0047) | 0.9188 (0.0026) | 0.6741 (0.0059) | 0.8572 (0.0034) |
| Average | 0.7755 (0.0051) | 0.9045 (0.0030) | 0.6361 (0.0070) | 0.8205 (0.0042) |

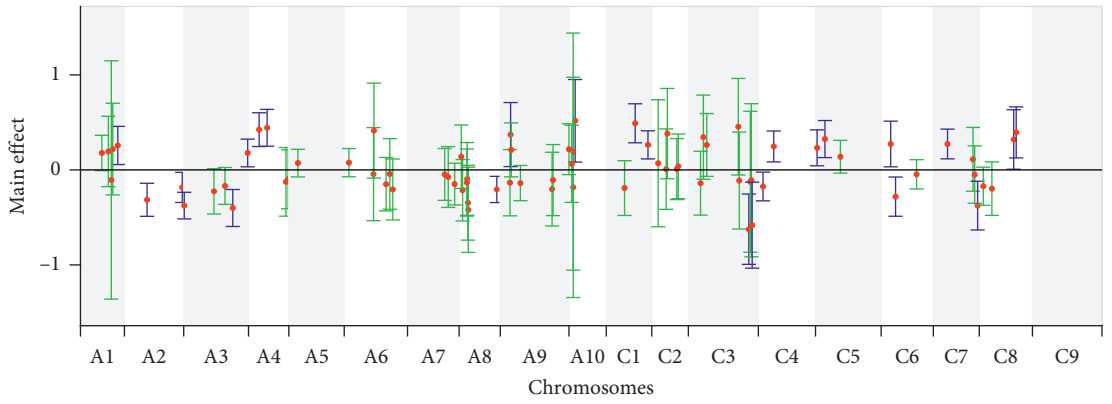
OC, oil content; LASSO, least absolute shrinkage and selection operator; OLS, ordinary least squares; GBLUP, genomic best linear unbiased prediction; $M\times E$, marker × environment interaction.

Using the TNDH population as a working example, prediction accuracy was compared for three complex traits (FT, OC, and SY) using four methods, namely, OLS post-LASSO, LASSO, $M\times E$ GBLUP model, and stratified GBLUP model. Generally, the two-stage method, that is, OLS post-LASSO, achieved the highest prediction accuracies on average across environments and also achieved the highest prediction accuracies within most of the ten environments. The $M\times E$ GBLUP model performed worse than the two-stage method but better than the other two approaches, namely, the LASSO and the stratified GBLUP model. Although LASSO performed worse than the $M\times E$ GBLUP model, the OLS after LASSO, that is, OLS post-LASSO, outperformed the $M\times E$ GBLUP model, and the improvement in prediction accuracy is significant. The results show that the OLS post-LASSO could always outperform the

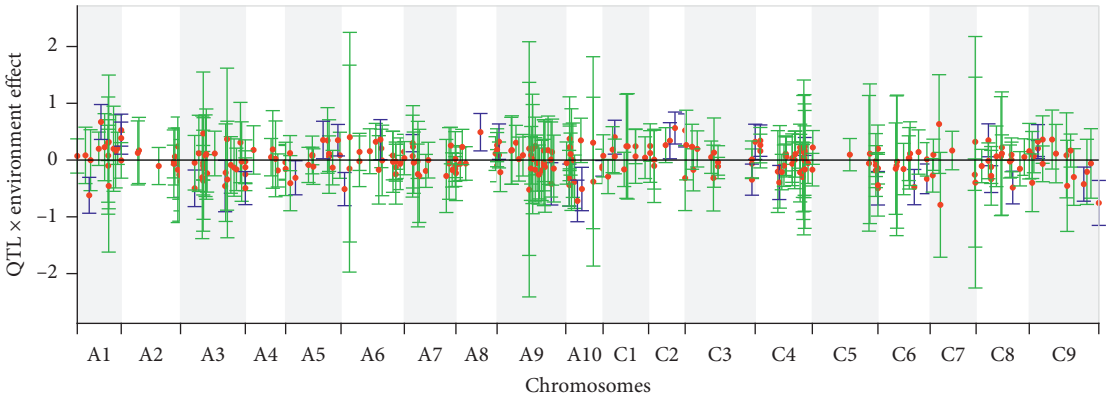
Table 5: The prediction accuracy (standard deviation, SD) of four methods for SY.

| Environment | LASSO | OLS post-LASSO | Stratified GBLUP | $M \times E$ GBLUP |
|---|---|---|---|---|
| E7 | 0.4952 (0.0082) | 0.5735 (0.0075) | 0.4418 (0.0076) | 0.4767 (0.0071) |
| N3 | 0.1600 (0.0104) | 0.6958 (0.0092) | 0.1336 (0.0097) | 0.2322 (0.0104) |
| N4 | 0.2635 (0.0115) | 0.3816 (0.0089) | 0.1509 (0.0110) | 0.3049 (0.0098) |
| N6 | 0.2811 (0.0097) | 0.7039 (0.0092) | 0.4038 (0.0089) | 0.4653 (0.0097) |
| N7 | 0.2224 (0.0106) | 0.6312 (0.0089) | 0.3251 (0.0079) | 0.4464 (0.0087) |
| S3 | 0.1351 (0.0111) | 0.4698 (0.0108) | 0.0907 (0.0110) | 0.2088 (0.0114) |
| S4 | 0.2887 (0.0109) | 0.6395 (0.0082) | 0.3056 (0.0099) | 0.4269 (0.0094) |
| S5 | 0.1232 (0.0093) | 0.2285 (0.0113) | 0.3595 (0.0106) | 0.3924 (0.0079) |
| S6 | 0.4142 (0.0100) | 0.6203 (0.0074) | 0.3973 (0.0099) | 0.5919 (0.0070) |
| S7 | 0.3351 (0.0093) | 0.5627 (0.0092) | 0.2517 (0.0099) | 0.4597 (0.0087) |
| Average | 0.2718 (0.0101) | 0.5507 (0.0091) | 0.2860 (0.0096) | 0.4005 (0.0090) |

SY, seed yield per plant; LASSO, least absolute shrinkage and selection operator; OLS, ordinary least squares; GBLUP, genomic best linear unbiased prediction; $M \times E$, marker × environment interaction.

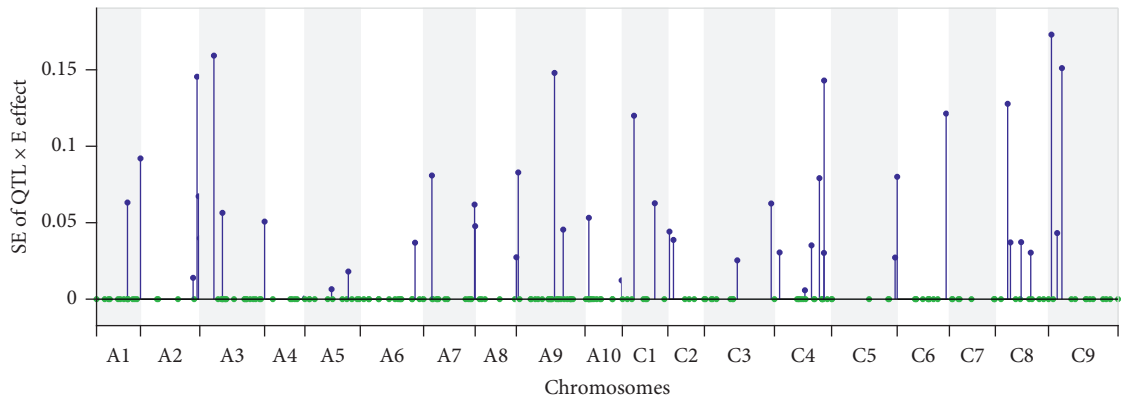



(a)




(b)

Figure 2: Continued.




(c)

Figure 2: The point estimates and 95% confidence intervals (95% CI) for marker main and environment-specific effects and the standard error (SE) of environment-specific effects for OC. (a) Point estimates (red dots) and 95% CI for main effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (b) Point estimates (red dots) and 95% CI for environment-specific effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (c) SE for environment-specific effects (the blue and green stems represent environment-specific effects found in multiple environments or only one environment, respectively).

LASSO, whether for FT, OC, or SY. Although the advantage of the two-stage method has been reported by Belloni and Chernozhukov [8], the application in GS and its benefit in increasing predictive ability are first studied in this study.

From the computational aspects of the model, the two-stage approach took around 45 minutes to compute (Windows10 Pro with a 1.6 GHz Intel Core i5-8250U processor and 8 GB of memory) in the first stage and around 0.1 seconds to compute in the second stage. The computation time is higher than that of the stratified and the $G \times E$ GBLUP models. It took around 30 seconds and 10 minutes for the stratified and the $G \times E$ GBLUP models, respectively. However, in the case of $G \times E$, the two-stage method tends to fit easily compared to traditional methods, such as the factor-analytic method. The factor-analytic method tries to simplify a complex covariance structure and, in some cases, for example, in the case of $G \times E$, difficulty in reaching convergence [32].

As a penalized regression method, LASSO was first implemented in GS by Usai et al. [33], and its prediction performance was evaluated by many studies, such as Ogutu et al. [34] and Xu et al. [35]. LASSO as well as GBLUP, the most commonly used method in GS, always outperformed other methods, such as rr-BLUP [34] and support vector machine (SVM) [35]. Based on the FT trait dataset of TNDH population, the study of Li et al. [26] indicated that the average prediction accuracies across the ten environments varied from 0.593 to 0.651 using the existing eight models: rr-BLUP, reproducing kernel Hilbert spaces (RKHS), Bayesian LASSO, BayesA, BayesB, random forest (RF), and SVM (linear kernel and Gaussian kernel). The average prediction accuracies obtained by Li et al. [26] for the FT trait were lower than those in the four methods evaluated in the present study (average correlations are 0.8789, 0.8347, 0.7583, and 0.6789 for OLS post-LASSO, $M \times E$ GBLUP model, LASSO, and the stratified GBLUP model, respectively). The stratified GBLUP model performed similarly bad to the eight models evaluated by Li et al. [26] because those methods ignore the $G \times E$ effects in the analysis. The results of our study confirmed that incorporating $G \times E$ effects into the GS model increased prediction accuracy, which has been noted by many studies, such as Lopez-Cruz et al. [20]. In particular, the two-stage method performed the best for complex traits FT, OC, and SY.

The percentage of decrement of MSE by our model (1) (corresponding to the alternative model with $E + Q + Q \times E$) is very close to 100% for the FT trait (98.6731%). This finding indicates that our proposed model fits the FT trait dataset perfectly. However, the performance is reduced when applying the same model (1) to other traits, for example, to OC (the percentage of decrement of MSE is 88.0751%) and SY (the percentage of decrement of MSE is 66.6784%). As noted by Luo et al. [27], the FT shows very high heritability; however, the SY shows low heritability. Thus, it is reasonable that our proposed model accounts for more variation of FT but less variation of SY. However, even in the case of more complex traits, that is, the OC and SY, the prediction performance of our proposed method remains superior compared with previous approaches, like the $M \times E$ GBLUP model and other models.

Although the FT is not complex as SY, the identified number of QTL for FT is larger than that for SY (Table 1). If we only focus on the number of identified QTL, there seems to be some irrationality. We can explain the issue from at least the following two aspects. First, the identified QTL by LASSO is suggestive, and further experimental identification is required. That means the detected QTL may not be the true QTL. Second, from the first subplot (a) of Figure 1, we can see that there exists a major QTL on chromosome C5 for FT, which has the largest main marker effect, and the other significant QTL (the blue lines) have smaller main marker effects than the major QTL. However, we cannot find the

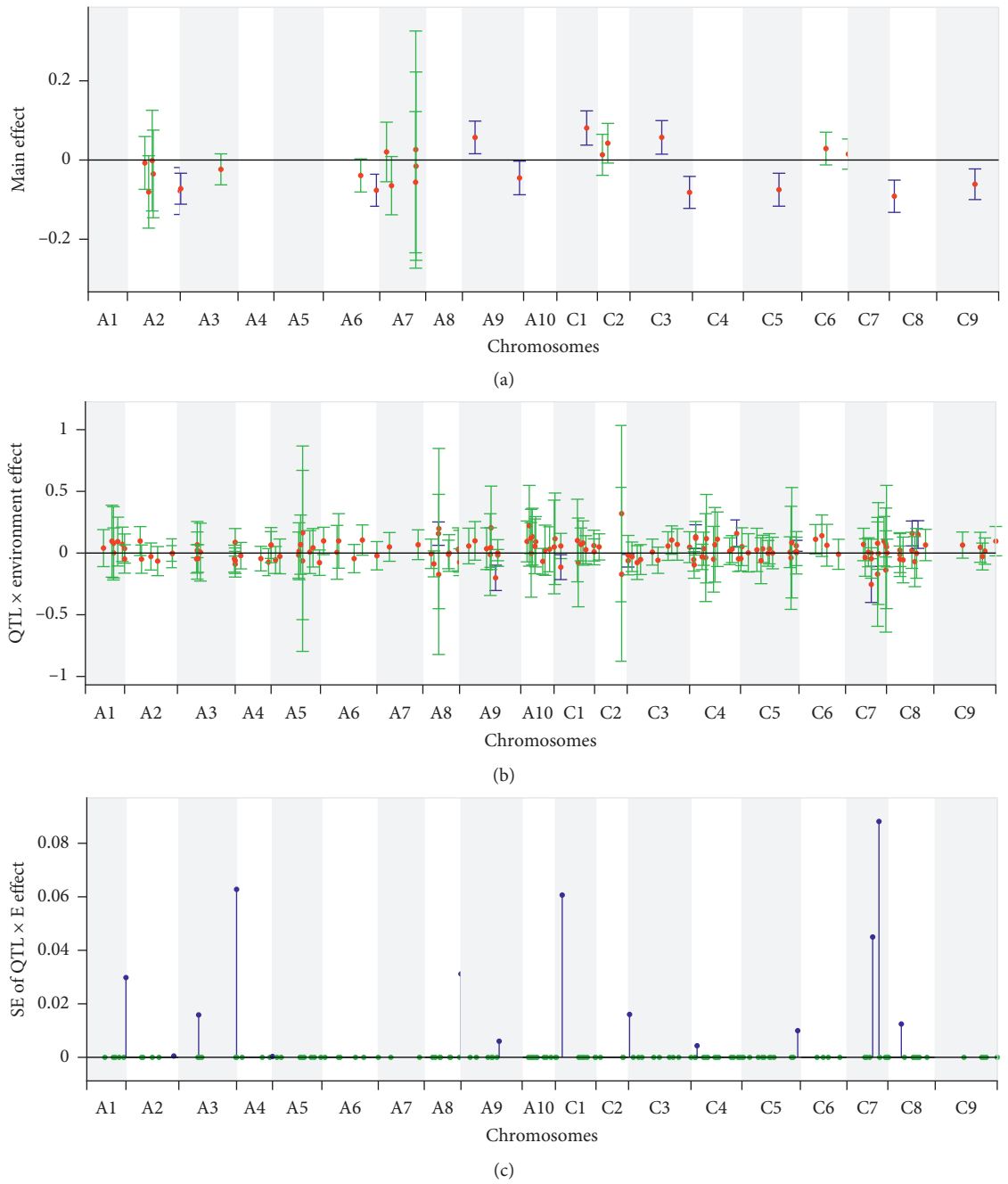



Figure 3: The point estimates and 95% confidence intervals (95% CI) for marker main and environment-specific effects and the standard error (SE) of environment-specific effects for SY. (a) Point estimates (red dots) and 95% CI for main effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (b) Point estimates (red dots) and 95% CI for environment-specific effects (the vertical blue and green CIs represent statistically significant or nonsignificant values, respectively). (c) SE for environment-specific effects (the blue and green stems represent environment-specific effects found in multiple environments or only one environment, respectively).

major QTL for OC and SY (please see subplots (a) in Figures 2 and 3). Meanwhile, from the magnitude of the absolute value of the main marker effects, we can see that it decreases from around 4 for FT to around 1.5 for OC and around 0.3 for SY. Similar patterns can be found for the environment-specific marker effects (please see the subplots (b) of Figures 1–3 for details). Thus, the finding of our study also supports that the FT trait is not as complex as OC and SY as we expected.

## Data Availability

Phenotypic and marker data used in the article can be found in Supplemental file S1.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

The authors acknowledge support by the National Natural Science Foundation of China (NSFC) (Project nos. 31970564, 11971362, 11661003, and 11661004) and funding for the Doctoral Research of ECUT under Grant no. DHBK2018052.

## Supplementary Materials

The compressed supplementary file named "S1_TN182 Phenotypic and marker data.zip" includes an Excel file named "TN182 Phenotypic and marker data.xlsx." The Excel file includes four sheets. The "Environment" sheet shows the information about the ten environments, including the name of macro environments, the code of the experiment, and so on. The "Trait name" sheet shows the names, the abbreviation, and the measurement of the three traits which are studied in the paper. The "Phenotype" sheet shows all the phenotypic values of the three traits collected from all the ten environments. The "Genotype" sheet shows the genotype matrix of all the individuals. (*Supplementary Materials*)

## References

[1] F. Khan, "Molecular markers: an excellent tool for genetic analysis," *Journal of Molecular Biomarkers & Diagnosis*, vol. 06, no. 03, p. 233, 2015.

[2] R. Lande and R. Thompson, "Efficiency of marker-assisted selection in the improvement of quantitative traits," *Genetics*, vol. 124, no. 3, pp. 743–756, 1990.

[3] T. H. Meuwissen, B. J. Hayes, and M. E. Goddard, "Prediction of total genetic value using genome-wide dense marker maps," *Genetics*, vol. 157, no. 4, pp. 1819–1829, 2001.

[4] P. Pérez, G. de Los Campos, J. Crossa, and D. Gianola, "Genomic-enabled prediction based on molecular markers and pedigree using the bayesian linear regression package in R," *The Plant Genome*, vol. 3, no. 2, pp. 106–116, 2010.

[5] A. E. Hoerl and R. W. Kennard, "ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[8] A. Belloni and V. Chernozhukov, "Least squares after model selection in high-dimensional sparse models," *Bernoulli*, vol. 19, no. 2, pp. 521–547, 2013.

[9] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity*, Chapman and Hall/CRC, New York, NY, USA, 2015.

[10] N. Meinshausen, "Relaxed lasso," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 374–393, 2007.

[11] G. de los Campos, J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. L. Calus, "Whole-genome regression and prediction methods applied to plant and animal breeding," *Genetics*, vol. 193, no. 2, pp. 327–345, 2013.

[12] C. R. Henderson and R. L. Quaas, "Multiple trait evaluation using relatives' records," *Journal of Animal Science*, vol. 43, no. 6, pp. 1188–1197, 1976.

[13] D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick, "Extension of the Bayesian alphabet for genomic selection," *BMC Bioinformatics*, vol. 12, no. 1, p. 186, 2011.

[14] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[15] G. K. Robinson, "That BLUP is a good thing: the estimation of random effects," *Statistical Science*, vol. 6, no. 1, pp. 15–32, 1991.

[16] D. Ruppert, M. P. Wand, and R. J. Carroll, *Semiparametric Regression*, Cambridge University Press, Cambridge; NY, USA, 2003.

[17] J. B. Endelman, "ridge regression and other kernels for genomic selection with R package rrBLUP," *The Plant Genome*, vol. 4, no. 3, pp. 250–255, 2011.

[18] P. M. VanRaden, "Efficient methods to compute genomic predictions," *Journal of Dairy Science*, vol. 91, no. 11, pp. 4414–4423, 2008.

[19] P. Pérez and G. de los Campos, "Genome-wide regression and prediction with the BGLR statistical package," *Genetics*, vol. 198, no. 2, pp. 483–495, 2014.

[20] M. Lopez-Cruz, J. Crossa, D. Bonnett et al., "Increased prediction accuracy in wheat breeding trials using a marker × environment interaction genomic selection model," *Genes|Genomes|Genetics*, vol. 5, no. 4, pp. 569–582, 2015.

[21] M. Bandeira E Sousa, J. Cuevas, E. G. de Oliveira Couto et al., "Genomic-enabled prediction in maize using kernel models with genotype × environment interaction," *Genes|Genomes|Genetics*, vol. 7, no. 6, pp. 1995–2014, 2017.

[22] E. Monteverde, J. E. Rosas, P. Blanco et al., "Multienvironment models increase prediction accuracy of complex traits in advanced breeding lines of rice," *Crop Science*, vol. 58, no. 4, pp. 1519–1530, 2018.

[23] D. Qiu, C. Morgan, J. Shi et al., "A comparative linkage map of oilseed rape and its use for QTL analysis of seed oil and erucic acid content," *Theoretical and Applied Genetics*, vol. 114, no. 1, pp. 67–80, 2006.

[24] J. Shi, R. Li, D. Qiu et al., "Unraveling the complex trait of crop yield with quantitative trait loci mapping in *Brassica napus*," *Genetics*, vol. 182, no. 3, pp. 851–861, 2009.

[25] Y. Zhang, C. L. Thomas, J. Xiang et al., "QTL meta-analysis of root traits in *Brassica napus* under contrasting phosphorus supply in two growth systems," *Scientific Reports*, vol. 6, no. 1, 2016.

[26] L. Li, Y. Long, L. Zhang et al., "Genome wide analysis of flowering time trait in multiple environments via high-throughput genotyping technique in *Brassica napus* L," *PLoS One*, vol. 10, no. 3, p. e0119425, Article ID e0119425, 2015.

[27] Z. Luo, M. Wang, Y. Long et al., "Incorporating pleiotropic quantitative trait loci in dissection of complex traits: seed yield in rapeseed as an example," *Theoretical and Applied Genetics*, vol. 130, no. 8, pp. 1569–1585, 2017.

[28] M. Zhang, K. L. Montooth, M. T. Wells, A. G. Clark, and D. Zhang, "Mapping multiple quantitative trait loci by bayesian classification," *Genetics*, vol. 169, no. 4, pp. 2305–2318, 2005.

[29] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[30] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, Austria, 2017.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[32] L. S. Peixouto, J. A. R. Nunes, and D. F. Furtado, "Factor analysis applied to the G+GE matrix via REML/BLUP for multi-environment data," *Crop Breeding and Applied Biotechnology*, vol. 16, no. 1, pp. 1–6, 2016.

[33] M. G. Usai, M. E. Goddard, and B. J. Hayes, "LASSO with cross-validation for genomic selection," *Genetics Research*, vol. 91, no. 6, pp. 427–436, 2009.

[34] J. O. Ogutu, T. Schulz-Streeck, and H. P. Piepho, "Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions," *BMC*, vol. 6, no. Suppl 2, 2012.

[35] Y. Xu, X. Wang, X. Ding et al., "Genomic selection of agronomic traits in hybrid rice using an NCII population," *Rice*, vol. 11, no. 1, p. 32, 2018.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 9746538, 7 pages
https://doi.org/10.1155/2020/9746538

*Research Article*

# Optimization for Due-Window Assignment Scheduling with Position-Dependent Weights

**Li-Yan Wang,[1] Dan-Yang Lv,[1] Bo Zhang,[1] Wei-Wei Liu,[2] and Ji-Bo Wang[1]**

[1]*School of Science, Shenyang Aerospace University, Shenyang 110136, China*
[2]*Department of Science, Shenyang Sport University, Shenyang 110102, China*

Correspondence should be addressed to Wei-Wei Liu; liuww_2010@163.com and Ji-Bo Wang; wangjibo75@163.com

Received 14 May 2020; Accepted 16 June 2020; Published 22 July 2020

Academic Editor: Chin-Chia Wu



This paper considers a single-machine due-window assignment scheduling problem with position-dependent weights, where the weights only depend on their position in a sequence. The objective is to minimise the total weighted penalty of earliness, tardiness, due-window starting time, and due-window size of all jobs. Optimal properties of the problem are given, and then, a polynomial-time algorithm is provided to solve the problem. An extension to the problem is offered by assuming general position-dependent processing time.

## 1. Introduction

Conventionally, in scheduling theory, due-windows are job-dependent either if they are dictated by the customer (i.e., given constants) or they are decision variables (i.e., due-window assignment). A due-window for job $J_i$ is defined by a due-window starting time $d_i^1$ and a due-window finishing time $d_i'$, i.e., the due-window $[d_i^1, d_i']$, and the due-window size is $D_i = d_i' - d_i^1$. In the just-in-time (JIT) production methodology and scheduling theory, setting proper due-windows is challenging (see Gong et al. [1], Janiak et al. [2], and Geng et al. [3]). In the literature, three very popular due-window assignment methods are studied:

Common due-window (CON-DW) assignment method (Liman et al. [4, 5]): all jobs are assigned a common due-window, i.e., all the jobs have a common due-window $[d^1, d']$, where $d_i^1 = d^1$, $d_i' = d'$, the due-window size of all the jobs is $D = d' - d^1$, and both $d^1$ and $D$ are decision variables. In the literature, most studies considered the CON-DW assignment method, e.g., Mosheiov and Sarig [6] addressed a minmax CON-DW assignment problem, the objective of which is to minimise the largest cost among earliness, tardiness, due-window starting time, and due-window size. They proved that the single-machine and two-machine flow-shop problems can be solved in polynomial time. They also proved that the cases of parallel identical machines and uniform machines are NP-hard. Yin et al. [7] considered the batch delivery scheduling problem with an assignable common due-window on a single machine. Yin et al. [8] studied the single-machine scheduling problem with CON-DW assignment and batch delivery cost. Liu et al. [9] considered the single-machine CON-DW assignment scheduling problem with deteriorating jobs. For the weighted sum of earliness, tardiness, and due-window location penalty minimization, they proposed a polynomial-time algorithm to solve the problem. Wang and Wang [10] considered the single-machine resource allocation scheduling problem with learning effect and CON-DW assignment.

Slack due-window (SLK-DW) assignment method (Mosheiov and Oron [11]) is $d_i^1 = p_i + q^1$, $d_i' = p_i + q'$, and $D_i = d_i' - d_i^1 = q' - q^1 = D$, where $p_i$ is the normal processing time of job $J_i$ and $q^1$ and $D$ are decision variables. Wang et al. [12] considered the single-machine SLK-DW assignment scheduling problem with deteriorating jobs and learning effect. Ji et al. [13] considered the single-machine SLK-DW assignment scheduling problem with group technology. Yin et al. [14], Yin et al. [15], and Wang et al. [16] considered SLK-DW assignment scheduling problems with

resource allocation (controllable processing time). Mor and Mosheiov [17] considered SLK-DW assignment proportionate flow-shop scheduling problems.

Different due-windows' (DIF-DW) assignment method: it is assumed that the job $J_i$ has a due-window $[d_i^1, d_i']$, where $d_i^1 \geq 0$ and $d_i' \geq 0\,(d_i^1 \leq d_i')$ denote the starting time and finishing time of the due-window, respectively. The due-window size of the job $J_i$ is $D_i = d_i' - d_i^1$, and both $d_i^1$ and $D_i$ are decision variables. Wang et al. [12] considered DIF-DW assignment scheduling problems with deteriorating jobs and learning effect.

In a recent paper, Wang et al. [18] considered CON-DW and SLK-DW assignment methods with position-dependent weights, i.e., the weight does not correspond with the job but with the position in which some job is scheduled. They proved that both these due-window assignment methods with position-dependent weights can be solved in polynomial time, respectively. *"The scheduling with due-window assignment has many real-world applications. For example, the due-window might reflect an assembly environment in which the components of the product should be ready within a time interval in order to avoid staging delays or a shop where several jobs constitute a single customer's order. It is clear that a wide due-window increases the supplier's production and delivery flexibility. However, a large due-window and delaying job completion reduce the supplier's competitiveness and customer service level"* (Yang et al. [19]). It is natural and interesting to continue the work of Wang et al. [18] but study the DIF-DW assignment scheduling problem with position-dependent weights. The contributions of this paper are given as follows: (1) the structural properties of scheduling problems are derived; (2) the total weighted penalty of earliness, tardiness, due-window starting time, and due-window size of all jobs' minimization can be solved in polynomial time; and (3) it is further extended the model to the case with general position-dependent processing time. We refer the reader to the survey of Janiak et al. [2] on the scheduling problems with (CON-DW, SLK-DW, and DIF-DW) due-windows.

The remainder of the paper is organized as follows. In Section 2, we formulate the problem. Section 3 gives some results and an optimal policy for the proposed problem. An extension of the proposed problem is given in Section 4. Finally, the conclusion and future work are given.

## 2. Problem Description

A set of $n$ jobs $\tilde{N} = \{J_1, J_2, \ldots, J_n\}$ needs to be processed on a single machine. All the independent jobs are available at time zero, and preemption is not allowed. For a given sequence, we assume that job $J_i$ has a due-window $[d_i^1, d_i']$, where $d_i^1 \geq 0$ $(d_i' \geq 0)$ denote the starting time (finishing time) of the due-window, $d_i^1 \leq d_i'$. The due-window size of job $J_i$ is defined by $D_i = d_i' - d_i^1$, and $d_i^1$ and $D_i$ of all jobs are decision variables. The normal processing time of job $J_i$ is denoted by $p_i$ (i.e., the processing time without being influenced by any factor), $i = 1, 2, \ldots, n$. For a given sequence, let $C_i$ be the completion time of job $J_i$. The aim is to find the optimal starting time of the due-windows, the size of the due-windows, and the sequence of jobs $\delta$ such that the following measure is minimized:

$$Z\left(d_i^1, D_i, \delta(i)\right) = \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d_{\delta(i)}^1 + \psi_{n+1} D_{\delta(i)}, \quad (1)$$

where $\delta(i)$ denotes the job scheduled in the $i$th position, $\psi_i > 0$ $(i = 1, 2, \ldots, n)$ denote a position-dependent weight (i.e., weight $\psi_i$ does not correspond with the job but with the position in which some job is scheduled), $\psi_0$ $(\psi_{n+1})$ is the unit cost of $d_{\delta(i)}^1$ $(D_{\delta(i)})$, $L_{\delta(i)}$ is the earliness-tardiness of job $J_{\delta(i)}$ $(i = 1, 2, \ldots, n)$, and

$$L_{\delta(i)} = \begin{cases} d_{\delta(i)}^1 - C_{\delta(i)}, & \text{for } d_{\delta(i)}^1 > C_{\delta(i)}, \\ 0, & \text{for } d_{\delta(i)}^1 \leq C_{\sigma(i)} \leq d_{\delta(i)}', \\ C_{\delta(i)} - d_{\delta(i)}', & \text{for } C_{\delta(i)} > d_{\delta(i)}'. \end{cases} \quad (2)$$

Using the three-field notation (Graham et al. [20]), the problem studied here is $1|\text{DIF} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d_{\delta(i)}^1 + \psi_{n+1} D_{\delta(i)}$. Wang et al. [18] considered single-machine scheduling problems with common due-window (CON-DW) and slack due-window (SLK-DW) assignments. They proved that the problems $1|\text{CON} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1 + \psi_{n+1} D$ and $1|\text{SLK} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 q^1 + \psi_{n+1} D$ can be solved in $O(n \log n)$ time, respectively.

## 3. Main Results

Obviously, there exists an optimal sequence $\delta^*$ without any machine idle time between the processing of jobs, and the first job in the sequence starts at time zero.

**Lemma 1.** *There exists an optimal sequence such that* $d_{\delta(i)}^1 \leq d_{\delta(i)}' \leq C_{\delta(i)}$.

*Proof.* We consider two cases that contradict this optimal property:

Case i: if $d_{\delta(i)}^1 \leq C_{\delta(i)} < d_{\delta(i)}'$, then the total cost for job $J_{\delta(i)}$ is

$$z_{\delta(i)} = \psi_0 d_{\delta(i)}^1 + \psi_{n+1}\left(d_{\delta(i)}' - d_{\delta(i)}^1\right). \quad (3)$$

We shift $d_{\delta(i)}'$ to the left such that $d_{\delta(i)}' = C_{\delta(i)}$, and we have

$$\widetilde{z}_{\delta(i)} = \psi_0 d^1_{\delta(i)} + \psi_{n+1}\left(C_{\delta(i)} - d^1_{\delta(i)}\right) < z_{\delta(i)}. \quad (4)$$

Hence, Case i is not an optimal due-window assignment.

Case ii: if $C_{\delta(i)} < d^1_{\delta(i)} \le d'_{\delta(i)}$, then the total cost for job $J_{\delta(i)}$ is

$$z_{\delta(i)} = \psi_i\left(d^1_{\delta(i)} - C_{\delta(i)}\right) + \psi_0 d^1_{\delta(i)} + \psi_{n+1}\left(d'_{\delta(i)} - d^1_{\delta(i)}\right). \quad (5)$$

We shift $d^1_{\delta(i)}$ and $d'_{\delta(i)}$ to the left such that $d^1_{\delta(i)} = d_{\delta(i)}{}' = C_{\delta(i)}$, and we have

$$\widetilde{z}_{\delta(i)} = \psi_0 C_{\delta(i)} < z_{\delta(i)}. \quad (6)$$

Hence, Case ii is not an optimal due-window assignment.

To summarise, we have $d^1_{\delta(i)} \le d'_{\delta(i)} \le C_{\delta(i)}$. □

**Lemma 2.** *For a given sequence $\delta$, the optimal due-window locations $d^1_{\delta(i)}$ and $d_{\delta(i)}{}'$ for job $J_{\delta(i)}$ can be obtained as follows:*

*(1) When* $\min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_i$, *then set* $d^1_{\delta(i)} = d'_{\delta(i)} = 0$

*(2) When* $\min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_0$, *then set* $d^1_{\delta(i)} = d'_{\delta(i)} = C_{\delta(i)}$

*(3) When* $\min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_{n+1}$, *then set* $d^1_{\delta(i)} = 0$ *and* $d'_{\delta(i)} = C_{\delta(i)}$

*Proof*

(1) When $\min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_i$ and $d^1_{\delta(i)} = d'_{\delta(i)} = 0$, we have $z_{\delta(i)} = \psi_i C_{\delta(i)}$

From Lemma 1, we consider the following two cases:

Case i: if $d^1_{\delta(i)} \le C_{\delta(i)} \le d'_{\delta(i)}$, then the total cost for job $J_{\delta(i)}$ is $\widetilde{z}_{\delta(i)} = \psi_0 d^1_{\delta(i)} + \psi_{n+1}(d'_{\delta(i)} - d^1_{\delta(i)}) \ge \psi_i d^1_{\delta(i)} + \psi_i(d'_{\delta(i)} - d^1_{\delta(i)}) = \psi_i d'_{\delta(i)} \ge \psi_i C_{\delta(i)} = z_{\delta(i)}$

Case ii: if $d^1_{\delta(i)} \le d'_{\delta(i)} \le C_{\delta(i)}$, then the total cost for job $J_{\delta(i)}$ is $\widetilde{z}_{\delta(i)} = \psi_i(C_{\delta(i)} - d'_{\delta(i)}) + \psi_0 d^1_{\delta(i)} + \psi_{n+1}(d'_{\delta(i)} - d^1_{\delta(i)}) \ge \psi_i(C_{\delta(i)} - d'_{\delta(i)}) + \psi_i d^1_{\delta(i)} + \psi_i(d'_{\delta(i)} - d^1_{\delta(i)}) = \psi_i C_{\delta(i)} = z_{\delta(i)}$

To summarise, if $\min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_i$, then set $d^1_{\delta(i)} = d'_{\delta(i)} = 0$.

Similarly, cases (2) and (3) can be proved. □

**Lemma 3.** *For a given sequence $\delta$, the optimal due-window locations $d^1_{\delta(i)}$ and $d'_{\delta(i)}$ for job $J_{\delta(i)}$ can be obtained as follows:*

*(1) When* $\psi_i = \psi_0 < \psi_{n+1}$, *then set* $d^1_{\delta(i)} = d'_{\delta(i)} = C_{\delta(j)}$, *where* $j = 0, 1, \ldots, i$

*(2) When* $\psi_i = \psi_{n+1} < \psi_0$, *then set* $d^1_{\delta(i)} = 0, d'_{\delta(i)} = C_{\delta(j)}$, *where* $j = 0, 1, \ldots, i$

*(3) When* $\psi_0 = \psi_{n+1} < \psi_i$, *then set* $d^1_{\delta(i)} = C_{\delta(j)}$ *and* $d'_{\delta(i)} = C_{\delta(i)}$, *where* $j = 0, 1, \ldots, i$

*(4) When* $\psi_0 = \psi_{n+1} = \psi_i$, *then set* $d^1_{\delta(i)} = C_{\delta(j_1)}$ *and* $d'_{\delta(i)} = C_{\delta(j_2)}$, *where* $j_1 = 0, 1, \ldots, i$ *and* $j_2 = j_1, j_1 + 1, \ldots, i$

*Proof.* The proof is similar to the proof of Lemma 2. □

**Lemma 4.** *The optimal sequence of the problem $1|DIF - DW|\sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ can be obtained by sequencing the jobs in a nondecreasing order of $p_i$, i.e., the smallest processing time (SPT) first rule.*

*Proof.* From Lemmas 1–3, the objective function $\sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ can be transformed into the following three cases: (1) $\sum_{i=1}^{n} \psi_i C_{\delta(i)}$; (2) $\psi_0 C_{\delta(i)}$; and (3) $\psi_{n+1} C_{\delta(i)}$. For all the three cases, it is easy to verify (by the pairwise interchange method) that sequencing the jobs in a nondecreasing order of $p_i$ is optimal.

Let $A = \{i \mid \min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_i, \quad i = 1, 2, \ldots, n\} \cup \{i \mid \psi_i = \psi_0 < \psi_{n+1}, \quad i = 1, 2, \ldots, n\} \cup \{i \mid \psi_i = \psi_{n+1} < \psi_0, \quad i = 1, 2, \ldots, n\} \cup \{i \mid \psi_i = \psi_0 = \psi_{n+1}, \quad i = 1, 2, \ldots, n\}$, $B = \{i \mid \min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_0, \quad i = 1, 2, \ldots, n\} \cup \{i \mid \psi_0 = \psi_{n+1} < \psi_i, \quad i = 1, 2, \ldots, n\}$, and $C = \{i \mid \min\{\psi_i, \psi_0, \psi_{n+1}\} = \psi_{n+1}, \quad i = 1, 2, \ldots, n\}$; then,

$$\begin{aligned} Z\left(d^1_i, D_i, \delta\right) &= \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)} \\ &= \sum_{i\in A} \psi_i \sum_{j=1}^{i} p_{\delta(j)} + \sum_{i\in B} \psi_i \sum_{j=1}^{i} p_{\delta(j)} + \sum_{i\in C} \psi_i \sum_{j=1}^{i} p_{\delta(j)} \\ &= \sum_{i=1}^{n} \psi'_i \sum_{j=1}^{i} p_{\delta(j)} \\ &= \sum_{i=1}^{n} p_{\delta(i)} \sum_{j=i}^{n} \psi'_i \\ &= \sum_{i=1}^{n} \lambda_i p_{\delta(i)}, \end{aligned} \quad (7)$$

where

$$\psi_i' = \begin{cases} \psi_i, & i \in A, \\ \psi_0, & i \in B, \\ \psi_{n+1}, & i \in C. \end{cases} \tag{8}$$

And

$$\lambda_i = \sum_{j=i}^{n} \psi_i'. \tag{9}$$

□

*Remark 1.* Obviously, $\lambda_i = \sum_{j=i}^{n} \psi_i'$ is a decreasing function on $i$; from Hardy et al. [21], the optimal sequence can be obtained by the SPT rule, and it is the same as Lemma 4.

From Lemmas 1–4, a polynomial-time algorithm can be proposed for the $1|\text{DIF} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ problem.

**Theorem 1.** *Algorithm 1 solves the problem* $1|DI\,F - DW| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ *in* $O(n \log n)$ *time.*

*Proof.* Optimality can be guaranteed by Lemmas 1–4. In Algorithm 1, Step 1 needs $O(n \log n)$ time by the SPT rule; Steps 2 and 3 can be performed in $O(n)$ time. Thus, the total time for Algorithm 1 is $O(n \log n)$.

In order to illustrate Algorithm 1 for the problem $1|\text{DIF} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$, we present the following instance. □

*Example 1.* The data are as follows: $n = 10$, $p_1 = 15$, $p_2 = 20$, $p_3 = 26$, $p_4 = 24$, $p_5 = 17$, $p_6 = 28$, $p_7 = 21$, $p_8 = 25$, $p_9 = 27$, $p_{10} = 14$, $\psi_0 = 14$, $\psi_1 = 7$, $\psi_2 = 20$, $\psi_3 = 12$, $\psi_4 = 24$, $\psi_5 = 14$, $\psi_6 = 22$, $\psi_7 = 15$, $\psi_8 = 8$, $\psi_9 = 19$, $\psi_{10} = 12$, and $\psi_{11} = 50$.

Now, we can solve the problem $1|\text{DIFW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ according to Algorithm 1 as follows:

Step 1: according to Lemma 4, the optimal sequence is $\delta^* = (J_{10}, J_1, J_5, J_2, J_7, J_4, J_8, J_3, J_9, J_6)$

Step 2: for the optimal sequence $\delta^* = (J_{10}, J_1, J_5, J_2, J_7, J_4, J_8, J_3, J_9, J_6)$, the completion time of all jobs is $C_{10} = 14$, $C_1 = 29$, $C_5 = 46$, $C_2 = 66$, $C_7 = 87$, $C_4 = 111$, $C_8 = 136$, $C_3 = 162$, $C_9 = 189$, and $C_6 = 217$, and the optimal due-window locations $d^1_{\delta(i)}$ and $d_{\delta(i)}'$ for each job are given in Table 1

Step 3: the optimal due-window sizes are $D_{\delta(i)} = 0$ $(i = 1, 2, \ldots, 10)$, $\lambda_1 = 123$, $\lambda_2 = 116$, $\lambda_3 = 102$, $\lambda_4 = 90$, $\lambda_5 = 76$, $\lambda_6 = 62$, $\lambda_7 = 48$, $\lambda_8 = 34$, $\lambda_9 = 26$, and $\lambda_{10} = 12$, and the objective function is $Z = \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)} = \sum_{i=1}^{n} \lambda_i p_{\delta(i)} = 13202$.

## 4. An Extension

In this section, the problem $1|\text{DIF} - \text{DW}| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ is extended to a setting of general position-dependent processing time. Let $p_i^A$ be the actual processing time of $J_i$; under the general position-dependent processing time setting, the actual processing time of $J_i$ is $p_i^A = \theta(i, r)$ if it is assigned to position $r$, $i, r = 1, \ldots, n$. Thus, the input for the problem contains a matrix of $(n \times n)$ job-position values. Biskup [22] introduced a job-independent learning effect model in which $p_i^A = \theta(i, r) = p_i r^{\alpha}$, where $\alpha \le 0$ is the learning index (see also Wang et al. [23]). Mosheiov and Sidney [24] introduced job-dependent learning effects, i.e., $p_i^A = \theta(i, r) = p_i r^{\alpha_i}$, where $\alpha_i \le 0$ is the job-dependent learning index of job $J_i$. Wang et al. [25] introduced truncated job-dependent learning effects, i.e., $p_i^A = \theta(i, r) = p_i \max\{r^{\alpha_i}, \beta\}$, where $0 < \beta < 1$ is a truncation parameter. We refer the reader to the survey of Azzouz et al. [26] on scheduling problems with learning effects.

From (7), we have

$$Z\left(d^1_i, D_i, \delta\right) = \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)} = \sum_{i=1}^{n} \lambda_i \theta(i, r), \tag{10}$$

where $\lambda_i$ are given by (9).

From (10), the optimal sequence of the problem $1|\text{DIF} - \text{DW}, p_i^A = \theta(i, r)| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ can be obtained by solving the following assignment problem:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} \sum_{r=1}^{n} \lambda_r \theta(i, r) x_{ir}, \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ir} = 1, \quad r = 1, \ldots, n, \\ & \sum_{r=1}^{n} x_{ir} = 1, \quad i = 1, \ldots, n, \end{aligned} \tag{11}$$

where $\lambda_r, r = 1, \ldots, n$, are given by (9), and

$$x_{ir} = \begin{cases} 1, & \text{if job } J_i \text{ is assigned to position } r, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Based on the above analysis, the solution procedure of the problem $1|\text{DIF} - \text{DW}, p_i^A = \theta(i, r)| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ can be summarized as follows.

**Theorem 2.** *Algorithm 2 solves the problem* $1|DIF - DW, p_i^A = \theta(i, r)| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ *in* $O(n^3)$ *time.*

*Proof.* Optimality is guaranteed by Lemmas 1–3 and the above analysis. In Algorithm 2, Step 1 needs $O(n^3)$ time by the SPT rule; Steps 2 and 3 can be performed in $O(n)$ time. Thus, the total time for Algorithm 2 is $O(n^3)$. In order to illustrate Algorithm 2 for the problem $1|\text{DIF} - \text{DW}, p_i^A = \theta(i, r)| \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$, we present the following instance. □

*Example 2.* The data are as follows: $n = 8$, $\psi_0 = 14$, $\psi_1 = 8$, $\psi_2 = 18$, $\psi_3 = 12$, $\psi_4 = 24$, $\psi_5 = 10$, $\psi_6 = 20$, $\psi_7 = 15$, $\psi_8 = 7$, and $\psi_9 = 21$. The job-dependent processing time is given in Table 2.

Step 1: obtain the optimal sequence by the SPT rule (see Lemma 4)
Step 2: calculate the completion time of each job under the optimal sequence, and determine the optimal due-window locations $d^1_{\delta(i)}$ and $d'_{\delta(i)}$ for each job according to Lemmas 2 and 3
Step 3: obtain the optimal due-window size by setting $D_{\delta(i)} = d'_{\delta(i)} - d^1_{\delta(i)}$ $(i = 1, 2, \ldots, n)$, and calculate the objective function $\sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ by equation (7)

Algorithm 1

Table 1: Results of the optimal due-window location.

| Job $J_{\delta(i)}$ | Job $J_{\delta(i)}$ | $d^1_{\delta(i)}$ | $d_{\delta(i)}{}'$ |
|---|---|---|---|
| $J_{\delta(1)}$ | $J_{10}$ | $d^1_{\delta(i)} = 0$ | $d_{\delta(1)}{}' = 0$ |
| $J_{\delta(2)}$ | $J_1$ | $d^1_{\delta(2)} = C_{\delta(2)} = 29$ | $d_{\delta(2)}{}' = C_{\delta(2)} = 29$ |
| $J_{\delta(3)}$ | $J_5$ | $d^1_{\delta(3)} = 0$ | $d_{\delta(3)}{}' = 0$ |
| $J_{\delta(4)}$ | $J_2$ | $d^1_{\delta(4)} = C_{\delta(4)} = 66$ | $d_{\delta(4)}{}' = C_{\delta(4)} = 66$ |
| $J_{\delta(5)}$ | $J_7$ | $d^1_{\delta(5)} \in \{C_{\delta(i)} \mid i = 0, 1, 2, 3, 4, 5\}$ i.e., $d^1_{\delta(5)} \in \{0, 14, 29, 46, 66, 87\}$ | $d_{\delta(5)}{}' = d^1_{\delta(5)}$ |
| $J_{\delta(6)}$ | $J_4$ | $d^1_{\delta(6)} = C_{\delta(6)} = 111$ | $d_{\delta(6)}{}' = C_{\delta(6)} = 111$ |
| $J_{\delta(7)}$ | $J_8$ | $d^1_{\delta(7)} = C_{\delta(7)} = 136$ | $d_{\delta(7)}{}' = C_{\delta(7)} = 136$ |
| $J_{\delta(8)}$ | $J_3$ | $d^1_{\delta(8)} = 0$ | $d_{\delta(8)}{}' = 0$ |
| $J_{\delta(9)}$ | $J_9$ | $d^1_{\delta(9)} = C_{\delta(9)} = 189$ | $d_{\delta(9)}{}' = C_{\delta(9)} = 189$ |
| $J_{\delta(10)}$ | $J_6$ | $d^1_{\delta(10)} = 0$ | $d_{\delta(10)}{}' = 0$ |

Step 1: solve assignment problem (11) to obtain the optimal sequence
Step 2: calculate the completion time of each job under the optimal sequence, and determine the optimal due-window locations $d^1_{\delta(i)}$ and $d'_{\delta(i)}$ for each job according to Lemmas 2 and 3
Step 3: obtain the optimal due-window size by setting $D_{\delta(i)} = d'_{\delta(i)} - d^1_{\delta(i)}$ $(i = 1, 2, \ldots, n)$, and calculate the objective function $\sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)}$ by assignment problem (11)

Algorithm 2

Table 2: Date of Example 2.

| $J_i$ | $r$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $J_1$ | 6 | 7 | 11 | 5 | 6 | 22 | 13 | 21 |
| $J_2$ | 8 | 14 | 7 | 8 | 11 | 17 | 12 | 6 |
| $J_3$ | 9 | 11 | 13 | 32 | 7 | 10 | 12 | 8 |
| $J_4$ | 17 | 22 | 19 | 10 | 5 | 9 | 13 | 7 |
| $J_5$ | 16 | 8 | 15 | 14 | 11 | 17 | 13 | 14 |
| $J_6$ | 18 | 17 | 31 | 14 | 8 | 23 | 15 | 20 |
| $J_7$ | 15 | 12 | 18 | 19 | 8 | 16 | 21 | 13 |
| $J_8$ | 13 | 17 | 24 | 16 | 18 | 16 | 13 | 15 |

Table 3: Results of the optimal due-window location.

| Job $J_{\delta(i)}$ | Job $J_{\delta(i)}$ | $d^1_{\delta(i)}$ | $d_{\delta(i)}{}'$ |
|---|---|---|---|
| $J_{\delta(1)}$ | $J_3$ | $d^1_{\delta(1)} = 0$ | $d_{\delta(1)}{}' = 0$ |
| $J_{\delta(2)}$ | $J_5$ | $d^1_{\delta(2)} = C_{\delta(2)} = 17$ | $d_{\delta(2)}{}' = C_{\delta(2)} = 17$ |
| $J_{\delta(3)}$ | $J_2$ | $d^1_{\delta(3)} = 0$ | $d_{\delta(3)}{}' = 0$ |
| $J_{\delta(4)}$ | $J_1$ | $d^1_{\delta(4)} = C_{\delta(4)} = 29$ | $d_{\delta(4)}{}' = C_{\delta(4)} = 29$ |
| $J_{\delta(5)}$ | $J_6$ | $d^1_{\delta(5)} = 0$ | $d_{\delta(5)}{}' = 0$ |
| $J_{\delta(6)}$ | $J_4$ | $d^1_{\delta(6)} = C_{\delta(6)} = 46$ | $d_{\delta(6)}{}' = C_{\delta(6)} = 46$ |
| $J_{\delta(7)}$ | $J_8$ | $d^1_{\delta(7)} = C_{\delta(7)} = 59$ | $d_{\delta(7)}{}' = C_{\delta(7)} = 59$ |
| $J_{\delta(8)}$ | $J_7$ | $d^1_{\delta(8)} = 0$ | $d_{\delta(8)}{}' = 0$ |

## 5. Conclusion and Future Work

This study addressed the due-window (DIF-DW) assignment scheduling problem under the consideration of position-dependent weights. The goal is to determine the optimal sequence, the optimal due-window location, and size such that the total penalty (including the earliness, tardiness, due-window starting time, and due-window size of all jobs) is minimized. It was proved that the problem can be solved in polynomial time. The proposed model was also extended to the general position-dependent processing time, and the polynomial-time solution was provided. Further extensions are considering the above problems in the setting of $m$-machine flow-shop and $m$-identical (unrelated) parallel machines (Hsu and Liao [27]), studying the scheduling with two-agent resource-dependent release time (Liu and Duan [28]), or investigating scheduling with rate-modifying activity under deterioration effect (Xue and Zhang [29]).

Step 1: by (9), we have $\lambda_1 = 93$, $\lambda_2 = 85$, $\lambda_3 = 71$, $\lambda_4 = 59$, $\lambda_5 = 45$, $\lambda_6 = 35$, $\lambda_7 = 21$, and $\lambda_8 = 7$. According to assignment problem (11), the optimal sequence is $\delta^* = (J_3, J_5, J_2, J_1, J_6, J_4, J_8, J_7)$.

Step 2: for the optimal sequence $\delta^* = (J_3, J_5, J_2, J_1, J_6, J_4, J_8, J_7)$, the completion time of all jobs is $C_3 = 9$, $C_5 = 17$, $C_2 = 24$, $C_1 = 29$, $C_6 = 37$, $C_4 = 46$, $C_8 = 59$, and $C_7 = 72$, and the optimal due-window locations $d^1_{\delta(i)}$ and $d_{\delta(i)}{}'$ for each job are given in Table 3.

Step 3: the optimal due-window sizes are $D_{\delta(i)} = 0$ $(i = 1, 2, \ldots, 8)$, and the objective function is $Z = \sum_{i=1}^{n} \psi_i L_{\delta(i)} + \psi_0 d^1_{\delta(i)} + \psi_{n+1} D_{\delta(i)} = 3348$.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation of Liaoning Province (2020-MS-233).

## References

[1] H. Gong, B. Zhang, and W. Peng, "Scheduling and common due date assignment on a single parallel-batching machine with batch delivery," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 464390, 7 pages, 2015.

[2] A. Janiak, W. A. Janiak, T. Krysiak, and T. Kwiatkowski, "A survey on scheduling problems with due windows," *European Journal of Operational Research*, vol. 242, no. 2, pp. 347–357, 2015.

[3] X.-N. Geng, J.-B. Wang, and D. Bai, "Common due date assignment scheduling for a no-wait flowshop with convex resource allocation and learning effect," *Engineering Optimization*, vol. 51, no. 8, pp. 1301–1323, 2019.

[4] S. D. Liman, S. S. Panwalkar, and S. Thongmee, "Determination of common due window location in a single machine scheduling problem," *European Journal of Operational Research*, vol. 93, no. 1, pp. 68–74, 1996.

[5] S. D. Liman, S. S. Panwalkar, and S. Thongmee, "Common due window size and location determination in a single machine scheduling problem," *The Journal of the Operational Research Society*, vol. 49, no. 9, pp. 1007–1010, 1998.

[6] G. Mosheiov and A. Sarig, "Minmax scheduling problems with a common due-window," *Computers & Operations Research*, vol. 36, no. 6, pp. 1886–1892, 2009.

[7] Y. Yin, T. C. E. Cheng, C.-J. Hsu, and C.-C. Wu, "Single-machine batch delivery scheduling with an assignable common due window," *Omega*, vol. 41, no. 2, pp. 216–225, 2013.

[8] Y. Yin, T. C. E. Cheng, J. Wang, and C.-C. Wu, "Single-machine common due window assignment and scheduling to minimize the total cost," *Discrete Optimization*, vol. 10, no. 1, pp. 42–53, 2013.

[9] J. Liu, Y. Wang, and X. Min, "Single-machine scheduling with common due-window assignment for deteriorating jobs," *Journal of the Operational Research Society*, vol. 65, no. 2, pp. 291–301, 2014.

[10] J.-B. Wang and M.-Z. Wang, "Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 5, Article ID 1450036, 2014.

[11] G. Mosheiov and D. Oron, "Job-dependent due-window assignment based on a common flow allowance," *Foundations of Computing and Decision Sciences*, vol. 35, no. 3, pp. 185–195, 2010.

[12] J.-B. Wang, L. Liu, and C. Wang, "Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs," *Applied Mathematical Modelling*, vol. 37, no. 18-19, pp. 8394–8400, 2013.

[13] M. Ji, K. Chen, J. Ge, and T. C. E. Cheng, "Group scheduling and job-dependent due window assignment based on a common flow allowance," *Computers & Industrial Engineering*, vol. 68, pp. 35–41, 2014.

[14] Y. Yin, T. C. E. Cheng, C.-C. Wu, and S.-R. Cheng, "Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time," *Journal of the Operational Research Society*, vol. 65, no. 1, pp. 1–13, 2013.

[15] Y. Yin, D.-J. Wang, T. C. E. Cheng, and C.-C. Wu, "Bi-criterion single-machine scheduling and due-window assignment with common flow allowances and resource-dependent processing times," *Journal of the Operational Research Society*, vol. 67, no. 9, pp. 1169–1183, 2016.

[16] D. Wang, Y. Yin, and T. C. E. Cheng, "A bicriterion approach to common flow allowances due window assignment and scheduling with controllable processing times," *Naval Research Logistics (NRL)*, vol. 64, no. 1, pp. 41–63, 2017.

[17] B. Mor and G. Mosheiov, "Minsum and minmax scheduling on a proportionate flowshop with common flow-allowance," *European Journal of Operational Research*, vol. 254, no. 2, pp. 360–370, 2017.

[18] J.-B. Wang, B. Zhang, L. Li, D. Bai, and Y.-B. Feng, "Due-window assignment scheduling problems with position-dependent weights on a single machine," *Engineering Optimization*, vol. 52, no. 2, pp. 185–193, 2020.

[19] D.-L. Yang, C.-J. Lai, and S.-J. Yang, "Scheduling problems with multiple due windows assignment and controllable

processing times on a single machine," *International Journal of Production Economics*, vol. 150, pp. 96–103, 2014.

[20] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[21] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, Cambridge, UK, 1967.

[22] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[23] J.-B. Wang, J. Xu, and J. Yang, "Bi-criterion optimization for flow shop with a learning effect subject to release dates," *Complexity*, vol. 2018, Article ID 9149510, 12 pages, 2018.

[24] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.

[25] X.-R. Wang, J.-B. Wang, J. Jin, and P. Ji, "Single machine scheduling with truncated job-dependent learning effect," *Optimization Letters*, vol. 8, no. 2, pp. 669–677, 2014.

[26] A. Azzouz, M. Ennigrou, and L. B. Said, "Scheduling problems under learning effects: classification and cartography," *International Journal of Production Research*, vol. 56, no. 3, pp. 1–20, 2017.

[27] C.-L. Hsu and J.-R. Liao, "Two parallel-machine scheduling problems with function constraint," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 2717095, 6 pages, 2020.

[28] P. Liu and L. Duan, "A note on two-agent scheduling with resource dependent release times on a single machine," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 503297, 4 pages, 2015.

[29] P. Xue and Y. Zhang, "Single-machine scheduling with upper bounded maintenance time under the deteriorating effect," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 756251, 6 pages, 2013.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 2717095, 6 pages
https://doi.org/10.1155/2020/2717095

*Research Article*

# Two Parallel-Machine Scheduling Problems with Function Constraint

**Chia-Lun Hsu** [1,2] **and Jan-Ray Liao** [1]

[1] *Department of Electrical Engineering, National Chung-Hsing University, Tai-Chung 402, Taiwan*
[2] *Aeronautical Systems Research Division, Chung Shan Institute of Science and Technology, Tai-Chung 402, Taiwan*

Correspondence should be addressed to Chia-Lun Hsu; d9864002@mail.nchu.edu.tw

Received 5 April 2020; Accepted 29 April 2020; Published 18 May 2020

Guest Editor: Yunqiang Yin



The objective of this paper is to minimize both the makespan and the total completion time. Since parallel-machine scheduling which contains the function constraint problem has been a new issue, this paper explored two parallel-machine scheduling problems with function constraint, which refers to the situation that the two machines have a same function but one of the machines has another. We pointed out that the function constraint occurs not only in the manufacturing system but also in the service system. For the makespan problem, we demonstrated that it is NP-hard in the ordinary sense. In addition, we presented a polynomial time heuristic for this problem and have proved its worst-case ratio is not greater than 5/4. Furthermore, we simulated the performance of the algorithm through computational testing. The overall mean percent error of the heuristic is 0.0565%. The results revealed that the proposed algorithm is quite efficient. For the total completion time problem, we have proved that it can be solved in $O(n^4)$ time.

## 1. Introduction

The scheduling problem studied in this paper was motivated by the manufacturing of the metal processing industry. In the traditional manufacturing, a lathe machine and a milling machine have different functions. Generally speaking, the lathe machine is a tool that rotates a workpiece about an axis of rotation to perform various operations such as cutting, deformation, knurling sanding, and turning. The milling machine is used to cut the plane, while the shape of the forming surface, the spiral groove, or the tooth shape of the gear is milled with a special-shaped milling cutter. During milling, the workpiece is mounted on a table or an indexing head attachment, and the milling cutter performs a cutting motion, supplemented by a table for feeding motion. And now, a 5-axis machining center for milling and turning has all the functions of lathe machine and milling machine. As the company's performance grows rapidly, companies must purchase machines to meet customer needs. While newly purchased machines tend to have more function than older machines, functional alternatives occur between machines. We named this phenomenon as function constraint (e.g., lathe machine versus 5-axis machining centers for milling and turning). The function constraint occurs not only in the manufacturing system but also in the service system. In the service system, the parallel-machine is composed by the employees. The senior staff, the junior employees, and the new employees all have the ability to limit and replace.

Parallel machine in a production environment can be divided into three categories according to the nature of the machine: identical, uniform, and unrelated parallel machines. For decades, the parallel-machine production scheduling problem has been extensively studied under various classical scheduling performance measurement criteria. The makespan and total completion time are the two best important performance measurement criteria. The makespan is also called the maximum completion time or the completion time of the last workpiece on the last machine. The makespan is usually used to measure the utilization of machinery and equipment. If one shortens the

makespan and the machine utilization and production efficiency is improved, more flexible time can be reserved to prevent the sudden occurrence of the production line. The completion time refers to the time spent in the system from the time of the workpiece that arrives at the site to its completion, and the total completion time is the sum of the completion time of all the workpieces. Therefore, the total completion time is expected to be minimized. That is, the in-process inventory of the factory is expected to be minimized to reduce the cost of inventory.

Chung et al. [1] explored a two identical parallel-machine scheduling problem with molds constraints. Their objective is to minimize the makespan. For the problem is NP-hard, they give three heuristics and analyze each heuristic has a worst-case performance ratio of 3/2. Computational results revealed that heuristics are efficient even for the large-sized problem. Xu and Yang [2] studied a two parallel-machine scheduling with a periodic availability constraint. Their objective is to minimize the makespan. Because the problem is NP-hard, they present a mathematical programming model to solve it and, next, compare the performance of the longest processing time first (LPT) algorithm with the list scheduling (LS) via computational experiments. Most of the results showed that the LPT is better than the LS. Xu et al. [3] examined a two parallel machine scheduling problem to minimize the makespan. The problem is known as NP-hard. They applied the branch-and-bound method to solve the small-sized problem and presented a Tabu search algorithm for the large-sized problem. Ji and Cheng [4] presented a fully polynomial-time approximation scheme for parallel-machine scheduling under a grade of service provision to minimize makespan consideration. Lee et al. [5] addressed a makespan minimization scheduling problem on identical parallel-machine. They applied the simulated annealing method, and several heuristic algorithms have been proposed to tackle the problem. Computational results demonstrated that the simulated annealing method is efficient and better than the existing methods. Yin et al. [6] explored parallel-machine scheduling of deteriorating jobs with potential machine disruptions. The authors examined two cases of machine disruption (i.e., performing maintenance immediately on the disrupted machine when a disruption occurred and not performing machine maintenance). The nature of the jobs has two types: nonresumable and resumable. They determined the computational complexity status of various cases of the total completion time minimization problem and provided pseudopolynomial-time solution algorithms and fully polynomial-time approximation schemes for them.

Zhao et al. [7] explored a two parallel-machine scheduling problem where one machine is not available in a specified time period. The unavailable time period is fixed and known in advance. They proposed a fully polynomial-time approximation scheme for the total weighted completion time minimization problem and generalized the results to the case with $m$ parallel machine. Kuo and Yang [8] studied parallel-machine scheduling with time-dependent processing time. For the total completion time and the total load problems, they showed that the two problems are polynomially solvable. Gerstl and Mosheiov [9] proposed a general position-dependent processing time model. They considered scheduling problems which combine the option of job rejection and the model on parallel machine. Their objectives are total flow-time and total load. They proved that both problems can be solved in polynomial time in the number of jobs. Huang and Wang [10] stressed parallel-machine scheduling problems with deteriorating jobs. They showed that the total absolute differences in completion time and the total absolute differences in waiting time minimization problems can be solved in polynomial time.

Wang and Wang [11] studied a three-machine permutation flow shop scheduling problem with time-dependent processing times. The objective is to find a sequence that minimizes the makespan. Several dominance properties and a lower bound are derived to speed up the elimination process of a branch-and-bound algorithm. Moreover, two heuristic algorithms are proposed to overcome the inefficiency of the branch-and-bound algorithm. Computational results show that the heuristic algorithm M-NEH performs effectively and efficiently.

For more information, the reader may refer to the concise surveys on this topic by Cheng and Sin [12], Mokotoff [13], Pfund et al. [14], Kravchenko and Werner [15], Kaabi and Harrath [16], and Wang and Li [17].

Parallel-machine scheduling combines the function constraint problem is a new issue. Therefore, this paper explored two parallel-machine scheduling problems with function constraint. The objectives are to minimize the makespan and the total completion time.

The rest of the paper is organized as follows. In Section 2, we introduced the notation and formally formulated our problems. In Section 3, we demonstrated the computational complexity status and presented a heuristic algorithm, worst-case ratio, and computational experiments for the makespan minimization problem. In Section 4, we formulated the total completion time minimization problem as an assignment. In the last section, we concluded the paper and suggested issues for the future research.

## 2. Problem Formulation

There are $n$ jobs in the set $N = (J_1, J_2, \ldots, J_{n_1}, J_{n_1+1}, \ldots, J_n)$. Assume set $N$ consists of two classes of job that are the first class jobs and the second class jobs. Let $n_1$ and $n_2$ denote the number of jobs of the first class and the second class, respectively. Let $S_1$ and $S_2$ denote the set of the jobs of the first class and the second class, respectively. That is, $N = S_1 \cup S_2$, where $S_1 = (J_1, J_2, \ldots, J_{n_1})$ and $S_2 = (J_{n_1+1}, \ldots, J_n)$. Let $p_j$ denote the processing time of $J_j$ $(j = 1, 2, \ldots, n)$. Assume there are two parallel machines $M_i$ $(i = 1, 2)$, where $M_1$ can process the first class jobs only and $M_2$ can process both of the first class jobs and the second class jobs. Some common assumptions are as follows: (1) all jobs are non-preemptive and available for processing at time zero; (2) each machine can handle at most one job at a time and cannot stand idle until the last job assigned to it has finished processing; and

(3) each job can be processed on at most one machine at a time. The objectives are to minimize the makespan and the total completion time. We used "fc" to denote the function constraint relations between the two machines. Following the common scheduling notation, the problems can be described as $P2|fc|C_{\max}$ and $P2|fc|TC$, respectively.

## 3. Minimization of the Makespan

In this section, we confirmed the complexity and presented the worst-case bound for $P2|fc|C_{\max}$. First, we excluded the case of the sum of processing time in the second class jobs is larger than the sum of processing time in the first class jobs because the first class jobs processed on $M_1$ and the second class jobs processed on $M_2$ are optimal for the abovementioned case. Second, we analyze the complexity of the problem $P2|fc|C_{\max}$. Third, we presented a heuristic approach and analyze the worst-case bound of the algorithm. Finally, a computational experiment was conducted to evaluate the performance of the proposed algorithm.

*3.1. The Heuristic and Worst-Case Bound.* The special case of all the jobs belongs to the first class, $n_1 = n$, and then the problem $P2|fc|C_{\max}$ becomes the classical problem $P2\|C_{\max}$. It is known in advance that $P2\|C_{\max}$ is NP-hard in the ordinary sense [18]. Therefore, the following theorem holds.

**Theorem 1.** *Scheduling problem $P2|fc|C_{\max}$ is NP-hard in the ordinary sense.*

A simple heuristic approach is described in Algorithm 1.

A simple sort can be done in $O(n\log n)$ (e.g., heap sorting). Hence, the complexity of Step 1 is $O(n\log n)$. The complexity of Step 2 is $O(n)$. Overall, the complexity of the algorithm is $O(n\log n)$.

A straightforward result is given as follows.

**Lemma 1.** *For the problem $P2|fc|C_{\max}$, the following holds:*

*(1) $C^*_{\max} \geq \sum_{j=n_1+1}^{n} p_j$, where $C^*_{\max}$ denotes the makespan of the optimal schedule*

*(2) $C^*_{\max} \geq (\sum_{j=1}^{n_1} p_j + \sum_{j=n_1+1}^{n} p_j)/2 = LB$, where LB denotes the lower bound of the problem*

**Theorem 2.** *For the problem $P2|fc|C_{\max}$, $C_{\max}(LPT) \leq 5C^*_{\max}/4$.*

*Proof.* Assume $J_k$ is the last processed job based on the LPT approach. If $J_k \in S_2$, then $C_{\max}(LPT) = C^*_{\max}$. Therefore, the following proof processes are under the assumption of $J_k \in S_1$.

If $k = 1$, then $C_{\max}(LPT) = C^*_{\max}$.

For the case of $k \geq 2$, assume $s_k$ denotes the starting time of $J_k$. Since there is no idle time on the machines prior to the time $s_k$,

$$\begin{aligned} s_k &\leq \frac{\left(\sum_{j=1}^{k-1} p_j + \sum_{j=n_1+1}^{n} p_j\right)}{2} \\ &= \frac{\left(\sum_{j=1}^{k} p_j + \left(\sum_{j=n_1+1}^{n} p_j\right) - p_k\right)}{2} \\ &\leq \frac{\left(\sum_{j=1}^{n_1} p_j + \sum_{j=n_1+1}^{n} p_j - p_k\right)}{2} \\ &\leq C^*_{\max} - \frac{p_k}{2}. \end{aligned} \tag{1}$$

Consider the following two cases: $p_k \leq C^*_{\max}/2$ and $p_k > C^*_{\max}/2$.

When $p_k \leq C^*_{\max}/2$, we have

$$\begin{aligned} C_{\max}(LPT) &= s_k + p_k \\ &\leq C^*_{\max} - \frac{p_k}{2} + p_k \\ &= C^*_{\max} + \frac{p_k}{2} \\ &\leq C^*_{\max} + \frac{C^*_{\max}}{4} \\ &= 5\frac{C^*_{\max}}{4}. \end{aligned} \tag{2}$$

When $p_k > C^*_{\max}/2$, based on optimal consideration, we have $\sum_{j=n_1+1}^{n} p_j \leq C^*_{\max}/2$ and each machine cannot arrange two jobs that belong to $S_1$ and whose job's index is less or equal to $k$. Therefore, $J_k$ is the only job that belongs to $S_1$ and being processed on $M_2$. That is, $C_{\max}(LPT) = p_k + \sum_{j=n_1}^{n} p_j$. Within the optimal sequence, $J_k$ is the only job that belongs to $S_1$ and can be processed on $M_2$. We have $C^*_{\max} \geq C_{\max}(LPT)$. Hence, $C^*_{\max} = C_{\max}(LPT)$. □

*3.2. The Computational Experiments.* Although the worst case above seems quite good from the theoretical aspects, 25% errors cannot be accepted from the application dimensions. Therefore, some computational experiments were conducted to evaluate the performance of the LPT approach. The LPT approach was coded in Visual BASIC 6.0 and implemented on a personal computer with Intel core *i*7 16G CPU. Some test problems for each environment were randomly generated, the details of which are as follows:

(1) $n$ is equal to 50, 100, 150, 200, 250, 300, 500, and 1000

(2) $n_1$ is uniformly distributed over [1, $n$]. $n_2 = n - n_1$

(3) $p_i$ is uniformly distributed over [1, 50], [1, 100], [1, 200], [1, 500], and [1, 1000]

A ratio of $((C_{\max}(LPT) - LB) \times 100)/LB$ is used as an index to evaluate the performance of the LPT approach, where $C_{\max}(LPT)$ is the solution found by the LPT approach and $LB$

Step 1. Arrange the first class jobs in nonincreasing order, without loss of generality, and assume the LPT sequence remains $J_1, J_2, \ldots, J_{n_1}$.
Step 2. Assign $J_{n_1+1}, J_{n_1+2}, \ldots, J_n$ to the $M_2$. After that, whenever a machine is freed, the longest job among those not yet processed in the first class is put on the machine.

Algorithm 1: The longest processing time first (LPT) approach.

Table 1: Computational results of the LPT approach with five different processing time distribution periods.

| $n$ | $p(i)$ | | | | | Average |
|---|---|---|---|---|---|---|
| | $U$ (1, 50) | $U$ (1, 100) | $U$ (1, 200) | $U$ (1, 500) | $U$ (1, 1000) | |
| 50 | 0.0971 | 0.1884 | 0.2817 | 0.3714 | 0.4576 | 0.2792 |
| 100 | 0.0270 | 0.0492 | 0.0715 | 0.0939 | 0.1167 | 0.0716 |
| 150 | 0.0156 | 0.0264 | 0.0372 | 0.0483 | 0.0580 | 0.0371 |
| 200 | 0.0105 | 0.0174 | 0.0236 | 0.0295 | 0.0350 | 0.0232 |
| 250 | 0.0081 | 0.0132 | 0.0175 | 0.0211 | 0.0247 | 0.0169 |
| 300 | 0.0066 | 0.0107 | 0.0137 | 0.0163 | 0.0189 | 0.0132 |
| 500 | 0.0040 | 0.0059 | 0.0071 | 0.0080 | 0.0089 | 0.0068 |
| 1000 | 0.0022 | 0.0033 | 0.0038 | 0.0041 | 0.0043 | 0.0035 |
| Average | 0.0214 | 0.0397 | 0.0570 | 0.0741 | 0.0905 | 0.0565 |

is the solution found by lower bound. The ratio is computed for 300 test problems in each problem size; hence, 12000 ($300 \times 8 \times 5$) test problems are generated for the proposed problem $P2|fc|C_{\max}$. The computational results are shown in Table 1. The results revealed that the mean percentage errors of the LPT approach are 0.0214, 0.0397, 0.0560, 0.0741, and 0.0905 with five different processing time distribution periods, respectively. From Table 1, we found that the larger the problem the smaller the error percentages and the wider the processing time range the larger the error percentages. The overall mean percent error of the LPT approach is 0.0565%.

## 4. Minimization of the Total Completion Time

In this section, we showed that the problem $P2|fc|TC$ can be solved in $O(n^4)$ time. Let $(k_1, k_2)$ denotes the vector of the number of jobs allocated to $M_1$ and $M_2$, where $1 \le k_1 \le n_1$ and $n_2 \le k_2 \le n-1$. We excluded the case of $(k_1, k_2) = (0, n)$ because it becomes a single machine scheduling problem and the SPT (shortest processing time first) is optimal. Let $J_{[ij]}$ and $p_{[ij]}$ denote a job scheduled in the $j$th position on a machine $i$ and its processing time, respectively. For the convenience, we rename the job sequences processing on $M_1$ and $M_2$ as $J_{[11]}, J_{[12]}, \ldots, J_{[1k_1]}$ and $J_{[21]}, J_{[22]}, \ldots, J_{[2k_2]}$, respectively. Then, the total completion time can be calculated as follows:

$$TC = \sum C_j = \sum_{j=1}^{k_1} (k_1 - j + 1) p_{[1j]} + \sum_{j=1}^{k_2} (k_2 - j + 1) p_{[2j]}. \tag{3}$$

We rewrite equation (3) as

$$TC = \sum C_j = \sum_{j=1}^{n} w_j p_{[j]}, \tag{4}$$

where

$$w_j = \begin{cases} k_1 - j + 1, & 1 \le j \le k_1, \\ n - j + 1, & k_1 + 1 \le j \le n, \end{cases}$$
$$p_{[j]} = \begin{cases} p_{[1j]}, & 1 \le j \le k_1, \\ p_{[2j-k_1]}, & k_1 + 1 \le j \le n. \end{cases} \tag{5}$$

Assume the position of job that is processing on $M_1$ and $M_2$ is $1, \ldots, k_1$ and $k_1 + 1, k_1 + 2, k_1, k_2(n)$, respectively. Obviously, the second class of jobs cannot be scheduled at the previous position of $k_1$. Then, we gave the elements of assignment vector as follows:

$$a_{ij} = \begin{cases} w_j p_i, & 1 \le i \le n_1,\ 1 \le j \le n, \\ \infty, & n_1 + 1 \le i \le n,\ 1 \le j \le k_1, \\ w_j p_i, & n_1 + 1 \le i \le n,\ k_1 + 1 \le j \le n, \end{cases} \tag{6}$$

where

$$w_j = \begin{cases} k_1 - j + 1, & 1 \le j \le k_1 \\ n - j + 1, & k_1 + 1 \le j \le n. \end{cases} \tag{7}$$

Constraint of the second line at $a_{ij}$ ensures that the second class jobs are not scheduled before the position of $k_1$ under minimization consideration. For a given vector $(k_1, k_2)$, $P2|fc|TC$ can be formulated as the following assignment problem:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij}, \\ &\sum_{i=1}^{n} x_{ij} = 1, j = 1, \ldots, n \\ &\sum_{j=1}^{n} x_{ij} = 1, i = 1, \ldots, n \\ &x_{ij} = \begin{cases} 1, & \text{if job } i \text{ is scheduled in position } j, \\ 0. \end{cases} \end{aligned} \tag{8}$$

It is well-known that an assignment problem can be solved in $O(n^3)$ time. For a given vector of $(k_1, k_2)$, $1 \le k_1 \le n_1$, and $n_2 \le k_2 \le n-1$, problem $P2|fc|TC$ can be solved in $O(n^3)$ time. There are $n_1$ possible vectors of $(k_1, k_2)$, that is, $(1, n-1), \ldots, (n_1, n-n_1)$. Hence, problem $P2|fc|TC$ can be solved in $O(n_1 n^3)$ time, since $n_1 \le n$. This implies that the following theorem holds.

**Theorem 3.** *Problem $P2|fc|TC$ can be solved in $O(n^4)$ time.*

*Example 1.* Assume $N = S_1 \cup S_2$, $S_1 = \{J_1, J_2, J_3\}$, $S_2 = \{J_4, J_5, J_6, J_7, J_8, J_9, J_{10}\}$, $n = 10$, $n_1 = 3$, and $n_2 = 7$, and their processing time is listed in Table 2.

Table 2: The 10 jobs processing time.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 50 | 40 | 30 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

The following solution reports were obtained from Lingo 14.0 package. If $(k_1, k_2) = (1, 9)$, then $TC = 290$. If $(k_1, k_2) = (2, 8)$, then $TC = 262$. Therefore, the optimal solution is 262.

The optimal sequences those scheduling on $M_1$ and $M_2$ are $J_3 \longrightarrow J_2$ and $J_{10} \longrightarrow J_9 \longrightarrow J_8 \longrightarrow J_7 \longrightarrow J_6 \longrightarrow J_5 \longrightarrow J_4 \longrightarrow J_1$, respectively.

The Lingo programming code under $(k_1, k_2) = (2, 8)$ is listed as follows:

```
Model:
Sets:
 Jindex/1..10/;
 Assign (jindex,jindex):w, x;
Endsets
Data:
 w = 100  50  400  350  300  250  200  150  100  50
       80  40  320  280  240  200  160  120  80  40
       60  30  240  210  180  150  120  90  60  30
   35000   35000   56  49  42  35  28  21  14  7
   30000   30000   48  42  36  30  24  18  12  6
   25000   25000   40  35  30  25  20  15  10  5
   20000   20000   32  28  24  20  16  12  8  4
   15000   15000   24  21  18  15  12  9  6  3
   10000   10000   16  14  12  10  8  6  4  2
    5000    5000   8  7  6  5  4  3  2  1;
Enddata
Min = @sum (assign:w*x);
 @for (jindex(i):
    @sum (jindex (j): x (I,j)) = 1;
    @sum (jindex (j): x (j,i)) = 1;
 );
End
```

## 5. Conclusions

This paper explored two parallel-machine scheduling problems with function constraint. We pointed out that the function constraint occurs not only in the manufacturing system but also in the service system. Therefore, this work is meaningful.

In this research, two important and famous classical objectives, the makespan and the total completion time, were studied. For the makespan problem, we demonstrated that it is NP-hard in the ordinary sense. We presented a polynomial time heuristic for this problem and have proved its worst-case ratio is not greater than 5/4. Computational testing of the heuristic was conducted, and the results revealed that the overall mean percent error of the proposed algorithm is 0.0565%. For the total completion time problem, we have proved that it can be solved in $O(n^4)$ time.

Future research may examine other topics such as the total load, total tardiness, and number of tardy jobs or extending the model to the uniform parallel-machine setting or other settings.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] T. Chung, J. N. D. Gupta, H. Zhao, and F. Werner, "Minimizing the makespan on two identical parallel machines with mold constraints," *Computers & Operations Research*, vol. 105, pp. 141–155, 2019.

[2] D. Xu and D.-L. Yang, "Makespan minimization for two parallel machines scheduling with a periodic availability constraint: mathematical programming model, average-case analysis, and anomalies," *Applied Mathematical Modelling*, vol. 37, no. 14-15, pp. 7561–7567, 2013.

[3] J. Xu, S.-C. Liu, C. Zhao, J. Wu, W.-C. Lin, and P.-W. Yu, "An iterated local search and tabu search for two-parallel machine scheduling problem to minimize the maximum total completion time," *Journal of Information and Optimization Sciences*, vol. 40, no. 3, pp. 751–766, 2019.

[4] M. Ji and T. C. E. Cheng, "An FPTAS for parallel-machine scheduling under a grade of service provision to minimize makespan," *Information Processing Letters*, vol. 108, no. 4, pp. 171–174, 2008.

[5] W.-C. Lee, C.-C. Wu, and P. Chen, "A simulated annealing approach to makespan minimization on identical parallel machines," *The International Journal of Advanced Manufacturing Technology*, vol. 31, no. 3-4, pp. 328–334, 2006.

[6] Y. Yin, Y. Wang, T. C. E. Cheng, W. Liu, and J. Li, "Parallel-machine scheduling of deteriorating jobs with potential machine disruptions," *Omega*, vol. 69, pp. 17–28, 2017.
[7] C. Zhao, M. Ji, and H. Tang, "Parallel-machine scheduling with an availability constraint," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 778–781, 2011.
[8] W.-H. Kuo and D.-L. Yang, "Parallel-machine scheduling with time dependent processing times," *Theoretical Computer Science*, vol. 393, no. 1–3, pp. 204–210, 2008.
[9] E. Gerstl and G. Mosheiov, "Scheduling on parallel identical machines with job-rejection and position-dependent processing times," *Information Processing Letters*, vol. 112, no. 19, pp. 743–747, 2012.
[10] X. Huang and M.-Z. Wang, "Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1349–1353, 2011.
[11] J. B. Wang and M. Z. Wang, "Minimizing makespan in three-machine flow shops with deteriorating jobs," *Computers & Operations Research*, vol. 40, pp. 47–557, 2013.
[12] T. C. E. Cheng and C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research," *European Journal of Operational Research*, vol. 47, no. 3, pp. 271–292, 1990.
[13] E. Mokotoff, "Parallel machine scheduling problems: a survey," *Asia-Pacific Journal of Operational Research*, vol. 18, no. 2, pp. 193–242, 2001.
[14] M. Pfund, J. W. Fowler, and J. N. D. Gupta, "A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems," *Journal of the Chinese Institute of Industrial Engineers*, vol. 21, no. 3, pp. 230–241, 2004.
[15] S. A. Kravchenko and F. Werner, "Parallel machine problems with equal processing times: a survey," *Journal of Scheduling*, vol. 14, no. 5, pp. 435–444, 2011.
[16] J. Kaabi and Y. Harrath, "A survey of parallel machine scheduling under availability constraints," *International Journal of Computer and Information Technology*, vol. 3, pp. 238–245, 2014.
[17] J.-B. Wang and L. Li, "Machine scheduling with deteriorating jobs and modifying maintenance activities," *The Computer Journal*, vol. 61, no. 1, pp. 47–53, 2018.
[18] M. Pinedo, *Scheduling: Theory, Algorithms, and System*, Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 2008.