Hindawi
Shock and Vibration
Volume 2024, Article ID 9872819, 1 page
https://doi.org/10.1155/2024/9872819

Retraction

Retracted: Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Sun, B. Wu, and L. Ning, "Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation," *Shock and Vibration*, vol. 2021, Article ID 6687525, 10 pages, 2021.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6687525, 10 pages
https://doi.org/10.1155/2021/6687525

Research Article

Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation

Li Sun,[1] **Bin Wu**,[1] **and Lei Ning**[2]

[1]*School of Business, Shanghai Dianji University, Shanghai 201306, China*
[2]*College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China*

Correspondence should be addressed to Bin Wu; wub@sdju.edu.cn

Received 19 November 2020; Revised 11 December 2020; Accepted 19 December 2020; Published 18 January 2021

Academic Editor: Tangbin Xia

We investigate parallel-machine scheduling with past-sequence-dependent (p-s-d) delivery times, DeJong's learning effect, rate-modifying activity, and resource allocation. Each machine has a rate-modifying activity. We consider two versions of the problem to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation, respectively. The problems under our present model can be solved in polynomial time.

1. Introduction

In practice, a finite amount of resource usually is allocated to a job to control its actual processing, which is the so-called scheduling problem with controllable processing times. Researchers in this case have to make two decisions–job sequence and resource allocation simultaneously–which is different from common scheduling problems. These kinds of scheduling problems have attracted a great deal of attention in the last three decades since Vickson. Vickson [1] initiated this field. The resource allocation function usually has two forms including a linear function and a convex function. Liu and Feng [2] address two-machine flowshop scheduling problems in which the processing time of a job is a function of its position in the sequence and its resource allocation. Zhu et al. [3]investigate scheduling problems with a deteriorating and resource-dependent maintenance activity. They show that all the considered problems are polynomially solvable. Liu et al. [4] consider a parallel-machine scheduling problem to minimize the sum of resource consumption and outsourcing cost. Liu et al. [5] consider single-machine scheduling problems which determine the optimal job schedule, due-window location, and resource allocation simultaneously.

In industrial production, machine unavailability periods are very common which is first studied by Lee and Leon. [6]. Motivated by this phenomenon, scheduling with a rate-modifying activity becomes a popular topic in the last decade. Zhu et al. [7] addresses a single-machine scheduling problem with resource allocation and a rate-modifying activity simultaneously. Ji et al. [8] consider single-machine scheduling with a common due-window and a deteriorating rate-modifying activity. Polynomial-time solution algorithms are provided for the corresponding problems. Yang and Yang [9] investigate parallel-machine scheduling problems with multiple rate-modifying activities. Zhu et al. [3] study single-machine scheduling problems with a deteriorating and resource-dependent maintenance activity. Luo [10] addresses a single-machine scheduling problem with a deteriorating rate-modifying activity to minimize the number of tardy jobs. He proposed an optimal polynomial time algorithm. Yu [11] considers an optimal single-machine scheduling with linear deterioration rate and rate-modifying activities.

In modern industry, the manufacturing environment has a great impact on jobs′ processing times. Such an extra time for eliminating the adverse effects between the main processing and the delivery of a job is viewed as a past-sequence-dependent (p-s-d) delivery time. Koulamas and Kyparisis [12] first introduced p-s-d delivery time into scheduling problem. Liu et al. [13] considered the problem of single-machine scheduling with p-s-d delivery times, which was introduced in Koulamas and Kyparisis [12]. Liu [14] introduced identical parallel-machine scheduling with p-s-d delivery times and the learning effect. Shen and Wu [15] studied single-machine scheduling with p-s-d delivery times and general learning effects.

The workers can acquire experience and improve the production efficiency continuously, and this phenomenon–first discussed by Wright [16]–is called the learning effect in the literature [17]. Wu et al. [18] study some single-machine scheduling problems with elapsed-time-based and position-based learning and forgetting effects. More recent papers that consider scheduling with learning effect include Rostami et al. [19], Zhang et al. [20], Yin et al. [21], Zhang and Wang [22], Toksari and Arik [23], Jiang et al. [24], Cheng et al. [25], Pei et al. [26], Mustu and Eren [27], and Liu and Feng. [2]. The above scheduling model with the position-based learning effect suffers a drawback that job's actual processing time is close to zero when the job's position is sufficiently large in a schedule. Scheduling problem with DeJong's learning effect is proposed, which overcomes the shortcomings in Wright's learning model. Okoowski and Gawiejnowicz [28] consider a parallel-machine scheduling problem with DeJong's learning effect and makespan objective. Ji et al. [29] consider a learning model in scheduling based on DeJong's learning effect. Ji et al. [30] consider parallel-machine scheduling with deteriorating jobs and DeJong's learning effect. They show minimizing the total completion time is polynomially solvable and minimizing the makespan is NP-hard. Throughout the paper, we will consider parallel-machine scheduling problem with DeJong's learning effect.

Scheduling problems concerning multimachine production environments are encountered in many modern manufacturing processes. To the best of our knowledge, scheduling with p-s-d delivery times, DeJong's learning effect, rate-modifying activity, and resource allocation has not been studied in the literature. In this paper, we study two versions of such problems under linear and convex resource consumption and show the problems are polynomially solvable. The remaining part of this paper is organized as follows. In Section 2, we formulate the problem and present some notation and one lemma. We introduce two versions of the problem to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation in Section 3. In Section 4, we conclude the paper.

2. Problem Formulation

There are a set of n independent and non-pre-emptive jobs simultaneously available for processing and m identical parallel machines. Each machine can handle one job at a time. With the assumption that $m < n$ throughout the paper, since the problem is trivial, if $m \geq n$, let p_{ij} (p_{ij}^A) be the normal (actual) processing time of job J_{ij} and $p_{i[r]}$ ($p_{i[r]}^A$) be the normal (actual) processing time of job $J_{i[r]}$ if it is scheduled in the rth position on machine M_i in a sequence. In view of the study of DeJong's learning model for scheduling, we adopt it in our paper as follows: $p_{i[r]}^A = p_{i[r]}(M + (1-M)r^{a_{i[r]}})$, where $a_{i[r]}$ is a nonpositive learning index and $a_{i[r]} < 0$. It is easy to know that if $M = 0$, the model reduces to the classical learning model.

In this paper, we will consider the situation of repairing or upgrading the machine that one rate-modifying activity is allowed on each machine throughout the scheduling to improve the machines production efficiency which is denoted by RMA. A rate-modifying activity (RMA) can be applied to the machine so as to change (usually to decrease) the normal processing times of the jobs. The time p_{ij} of processing job J_{ij} changes after the RMA to $\lambda_{ij}p_{ij}$. The machine will revert to its initial condition, and the learning effect will start anew after the rate-modifying activity. Suppose n_i is the number of jobs located on machine M_i and k_i is the position of the rate-modifying activity on machine M_i. In this paper, we consider two resource consumption functions.

A linear resource consumption function:

$$p_{i[r]}^A = p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad (1)$$

before rate-modifying activity and

$$p_{i[r]}^A = \lambda_{i[r]}p_{i[r]}\left(M + (1-M)\left(r-k_i\right)^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad (2)$$

after rate-modifying activity, where $\lambda_{i[r]}$ is the modifying rate to job $J_{i[r]}$ with $0 < \lambda_{i[r]} \leq 1$, $u_{i[r]}$ is the amount of the resource allocated to job $J_{i[r]}$ with $0 \leq u_{i[r]} \leq \overline{u}_{i[r]} < ((\lambda_{i[r]}p_{i[r]})/(b_{i[r]}))$, and $b_{i[r]}$ is the positive compression rate of job $J_{i[r]}$.

A convex resource consumption function:

$$p_{i[r]}^A = \left(\frac{p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad (3)$$

before rate-modifying activity and

$$p_{i[r]}^A = \left(\frac{\lambda_{i[r]}p_{i[r]}\left(M + (1-M)\left(r-k_i\right)^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad (4)$$

after rate-modifying activity, where v is a positive constant. The rate-modifying activity duration is a linear function of its starting time which is represented by $f(t) = \beta + \sigma t$, where $\beta > 0$ is the basic rate-modifying activity time, $\sigma > 0$ is a rate-modifying activity factor, and t is the starting time of the rate-modifying activity operation. The starting time of the rate-modifying activity is not known in advance, and it can be scheduled immediately after completing the processing of any job.

As in [12], the processing of job $J_{i[r]}$ must be followed by the p-s-d delivery time $q_{i[r]}$, which can be calculated as

$$\begin{aligned} q_{i[1]} &= 0, \\ q_{i[r]} &= \gamma W_{i[r]} = \gamma \sum_{l=1}^{r-1} p_{i[l]}^A, \end{aligned} \quad (5)$$

before rate-modifying activity and

$$q_{i[r]} = \gamma W_{i[r]} = \gamma\left(\sum_{l=1}^{r-1} p_{i[l]}^{A} + f(t)\right), \quad (6)$$

after rate-modifying activity, where $\gamma \geq 0$ is a normalizing constant and $W_{i[r]}$ denotes the waiting time of job $J_{i[r]}$. As usual, the postprocessing operation of any job $J_{i[l]}$ modelled by its delivery time $q_{i[l]}$ is performed off-line. Hence, it is not affected by the availability of the machine, and it can be implemented immediately upon completion of the main operation, and we have

$$\begin{aligned} C_{i[1]} &= p_{i[1]}, \\ C_{i[j]} &= W_{i[j]} + p_{i[j]}^{A} + q_{i[j]} = (1+\gamma)W_{i[j]} + p_{i[j]}^{A}, \end{aligned} \quad (7)$$

where $C_{i[j]}$ denotes the completion time of job $J_{i[j]}$.

Let denote the p-s-d delivery time by q_{psd}. In addition, we denote TADC_i the total absolute deviation of job completion times and TADW_i the total absolute deviation of job waiting times on machine M_i, i.e., $\text{TADC}_i = \sum_{l=1}^{n_i}\sum_{k=l}^{n_i} |C_{i[l]} - C_{i[k]}|$ and $\text{TADW}_i = \sum_{l=1}^{n_i}\sum_{k=l}^{n_i} |W_{i[l]} - W_{i[k]}|$. Let TC_i indicates the job's total processing times on machine M_i and TW_i indicates the job's total waiting times on machine M_i, i.e., $\text{TC}_i = \sum_{r=l}^{n_i} C_{i[r]}$ and $\text{TW}_i = \sum_{r=l}^{n_i} W_{i[r]}$. We will try to find the optimal job sequence, the optimal RMA, and the optimal resource consumption such that the following cost functions are minimized:

$$\begin{aligned} Z_1 &= \alpha_1 \sum_{i=1}^{m} \text{TC}_i + \delta_1 \sum_{i=1}^{m} \text{TADC}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}u_{ij}, \\ Z_2 &= \alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}u_{ij}, \end{aligned} \quad (8)$$

where $\alpha_1, \alpha_2, \delta_1, \delta_2 > 0$ represent the per unit time contribution for the total processing time, the total waiting time, the total absolute deviation of job completion times, and the total absolute deviation of job waiting times on machine M_i with $\alpha_1 > 0$, $\alpha_2 > 0$, $\delta_1 > 0$, and $\delta_2 > 0$. G_{ij} is the per unit time cost associated with resource allocation. Let DJLR denote DeJong's learning effect and linear resource consumption and DJCR denote DeJong's learning effect and convex resource consumption. Using the three-field notation introduced by Graham et al., for scheduling problems, we denote the two versions of the problems as

$$\begin{aligned} &P_m \left| q_{\text{psd}}, \text{DJLR}, \text{RMA} \right| Z, \\ &P_m \left| q_{\text{psd}}, \text{DJCR}, RMA \right| Z, \\ &Z \in \{Z_1, Z_2\}. \end{aligned} \quad (9)$$

We first present some notation and one lemma before the main results. On machine M_i, if the number of jobs n_i and the position of the job preceding the rate-modifying activity k_i are known in advance, then the job's completion times and the job's waiting times on machine M_i are as follows:

$$\begin{aligned} W_{i[1]} &= 0, \\ C_{i[1]} &= p_{i[1]}^{A}, \\ &\ldots, \\ W_{i[k_i]} &= p_{i[1]}^{A} + \cdots + p_{i[k_i-1]}^{A}, \\ C_{i[k_i]} &= (1+\gamma)\left(p_{i[1]}^{A} + p_{i[2]}^{A} + \cdots + p_{i[k_i-1]}^{A}\right) + p_{i[k_i]}^{A}, \\ f(t) &= \beta + \sigma\left(p_{i[1]}^{A} + p_{i[2]}^{A} + \cdots + p_{i[k_i]}^{A}\right), \\ W_{i[k_i+1]} &= \beta + (1+\sigma)\left(p_{i[1]}^{A} + \cdots + p_{i[k_i]}^{A}\right), \\ C_{i[k_i+1]} &= (1+\gamma)\left(\beta + (1+\sigma)\left(p_{i[1]}^{A} + \cdots + p_{i[k_i]}^{A}\right)\right) \\ &\quad + p_{i[k_i+1]}^{A}, \\ &\ldots, \\ W_{i[n_i]} &= \beta + (1+\sigma)\left(p_{i[1]}^{A} + \cdots + p_{i[k_i]}^{A}\right) + p_{i[k_i+1]}^{A} \cdots \\ &\quad + p_{i[n_i-1]}^{A}, \\ C_{i[n_i]} &= (1+\gamma)\left(\beta + (1+\sigma)\left(p_{i[1]}^{A} + \cdots + p_{i[k_i]}^{A}\right)\right. \\ &\quad \left. + p_{i[k_i+1]}^{A} + \cdots + p_{i[n_i-1]}^{A}\right) + p_{i[n_i]}^{A}. \end{aligned} \quad (10)$$

For the linear case,

$$\begin{aligned} p_{i[r]}^{A} &= p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad \text{if } r \leq k_i, \\ p_{i[r]}^{A} &= \lambda_{i[r]}p_{i[r]}\left(M + (1-M)(r-k_i)^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad \text{if } r \geq k_i. \end{aligned} \quad (11)$$

For the convex case,

$$\begin{aligned} p_{i[r]}^{A} &= \left(\frac{p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad \text{if } r \leq k_i, \\ p_{i[r]}^{A} &= \left(\frac{\lambda_{i[r]}p_{i[r]}\left(M + (1-M)(r-k_i)^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad \text{if } r \geq k_i. \end{aligned} \quad (12)$$

Let $P(n,m,k) = (n_1, n_2, \ldots, n_m; k_1, k_2, \ldots, k_m)$ denote an allocation vector. We provide a lemma concerning an upper bound on the number of $P(n,m,k)$ vectors.

Lemma 1. *The number of $P(n,m,k)$ vectors is bounded from above by $((n+1)^{2m-1})/m!$.*

Proof. See the work of Ma et al. [31].

3. Cases with Linear Resource Consumption Function

3.1. The Problem $P_m|q_{psd}, DJLR, RMA|Z_1$. In this section, we introduce the problem to minimize the sum of total completion times and total absolute deviation of job completion times with resource consumption on all the machines. For machine M_i, from the above analysis, we calculate the total completion times and the total absolute deviation of job completion times on this machine as follows:

$$\begin{aligned}
\mathrm{TC}_i &= (n_i - k_i)(1+\gamma)\beta + \sum_{h=1}^{k_i} \left(1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)\right) p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} \left(1 + (n_i - h)(1+\gamma)\right) p_{i[h]}^A, \\
\mathrm{TADC}_i &= \sum_{h=k_i+1}^{n_i} (2h - 1 - n_i)(1+\gamma)\beta + \sum_{h=1}^{k_i} \left((2h - 1 - n_i) + \sum_{l=h+1}^{k_i} (2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i)(1+\gamma)(1+\sigma) \right) p_{i[h]}^A \\
&\quad + \sum_{h=k_i+1}^{n_i-1} \left((2h - 1 - n_i) + \sum_{l=h+1}^{n_i} (2l - 1 - n_i)(1+\gamma) \right) p_{i[h]}^A + (n_i - 1) p_{i[n_i]}^A.
\end{aligned} \tag{13}$$

Hence, the sum of total completion times and total absolute deviation of job completion times with resource consumption on all the machines is

$$\begin{aligned}
&\alpha_1 \sum_{i=1}^{m} \mathrm{TC}_i + \delta_1 \sum_{i=1}^{m} \mathrm{TADC}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij} \\
&= (1+\gamma)\beta \left(\sum_{i=1}^{m} \alpha_1 (n_i - k_i) + \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} \delta_1 (2h - 1 - n_i) \right) + \sum_{i=1}^{m} \sum_{h=1}^{k_i} (\alpha_1 \\
&\quad \cdot (1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)) \\
&\quad + \delta_1 \left((2h - 1 - n_i) + \sum_{l=h+1}^{k_i} (2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i)(1+\gamma)(1+\sigma) \right) \Bigg) p_{i[h]}^A \\
&\quad + \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} \left(\alpha_1 (1 + (1+\gamma)(n_i - h)) + \delta_1 \left((2h - 1 - n_i) + \sum_{l=h+1}^{n_i} (2l - 1 - n_i)(1+\gamma) \right) \right) p_{i[h]}^A \\
&\quad + \sum_{i=1}^{m} (\alpha_1 + (n_i - 1)\delta_1) p_{i[n_i]}^A + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}.
\end{aligned} \tag{14}$$

Let

$$A_1 = (1+\gamma)\beta\left(\sum_{i=1}^{m}\alpha_1(n_i-k_i)+\sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}\delta_1(2h-1-n_i)\right),$$

$$w_{i[h]} = \begin{cases} \alpha_1\left(1+(k_i-h)(1+\gamma)+(n_i-k_i)(1+\gamma)(1+\sigma)\right)+\delta_1\Big((2h-1-n_i) \\ \quad +\sum_{l=h+1}^{k_i}(2l-1-n_i)(1+\gamma)+\sum_{l=k_i+1}^{n_i}(2l-1-n_i)(1+\gamma)(1+\sigma)\Big), \\ \qquad\qquad i=1,2,\ldots,m, h=1,2,\ldots,k_i, \\ \alpha_1\left(1+(1+\gamma)(n_i-h)\right)+\delta_1\Big((2h-1-n_i) \\ \quad +\sum_{l=h+1}^{n_i}(2l-1-n_i)(1+\gamma)\Big), \\ \qquad\qquad i=1,2,\ldots,m, h=k_i+1,k_i+2,\ldots,n_i-1, \\ \alpha_1+(n_i-1)\delta_1 \\ \qquad\qquad i=1,2,\ldots,m, h=n_i. \end{cases} \quad (15)$$

Thus,

$$\begin{aligned} &\alpha_1\sum_{i=1}^{m}\mathrm{TC}_i+\delta_1\sum_{i=1}^{m}\mathrm{TADC}_i+\sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{ij}u_{ij} \\ &= A_1+\sum_{i=1}^{m}\sum_{h=1}^{k_i}w_{i[h]}p_{i[h]}\left(M+(1-M)h^{a_{i[h]}}\right) \\ &\quad+\sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}w_{i[h]}\lambda_{i[h]}p_{i[h]}\left(M+(1-M)(h-k_i)^{a_{i[h]}}\right) \\ &\quad+\sum_{i=1}^{m}\sum_{h=1}^{n_i}\left(G_{i[h]}-w_{i[h]}b_{i[h]}\right)u_{i[h]}. \end{aligned} \quad (16)$$

From the above equation, for any job sequence, the optimal resource allocation for a job depends on the sign of $G_{i[h]}-w_{i[h]}b_{i[h]}$. If $G_{i[h]}-w_{i[h]}b_{i[h]}$ is negative, the maximum feasible amount of the resource should be allocated to job $J_{i[h]}$, if $G_{i[h]}-w_{i[h]}b_{i[h]}$ is positive, no resource should be allocated to job $J_{i[h]}$, and if $G_{i[h]}-w_{i[h]}b_{i[h]}$ is equal to zero, any of value of resource consumption will not affect the total cost. Let $u^*_{i[h]}$ denote the optimal resource allocation for job $J_{i[h]}$, where

$$u_{i[h]} = \begin{cases} \overline{u}_{i[h]}, & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}<0, \\ u_0\in[0,\overline{u}_{i[h]}], & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}=0, \\ 0, & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}>0. \end{cases} \quad (17)$$

From (17), we can obtain the optimal resource allocation for any given optimal sequence.

Since A_1 is a constant, when n_i and k_i is given, we can express the problem as the following assignment problem:

$$\begin{aligned} &F_1 = A_1+\min\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=1}^{n_i}\theta_{ij[h]}x_{ijh}, \\ &(AP_1)\ s.t.\ \sum_{j=1}^{n}x_{ijh}=1,\quad i=1,2,\ldots,m, h=1,2,\ldots,n_i, \\ &\sum_{i=1}^{m}\sum_{h=1}^{n_i}x_{ijh}=1,\quad j=1,2,\ldots,n, \\ &x_{ijh}=0 \text{ or } 1, i=1,2,\ldots,m, h=1,2,\ldots,n_i, j=1,2,\ldots,n, \end{aligned} \quad (18)$$

where

$$\begin{aligned} \theta_{ij[h]} &= \begin{cases} w_{i[h]}p_{ij[h]}+(G_{ij}-w_{i[h]}b_{ij})\overline{u}_{ij}, & \text{if } G_{ij}-w_{i[h]}b_{ij}<0, \\ w_{i[h]}\overline{p}_{ij[h]}, & \text{if } G_{ij}-w_{i[h]}b_{ij}\geq 0, \end{cases} \\ p_{ij[h]} &= \begin{cases} p_{ij}\left(M+(1-M)h^{a_{ij}}\right), & i=1,2,\ldots,m, j=1,2,\ldots,n, h=1,2,\ldots,k_i, \\ \lambda_{ij}p_{ij}\left(M+(1-M)(h-k_i)^{a_{ij}}\right), & i=1,2,\ldots,m, j=1,2,\ldots,n, h=k_i+1,k_i+2,\ldots,n_i. \end{cases} \end{aligned} \quad (19)$$

Consequently, when $P(n,m,k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 1.

Since the $P(n,m,k)$ vector is given, we know that the problem can be solved in $O(n^3)$ time. Together with Lemma 1, it is easy to obtain the following theorem.

Step 1: jobs are scheduled by (AP_1)
Step 2: optimal job resource allocation is calculated by formula (17)

Algorithm 1: Algorithm to solve the problem of minimizing the sum of total completion times and total absolute deviation of job completion times with linear resource consumption.

Theorem 1. *The problem $P_m|q_{psd}, DJLR, RMA|Z_1$ can be solved in $O(n^{2m+2})$ time.*

3.2. The Problem $P_m|q_{psd}, DJLR, RMA|Z_2$. In this section, we study the problem to minimize the sum of total waiting times and total absolute deviation of job waiting times with resource consumption on all the machines. For machine M_i, we compute the total waiting times and the total absolute deviation of job waiting times on this machine as follows:

$$\mathrm{TW}_i = (n_i - k_i)\beta + \sum_{h=1}^{k_i} \left((n_i - k_i)(1+\sigma) + k_i - h\right) p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} (n_i - h) p_{i[h]}^A,$$

$$\mathrm{TADW}_i = (k_i - 1)(n_i - k_i)\beta + \sum_{h=1}^{k_i} \left((k_i - h)(n_i - k_i)\sigma + (h-1)(k_i - h + (n_i - k_i)(1+\sigma)) \right) p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} k_i (n_i - h) p_{i[h]}^A. \quad (20)$$

Hence, the sum of total waiting times and total absolute deviation of job waiting times with resource consumption on all the machines is

$$\begin{aligned}
&\alpha_2 \sum_{i=1}^{m} \mathrm{TW}_i + \delta_2 \sum_{i=1}^{m} \mathrm{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij} u_{ij} \\
&= \sum_{i=1}^{m} \left(\alpha_2 + (k_i - 1)\delta_2\right)(n_i - k_i)\beta \\
&+ \sum_{i=1}^{m}\sum_{h=1}^{k_i} \left(\alpha_2 \left((n_i - k_i)(1+\sigma) + k_i - h\right)\right. \\
&+ \delta_2 \left((k_i - h)(n_i - k_i)\sigma + (h-1)\right. \\
&\cdot \left.\left.\left(k_i - h + (n_i - k_i)(1+\sigma)\right)\right)\right) p_{i[h]}^A \\
&+ \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i} \left(\alpha_2 + k_i\delta_2\right)(n_i - h) p_{i[h]}^A + \sum_{i=1}^{m}\sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}.
\end{aligned} \quad (21)$$

Let

$$A_2 = \sum_{i=1}^{m} (\alpha_2 + (k_i - 1)\delta_2)(n_i - k_i)\beta,$$

$$\varphi_{i[h]} = \begin{cases} \alpha_2 ((n_i - k_i)(1+\sigma) + k_i - h) + \delta_2 ((k_i - h)(n_i - k_i)\sigma \\ \quad + (h-1)(k_i - h + (n_i - k_i)(1+\sigma))), \\ \qquad\qquad i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ (\alpha_2 + k_i\delta_2)(n_i - h), \\ \qquad\qquad i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \quad (22)$$

Thus,

$$\begin{aligned}
&\alpha_2 \sum_{i=1}^{m} \mathrm{TW}_i + \delta_2 \sum_{i=1}^{m} \mathrm{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij} u_{ij} \\
&= A_2 + \sum_{i=1}^{m}\sum_{h=1}^{k_i} \varphi_{i[h]} p_{i[h]} \left(M + (1-M) h^{a_{i[h]}}\right) \\
&+ \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i} \lambda_{i[h]} \varphi_{i[h]} p_{i[h]} \left(M + (1-M)(h - k_i)^{a_{i[h]}}\right) \\
&+ \sum_{i=1}^{m}\sum_{h=1}^{n_i} \left(G_{i[h]} - \varphi_{i[h]} b_{i[h]}\right) u_{i[h]}.
\end{aligned} \quad (23)$$

For any job sequence, the optimal resource allocation for a job depends on the sign of $G_{i[h]} - \varphi_{i[h]} b_{i[h]}$. Let $u_{i[h]}^*$ denote the optimal resource allocation for job $J_{i[h]}$, where

$$u_{i[h]} = \begin{cases} \overline{u}_{i[h]}, & \text{if } G_{i[h]} - \varphi_{i[h]} b_{i[h]} < 0, \\ u_0 \in [0, \overline{u}_{i[h]}], & \text{if } G_{i[h]} - \varphi_{i[h]} b_{i[h]} = 0, \\ 0, & \text{if } G_{i[h]} - \varphi_{i[h]} b_{i[h]} > 0. \end{cases} \quad (24)$$

From (24), we can get the optimal resource allocation for any given optimal sequence.

Accordingly, when n_i and k_i is given, we can indicate the problem as the following assignment problem:

Step 1: jobs are scheduled by (AP_2)
Step 2: optimal job resource allocation is calculated by formula (24)

ALGORITHM 2: Algorithm to solve the problem of minimizing the sum of total waiting times and total absolute deviation of job waiting times with linear resource consumption.

$$F_2 = A_2 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{n_i} \rho_{ij[h]} y_{ijh},$$
$$(AP_2)\ s.t. \sum_{j=1}^{n} y_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,$$
$$\sum_{i=1}^{m} \sum_{h=1}^{n_i} y_{ijh} = 1, \quad j = 1, 2, \ldots, n, y_{ijh} = 0$$
$$\text{or } 1,\ i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n, \tag{25}$$

where

$$\rho_{ij[h]} = \begin{cases} \varphi_{i[h]} p_{ij[h]} + \left(G_{ij} - \varphi_{i[h]} b_{ij}\right) \overline{u}_{ij}, & \text{if } G_{ij} - \varphi_{i[h]} b_{ij} < 0, \\ \varphi_{i[h]} p_{ij[h]}, & \text{if } G_{ij} - \varphi_{i[h]} b_{ij} \geq 0, \end{cases}$$
$$p_{ij[h]} = \begin{cases} p_{ij}\left(M + (1-M)h^{a_{ij}}\right), \quad i = 1, 2, \ldots, m, \\ \qquad j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\ \lambda_{ij} p_{ij}\left(M + (1-M)\left(h - k_i\right)^{a_{ij}}\right), \\ i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{26}$$

Hence, when $P(n, m, k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 2.

Thus, when the $P(n, m, k)$ vector is given, the problem can be solved in $O(n^3)$ time. Together with Lemma 1, we have the following theorem.

Theorem 2. *The problem $P_m|q_{psd}, DJLR, RMA|Z_2$ can be solved in $O(n^{2m+2})$ time.*

4. Cases with Convex Resource Consumption Function

In this section, we will consider the problems under convex resource consumption function, i.e.,

$$P_m\Big|q_{\text{psd}}, \text{DJCR}, \text{RMA}\Big|Z, \quad Z \in \{Z_1, Z_2\}. \tag{27}$$

Let

$$\overline{p}_{i[h]} = \begin{cases} p_{i[h]}\left(M + (1-M)h\right)^{a_{i[h]}}, & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ \lambda_{i[h]} p_{i[h]}\left(M + (1-M)\left(h - k_i\right)^{a_{i[h]}}\right), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{28}$$

Similar to the analysis of problem $P_m|q_{\text{psd}}, \text{DJLR}, \text{RMA}|Z_1$, if n_i and k_i is given, we calculate the problem to minimize the sum of total completion times and total absolute deviation of job completion times with convex resource consumption as follows:

$$\begin{aligned} H_1 &= \alpha_1 \sum_{i=1}^{m} \text{TC}_i + \delta_1 \sum_{i=1}^{m} \text{TADC}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij} \\ &= A_1 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} w_{i[h]} \left(\frac{\overline{p}_{i[h]}}{u_{i[h]}}\right)^{v} + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}, \end{aligned} \tag{29}$$

where

$$A_1 = (1+\gamma)\beta\left(\sum_{i=1}^{m} \alpha_1 (n_i - k_i) + \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} \delta_1 (2h - 1 - n_i)\right),$$
$$w_{i[h]} = \begin{cases} \alpha_1 \left(1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)\right) + \delta_1 \Big((2h - 1 - n_i) \\ \quad + \sum_{l=h+1}^{k_i} (2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i)(1+\gamma)(1+\sigma)\Big), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ \alpha_1 \left(1 + (1+\gamma)(n_i - h)\right) + \delta_1 \Big((2h - 1 - n_i) + \sum_{l=h+1}^{n_i} (2l - 1 - n_i)(1+\gamma)\Big), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i - 1, \\ \alpha_1 + (n_i - 1)\delta_1, & i = 1, 2, \ldots, m, h = n_i. \end{cases} \tag{30}$$

Step 1: jobs are scheduled by (AP_3)
Step 2: optimal job resource allocation is calculated by formula (31)

Algorithm 3: Algorithm to solve the problem of minimizing the sum of total completion times and total absolute deviation of job completion times with convex resource consumption.

By taking the first derivative of H_1 with respect to $u_{i[h]}$, $i = 1, 2, \ldots, m$ and $h = 1, 2, \ldots, n_i$, equating the result to zero, and solving it for $u_{i[h]}$, we can obtain the optimal resource allocation (denoted by $u^*_{i[h]}$):

$$\frac{\partial H_1}{\partial u_{i[h]}} = -v w_{i[h]} \left(\overline{p}_{i[h]}\right)^{v} u_{i[h]}^{-v-1} + G_{i[h]} = 0,$$
$$u^*_{i[h]} = \left(\frac{v w_{i[h]}}{G_{i[h]}}\right)^{1/(v+1)} \left(\overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{31}$$

By substituting $u^*_{i[h]}$ into the objective function H_1, we obtain a new unified expression as follows:

$$H_1 = A_1 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{i[h]} \overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{32}$$

Therefore, we can formulate the minimum problem as the following assignment problem:

$$H_1 = A_1 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{n_i} \xi_{ij[h]} \overline{x}_{ijh},$$
$$(AP_3)\ s.t. \sum_{j=1}^{n} \overline{x}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,$$
$$\sum_{i=1}^{m} \sum_{h=1}^{n_i} \overline{x}_{ijh} = 1, \quad j = 1, 2, \ldots, n, \overline{x}_{ijh} = 0 \text{ or } 1,$$
$$i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n, \tag{33}$$

where

$$\xi_{ij[h]} = \begin{cases} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{ij} p_{ij} \left(M + (1-M) h^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\ \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{ij} \lambda_{ij} p_{ij} \left(M + (1-M)(h-k_i)^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{34}$$

Consequently, when $P(n, m, k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 3.

Together with Lemma 1, we have the following theorem.

Theorem 3. *The problem $P_m|q_{psd}, DJCR, RMA|Z_1$ can be solved in $O(n^{2m+2})$ time.*

Similar to the analysis of problem $P_m|q_{\text{psd}}, \text{DJLR}, \text{RMA}|\alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij}$, if n_i and k_i is given, we calculate the problem to minimize the sum of total waiting times and total absolute deviation of job waiting times with convex resource consumption as follows:

$$H_2 = \alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij}$$
$$= A_2 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \varphi_{i[h]} \left(\frac{\overline{p}_{i[h]}}{u_{i[h]}}\right)^{v} + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}, \tag{35}$$

where

$$A_2 = \sum_{i=1}^{m} (\alpha_2 + (k_i - 1)\delta_2)(n_i - k_i)\beta,$$
$$\varphi_{i[h]} = \begin{cases} \alpha_2 \left((n_i - k_i)(1+\sigma) + k_i - h\right) + \delta_2 \left((k_i - h)(n_i - k_i)\sigma \right. \\ \left. +(h-1)(k_i - h + (n_i - k_i)(1+\sigma))\right), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ (\alpha_2 + k_i \delta_2)(n_i - h), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i, \end{cases}$$
$$\overline{p}_{i[h]} = \begin{cases} p_{i[h]} (M + (1-M) h)^{a_{i[h]}}, & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ \lambda_{i[h]} p_{i[h]} \left(M + (1-M)(h-k_i)^{a_{i[h]}}\right), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{36}$$

Step 1: jobs are scheduled by (AP_4)
Step 2: optimal job resource allocation is calculated by formula (37)

Algorithm 4: Algorithm to solve the problem of minimizing the sum of total waiting times and total absolute deviation of job waiting times with convex resource consumption.

Hence, taking the first derivative of H_2 with respect to $u_{i[h]}$, $i = 1, 2, \ldots, m$ and $h = 1, 2, \ldots, n_i$, equating the result to zero, and solving it for $u_{i[h]}$, we can obtain the optimal resource allocation (denoted by $u^*_{i[h]}$):

$$\frac{\partial H_2}{\partial u_{i[h]}} = -\nu\varphi_{i[h]}\left(\overline{p}_{i[h]}\right)^{\nu} u_{i[h]}^{-\nu-1} + G_{i[h]} = 0,$$
$$u^*_{i[h]} = \left(\frac{\nu\varphi_{i[h]}}{G_{i[h]}}\right)^{1/(\nu+1)} \left(\overline{p}_{i[h]}\right)^{\nu/(\nu+1)}. \tag{37}$$

By substituting $u^*_{i[h]}$ into the objective function H_2, we obtain a new unified expression as follows:

$$H_2 = A_2 + \sum_{i=1}^{m}\sum_{h=1}^{n_i}\left(\nu^{-\nu/(\nu+1)} + \nu^{1/(\nu+1)}\right)\varphi_{i[h]}^{1/(\nu+1)}\left(G_{i[h]}\overline{p}_{i[h]}\right)^{\nu/(\nu+1)}. \tag{38}$$

Therefore, we can formulate the minimum problem as the following assignment problem:

$$H_2 = A_2 + \min \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=1}^{n_i} \eta_{ij[h]}\overline{y}_{ijh},$$
$$(AP_4)\ s.t.\ \sum_{j=1}^{n}\overline{y}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,$$
$$\sum_{i=1}^{m}\sum_{h=1}^{n_i}\overline{y}_{ijh} = 1, \quad j = 1, 2, \ldots, n, \overline{y}_{ijh} = 0 \text{ or } 1,$$
$$i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n, \tag{39}$$

where

$$\eta_{ij[h]} = \begin{cases} \left(\nu^{-\nu/(\nu+1)} + \nu^{1/(\nu+1)}\right)\varphi_{i[h]}^{1/(\nu+1)}\left(G_{ij}p_{ij}\left(M + (1-M)h^{a_{ij}}\right)\right)^{\nu/(\nu+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\ \left(\nu^{-\nu/(\nu+1)} + \nu^{1/(\nu+1)}\right)\varphi_{i[h]}^{1/(\nu+1)}\left(G_{ij}\lambda_{ij}p_{ij}\left(M + (1-M)(h-k_i)^{a_{ij}}\right)\right)^{\nu/(\nu+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{40}$$

Therefore, when $P(n, m, k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 4.

From the above analysis and Lemma 1, we have the following theorem.

Theorem 4. *The problem $P_m|q_{psd}, DJCR, RMA|Z_2$ can be solved in $O(n^{2m+2})$ time.*

5. Conclusions

In this paper, two versions of parallel-machine scheduling problems to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation are considered, respectively. We present the problems in this research can be solved polynomially. Future research will be worth extending to multiple rate-modifying activity or other objective scheduling problems.

Data Availability

All data generated or analysed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. G. Vickson, "Two single machine sequencing problems involving controllable job processing times," *A I I E Transactions*, vol. 12, no. 3, pp. 258–262, 1980.

[2] Y. Liu and Z. Feng, "Two-machine no-wait flowshop scheduling with learning effect and convex resource-dependent processing times," *Computers & Industrial Engineering*, vol. 75, pp. 170–175, 2014.

[3] H. Zhu, M. Li, and Z. Zhou, "Machine scheduling with deteriorating and resource-dependent maintenance activity," *Computers & Industrial Engineering*, vol. 88, pp. 479–486, 2015.

[4] Z. Liu, W.-C. Lee, and J.-Y. Wang, "Resource consumption minimization with a constraint of maximum tardiness on

parallel machines," *Computers & Industrial Engineering*, vol. 97, pp. 191–201, 2016.

[5] L. Liu, J.-J. Wang, F. Liu, and M. Liu, "Single machine due window assignment and resource allocation scheduling problems with learning and general positional effects," *Journal of Manufacturing Systems*, vol. 43, pp. 1–14, 2017.

[6] C.-Y. Lee and V. J. Leon, "Machine scheduling with a rate-modifying activity," *European Journal of Operational Research*, vol. 128, no. 1, pp. 119–128, 2001.

[7] Z. Zhu, F. Chu, L. Sun, and M. Liu, "Single machine scheduling with resource allocation and learning effect considering the rate-modifying activity," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5371–5380, 2013.

[8] M. Ji, J. Ge, K. Chen, and T. C. E. Cheng, "Single-machine due-window assignment and scheduling with resource allocation, aging effect, and a deteriorating rate-modifying activity," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 952–961, 2013.

[9] D.-L. Yang and S.-J. Yang, "Unrelated parallel-machine scheduling problems with multiple rate-modifying activities," *Information Sciences*, vol. 235, pp. 280–286, 2013.

[10] W.-C. Luo, "On scheduling a deteriorating rate-modifying activity to minimize the number of tardy jobs," *Journal of Operation Research Society of China*, vol. 20, 2018.

[11] S. Yu, "An optimal single-machine scheduling with linear deterioration rate and rate-modifying activities," *Journal of Combinatorial Optimization*, vol. 30, no. 2, pp. 242–252, 2015.

[12] C. Koulamas and G. J. Kyparisis, "Single-machine scheduling problems with past-sequence-dependent delivery times," *International Journal of Production Economics*, vol. 126, no. 2, pp. 264–266, 2010.

[13] M. Liu, F. Zheng, C. Chu, and Y. Xu, "New results on single-machine scheduling with past-sequence-dependent delivery times," *Theoretical Computer Science*, vol. 438, pp. 55–61, 2012.

[14] M. Liu, "Parallel-machine scheduling with past-sequence-dependent delivery times and learning effect," *Applied Mathematical Modelling*, vol. 37, no. 23, pp. 9630–9633, 2013.

[15] L. Shen and Y.-B. Wu, "Single machine past-sequence-dependent delivery times scheduling with general position-dependent and time-dependent learning effects," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5444–5451, 2013.

[16] T. P. Wright, "Factors affecting the cost of airplanes," *Journal of the Aeronautical Sciences*, vol. 3, no. 4, pp. 122–128, 1936.

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

[18] C.-H. Wu, W.-C. Lee, P.-J. Lai, and J.-Y. Wang, "Some single-machine scheduling problems with elapsed-time-based and position-based learning and forgetting effects," *Discrete Optimization*, vol. 19, pp. 1–11, 2016.

[19] M. Rostami, A. E. Pilerood, and M. M. Mazdeh, "Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment," *Computers & Industrial Engineering*, vol. 85, pp. 206–215, 2015.

[20] X. Zhang, L. Liao, W. Zhang, T. C. E. Cheng, Y. Tan, and M. Ji, "Single-machine group scheduling with new models of position-dependent processing times," *Computers & Industrial Engineering*, vol. 117, pp. 1–5, 2018.

[21] N. Yin, L. Kang, and X.-Y. Wang, "Single-machine group scheduling with processing times dependent on position, starting time and allotted resource," *Applied Mathematical Modelling*, vol. 38, no. 19-20, pp. 4602–4613, 2014.

[22] X. G. Zhang and Y. Wang, "Single-machine scheduling CON/SLK due window assignment problems with sum-of-processed times based learning effect," *Applied Mathematics and Computation*, vol. 250, pp. 628–635, 2015.

[23] M. D. Toksari and O. A. Arık, "Single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times," *Journal of Manufacturing Systems*, vol. 45, pp. 159–179, 2017.

[24] Z. Jiang, F. Chen, and H. Kang, "Single-machine scheduling problems with actual time-dependent and job-dependent learning effect," *European Journal of Operational Research*, vol. 227, no. 1, pp. 76–80, 2013.

[25] M. Cheng, P. R. Tadikamalla, J. Shang, and B. Zhang, "Single machine scheduling problems with exponentially time-dependent learning effects," *Journal of Manufacturing Systems*, vol. 34, pp. 60–65, 2015.

[26] J. Pei, X. Liu, B. Liao, P. M. Pardalos, and M. Kong, "Single-machine scheduling with learning effect and resource-dependent processing times in the serial-batching production," *Applied Mathematical Modelling*, vol. 58, pp. 245–253, 2018.

[27] S. Mustu and T. Eren, "The single machine scheduling problem with sequence-dependent setup times and a learning effect on processing times," *Applied Soft Computing*, vol. 71, pp. 291–306, 2018.

[28] D. Okoowski and S. Gawiejnowicz, "Exact and heuristic algorithms for parallel-machine scheduling with DeJongs learning effect," *Computers and Industrial Engineering*, vol. 59, pp. 272–279, 2010.

[29] M. Ji, D. Yao, Q. Yang, and T. C. E. Cheng, "Machine scheduling with DeJong's learning effect," *Computers & Industrial Engineering*, vol. 80, pp. 195–200, 2015.

[30] M. Ji, X. Tang, X. Zhang, and T. C. E. Cheng, "Machine scheduling with deteriorating jobs and DeJong's learning effect," *Computers & Industrial Engineering*, vol. 91, pp. 42–47, 2016.

[31] W.-M. Ma, L. Sun, S. C. Liu, and T. H. Wu, "Parall-machine scheduling with delivery times and deteriorating maintenance," *Asia-Pacific Journal of Operational Research*, vol. 32, no. 4, Article ID 155029, 2015.