Research Article

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

Li Sun, Bin Wu, and Lei Ning

1School of Business, Shanghai Dianji University, Shanghai 201306, China
2College 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 parallel-machine scheduling problem 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 $p_{ij}^A$ be the normal (actual) processing time of job $J_{ij}$ and $p_{ij}^A$ be the normal (actual) processing time of job $J_{ij}$. If it is scheduled in the $r$th position on machine $M_r$, in a sequence. In view of the study of DeJong's learning model for scheduling, we adopt it in our paper as follows: $p_{ij}^A = p_{ij}(M + (1 - M)r^{\alpha(r)})$, where $\alpha(r)$ is a nonpositive learning index and $\alpha(r) < 0$. 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 $p_{ij}^A$ of processing job $J_{ij}$ changes after the RMA to $p_{ij}$. 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:

$$p_{ij}^A = p_{ij}(M + (1 - M)r^{\alpha(r)}) - b_i u_i,$$

before rate-modifying activity and

$$p_{ij}^A = \lambda_i p_{ij}(M + (1 - M)(r - k_i)^{\alpha(r)}) - b_i u_i,$$

after rate-modifying activity, where $\lambda_i$ is the modifying rate to job $J_{ij}$ with $0 < \lambda_i \leq 1$, $u_i$ is the amount of the resource allocated to job $J_{ij}$ with $0 \leq u_i \leq \pi_i < (\lambda_i p_{ij}/(b_i))$, and $b_i$ is the positive compression rate of job $J_{ij}$.

A convex resource consumption function:

$$p_{ij}^A = \left(\frac{p_{ij}(M + (1 - M)r^{\alpha(r)})}{u_i}\right)^v,$$

before rate-modifying activity and

$$p_{ij}^A = \left(\frac{\lambda_i p_{ij}(M + (1 - M)(r - k_i)^{\alpha(r)})}{u_i}\right)^v,$$

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 $f(t) = \beta + at$, where $\beta > 0$ is the basic rate-modifying activity time, $a > 0$ 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 $J_{ij}$ must be followed by the p-s-d delivery time $q_{ij}$, which can be calculated as

$$q_{ij}^A = 0,$$

$$q_{ij}^A = \gamma W_{ij}(r) = \gamma \sum_{i=1}^{r-1} P_{ij}^A,$$
before rate-modifying activity and
\[ q_{ij[r]} = \gamma W_{ij[r]} = \gamma \left( \sum_{l=1}^{c-1} p_{ij[l]}^A + f(t) \right), \]  
(6)
after rate-modifying activity, where \( \gamma \geq 0 \) is a normalizing constant and \( W_{ij[r]} \) denotes the waiting time of job \( J_{ij[r]} \).

With delivery-time and TADW \( Ji \) constant and after rate-modifying activity, where \( Mi \) is the main results. On machine \( M_i \), we have
\[ C_{i[1]} = p_{i[1]}^A, \]
\[ C_{i[j]} = W_{ij[j]} + p_{i[j]}^A + q_{ij[j]} = (1 + \gamma)W_{ij[j]} + p_{i[j]}^A, \]  
(7)
where \( C_{i[j]} \) denotes the completion time of job \( J_{ij[j]} \).

Let denote the p-s delivery time by \( \eta_{psd} \). In addition, 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., TADC \(_i\) = \( \sum_{j=1}^n \sum_{k=1}^n |C_{ij[k]} - C_{ij[j]}| \) and TADW \(_i\) = \( \sum_{j=1}^n \sum_{k=1}^n |W_{ij[k]} - W_{ij[j]}| \). 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., TC \(_i\) = \( \sum_{j=1}^n C_{ij[j]} \) and TW \(_i\) = \( \sum_{j=1}^n W_{ij[j]} \). 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:
\[ Z_1 = \alpha_1 \sum_{j=1}^m TC_1 + \delta_1 \sum_{j=1}^m TADC_1 + \sum_{j=1}^n G_{ij}u_{ij}, \]
\[ Z_2 = \alpha_2 \sum_{j=1}^m TW_1 + \delta_2 \sum_{j=1}^m TADW_1 + \sum_{j=1}^n G_{ij}u_{ij}, \]  
(8)
where \( \alpha_1, \alpha_2, \delta_1, \delta_2 > 0 \) represent the per unit time contribution for the total processing 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
\[ P_m|\eta_{psd}, DJLR, RMA|Z, \]
\[ P_m|\eta_{psd}, DJCR, RMA|Z, \]  
(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:
\[ W_{ij[1]} = 0, \]
\[ C_{ij[1]} = p_{i[1]}^A, \]
\[ \ldots, \]
\[ W_{ij[k]} = p_{i[1]}^A + \cdots + p_{i[k-1]}^A, \]
\[ C_{ij[k]} = (1 + \gamma)(p_{i[1]}^A + p_{i[2]}^A + \cdots + p_{i[k-1]}^A) + p_{i[k]}^A, \]
\[ f(t) = \beta + \sigma(p_{i[1]}^A + p_{i[2]}^A + \cdots + p_{i[k]}^A), \]
\[ W_{ij[k+1]} = \beta + (1 + \sigma)(p_{i[1]}^A + \cdots + p_{i[k]}^A), \]
\[ C_{ij[k+1]} = (1 + \gamma)(\beta + (1 + \sigma)(p_{i[1]}^A + \cdots + p_{i[k]}^A)) + p_{i[k+1]}^A, \]
\[ \ldots, \]
\[ W_{ij[n]} = \beta + (1 + \sigma)(p_{i[1]}^A + \cdots + p_{i[k]}^A) + p_{i[k+1]}^A \cdots + p_{i[n-1]}^A, \]
\[ C_{ij[n]} = (1 + \gamma)(\beta + (1 + \sigma)(p_{i[1]}^A + \cdots + p_{i[k]}^A) + p_{i[k+1]}^A \cdots + p_{i[n-1]}^A) + p_{i[n]}^A. \]  
(10)
For the linear case,
\[ p_{i[r]}^A = p_{i[r]}(M + (1 - M)r^{\alpha_{ij[r]}} - b_{i[r]}u_{i[r]}), \quad \text{if } r \leq k_i, \]
\[ p_{i[r]}^A = \lambda_{i[r]}p_{i[r]}(M + (1 - M)(r - k_i)^{\alpha_{ij[r]}} - b_{i[r]}u_{i[r]}), \quad \text{if } r \geq k_i. \]  
(11)
For the convex case,
\[ p_{i[r]}^A = \left( \frac{p_{i[r]}(M + (1 - M)r^{\alpha_{ij[r]}})}{u_{i[r]}} \right)^\gamma, \quad \text{if } r \leq k_i, \]
\[ p_{i[r]}^A = \left( \frac{\lambda_{i[r]}p_{i[r]}(M + (1 - M)(r - k_i)^{\alpha_{ij[r]}})}{u_{i[r]}} \right)^\gamma, \quad \text{if } r \geq k_i. \]  
(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 $P_{m|q_{pud}}, DJLR, RMA|Z_1$. 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:

$$TC_i = (n_i - k_i) (1 + \gamma) + \sum_{h=1}^{k_i} (1 + (k_i - h) (1 + \gamma) + (n_i - k_i) (1 + \gamma) (1 + \sigma)) p^A_{i[h]} + \sum_{h=k_i+1}^{n_i} (1 + (n_i - h) (1 + \gamma)) p^A_{i[h]},$$

$$TADC_i = \sum_{h=k_i+1}^{n_i} (2h - 1 - n_i) (1 + \gamma) + \sum_{h=1}^{k_i} (2h - 1 - n_i) (1 + \gamma) + \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) (1 + \gamma) (1 + \sigma) p^A_{i[h]}$$

$$+ \sum_{h=k_i+1}^{n_i} (2h - 1 - n_i) + \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) (1 + \gamma) p^A_{i[h]} + (n_i - 1) p^A_{i[n_i]}. \tag{13}$$

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

\[
\alpha_1 \sum_{i=1}^{m} TC_i + \delta_1 \sum_{i=1}^{m} TADC_i + \sum_{i=1}^{m} \sum_{j=1}^{\sum_{h=k_i+1}^{n_i}} G_{ij}\mu_{ij}
\]

\[
= (1 + \gamma) \beta \left( \sum_{i=1}^{m} \alpha_1 (n_i - k_i) + \sum_{i=1}^{m} \delta_1 (2h - 1 - n_i) \right) + \sum_{i=1}^{m} \sum_{h=1}^{k_i} \alpha_1 \left( 1 + (k_i - h) (1 + \gamma) + (n_i - k_i) (1 + \gamma) (1 + \sigma) \right)
\]

\[
+ \delta_1 \left( 2h - 1 - n_i \right) + \sum_{i=1}^{m} \delta_1 (2l - 1 - n_i) (1 + \gamma) + \sum_{i=1}^{m} \left( 2l - 1 - n_i \right) (1 + \gamma) (1 + \sigma) \right) p^A_{i[h]} \tag{14}
\]

\[
+ \sum_{i=1}^{m} \sum_{j=1}^{\sum_{h=k_i+1}^{n_i}} \alpha_1 (1 + (1 + \gamma) (n_i - h)) + \delta_1 \left( 2h - 1 - n_i \right) + \sum_{i=1}^{m} \left( 2l - 1 - n_i \right) (1 + \gamma) \right) p^A_{i[k]} \]

\[
+ \sum_{i=1}^{m} \sum_{j=1}^{\sum_{h=k_i+1}^{n_i}} \left( \alpha_1 + (n_i - 1) \delta_1 \right) p^A_{i[n_i]} + \sum_{i=1}^{m} \sum_{j=1}^{\sum_{h=k_i+1}^{n_i}} G_{ij}\mu_{ij}. \]

Let
Algorithm 1.

\[
A_1 = (1 + \gamma)\beta\left(\sum_{i=1}^{m} \alpha_i (n_i - k_i) + \sum_{i=1}^{m} \sum_{h=k+1}^{n_i} \delta_i (2h - 1 - n_i)\right),
\]

\[
u_{i[h]} = \begin{cases} 
\alpha_i (1 + (k_i - h)(1 + \gamma)(n_i - k_i)(1 + (1 + \sigma)) + \delta_i (2h - 1 - n_i), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
\alpha_i (1 + (1 + \gamma)(n_i - h)) + \delta_i (2h - 1 - n_i), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i - 1, \\
\alpha_i + (n_i - 1)\delta_i, & i = 1, 2, \ldots, m, h = n_i. 
\end{cases}
\]

Thus,

\[
\alpha_i \sum_{i=1}^{m} TC_i + \delta_i \sum_{i=1}^{m} TADC_i + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{ij} u_{ij} = A_1 + \sum_{i=1}^{m} \sum_{h=1}^{k_i} w_{i[h]} P_{i[h]} (M + (1 - M) h^{a_i[h]}) + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \sum_{h=k+1}^{n_i} (G_{ij} - w_{i[h]} b_{i[h]}) u_{ij},
\]

From the above equation, for any job sequence, the optimal resource allocation for a job depends on the sign of \( G_{ij} - w_{i[h]} b_{i[h]} \). If \( G_{ij} - w_{i[h]} b_{i[h]} \) is negative, the maximum feasible amount of the resource should be allocated to job \( j_i[h] \). If \( G_{ij} - w_{i[h]} b_{i[h]} \) is positive, no resource should be allocated to job \( j_i[h] \), and if \( G_{ij} - w_{i[h]} b_{i[h]} \) is equal to zero, any 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

\[
\theta_{ij[h]} = \begin{cases} 
 w_{i[h]} P_{ij[h]} + (G_{ij} - w_{i[h]} b_{ij}) \pi_{ij}, & \text{if } G_{ij} - w_{i[h]} b_{ij} < 0, \\
 w_{i[h]} \pi_{ij[h]}, & \text{if } G_{ij} - w_{i[h]} b_{ij} \geq 0,
\end{cases}
\]

\[
P_{ij[h]} = \begin{cases} 
 P_{ij} (M + (1 - M) h^{a_i}), & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\
 \lambda_{ij} P_{ij} (M + (1 - M) (h - k_i)^{a_j}), & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases}
\]

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 \( O(n^2) \) time. Together with Lemma 1, it is easy to obtain the following theorem.
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 $O(n^{m+2})$ 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 completion times with linear resource consumption.

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

$$TW_i = (n_i - k_i)\beta + \sum_{h=1}^{k_i} ((n_i - k_i) (1 + \sigma) + k_i - h)p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} (n_i - h)p_{i[h]}^A,$$

$$TADW_i = (k_i - 1) (n_i - k_i)\beta + \sum_{h=1}^{k_i} ((k_i - h) (n_i - k_i)\sigma + (h - 1) (k_i - h + (n_i - k_i) (1 + \sigma )) p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} k_i (n_i - h)p_{i[h]}^A.$$  

(20)

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

$$\alpha_2 \sum_{i=1}^{m} TW_i + \delta_2 \sum_{i=1}^{m} TADW_i + \sum_{i=1}^{m} \sum_{j=1}^{n} G_{ij} u_{ij}$$

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

$$+ \sum_{i=1}^{m} \sum_{h=1}^{k_i} (\alpha_2 ((n_i - k_i) (1 + \sigma) + k_i - h)$$

$$+ \delta_2 ((k_i - h) (n_i - k_i)\sigma + (h - 1) (k_i - h + (n_i - k_i) (1 + \sigma ))) p_{i[h]}^A$$

$$+ \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} (\alpha_2 + k_i\delta_2) (n_i - h)p_{i[h]}^A + \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} G_{i[h]} u_{i[h]}.$$  

(21)

Let

$$A_2 = \sum_{j=1}^{m} (\alpha_2 + (k_j - 1)\delta_2) (n_j - k_j)\beta,$$

$$\varphi_{[i]\{k\}} = \begin{cases} 
\pi_{i[k]}, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} < 0, \\
0, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} = 0, \\
\varphi_{i[k]} b_{i[k]}, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} > 0.
\end{cases}$$  

(22)

Thus,

$$\alpha_2 \sum_{i=1}^{m} TW_i + \delta_2 \sum_{i=1}^{m} TADW_i + \sum_{i=1}^{m} \sum_{j=1}^{n} G_{ij} u_{ij}$$

$$= A_2 + \sum_{i=1}^{m} \sum_{h=1}^{k_i} \varphi_{i[h]} p_{i[h]} (M + (1 - M) h^{a(h)})$$

$$+ \sum_{i=1}^{m} \sum_{h=k_i+1}^{n_i} \lambda_{i[h]} \varphi_{i[h]} p_{i[h]} (M + (1 - M) (h - k_i)^{a(h)})$$

$$+ \sum_{i=1}^{m} \sum_{j=1}^{n} (G_{i[h]} - \varphi_{i[h]} b_{i[h]}) u_{i[h]}.$$  

(23)

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

$$u_{i[k]} = \begin{cases} 
\pi_{i[k]}, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} < 0, \\
0, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} = 0, \\
\varphi_{i[k]} b_{i[k]}, & \text{if } G_{i[k]} - \varphi_{i[k]} b_{i[k]} > 0.
\end{cases}$$  

(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:
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.

\[ F_2 = A_3 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \rho_{ij[h]} |y_{ij[h]}| \]

(\(AP_2\)) s.t. \( \sum_{j=1}^{n} y_{ij[h]} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i \),
\( \sum_{i=1}^{m} \sum_{h=1}^{n} y_{ij[h]} = 1, \quad j = 1, 2, \ldots, n, y_{ij[h]} = 0 \)

or \( i, j = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n_j \).

(25)

where

\[ \rho_{ij[h]} = \begin{cases} \phi_{ij[h]} P_{ij[h]} + (G_{ij} - \phi_{ij[h]} b_{ij}) \pi_{ij}, & \text{if } G_{ij} - \phi_{ij[h]} b_{ij} < 0, \\ \phi_{ij[h]} P_{ij[h]}, & \text{if } G_{ij} - \phi_{ij[h]} b_{ij} \geq 0. \end{cases} \]

(26)

\[ \bar{P}_{i[h]} = \begin{cases} P_{i[h]} (M + (1 - M) h^\alpha), & i = 1, 2, \ldots, m, \\ \lambda_{ij} P_{ij} (M + (1 - M) (h - k_i)^\alpha), & i = 1, 2, \ldots, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \]

Similar to the analysis of problem \(P_m q, \text{DJLR}, \text{RMA}|Z_1\), 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:

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 \(O(n^2)\) time. Together with Lemma 1, we have the following theorem.

Theorem 2. The problem \(P_m q, \text{DJLR}, \text{RMA}|Z_2\) 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.,

\[ P_m q, \text{DJCR}, \text{RMA}|Z_2, \quad Z \in \{Z_1, Z_2\}. \]

Let

\[ H_1 = \alpha_1 \sum_{i=1}^{m} TC_i + \delta_1 \sum_{i=1}^{m} TADC_i + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{ij[h]} u_{ij} \]

(29)

\[ = A_1 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \nu_{ij[h]} \frac{H_1[i]}{H_1[h]} + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{ij[h]} u_{ij[h]}, \]

where

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

\[ w_{j[h]} = \begin{cases} a_0 (1 + (k_i - h) (1 + \gamma) (n_i - k_i) (1 + \gamma) (1 + \sigma)) + \delta_1 (2h - 1 - n_i), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ + \sum_{i=h+1}^{k_i} (2l - n_i) (1 + \gamma) + \sum_{i=h+1}^{n} (2l - l - n_i) (1 + \gamma) (1 + \sigma), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i, \end{cases} \]

\[ \alpha_1 (1 + (1 + \gamma) (n_i - h)) + \delta_1 (2h - 1 - n_i) + \sum_{i=h+1}^{n} (2l - l - n_i) (1 + \gamma), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i - 1, \\ + a_1 (n_i - 1) \delta_1, & i = 1, 2, \ldots, m, h = n_i. \]

(30)
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 $u^*_{i[h]}$):

$$\frac{\partial H_1}{\partial u_{i[h]}} = -v u_{i[h]} (\overline{P_i[h]})^{e_{(v+1)}} u_{i[h]}^{-1} + G_i[h] = 0,$$

$$u^*_{i[h]} = \left(\frac{v u_{i[h]}}{G_i[h]}\right)^{1/(v+1)} (\overline{P_i[h]})^{e_{(v+1)}}. \quad \text{(31)}$$

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

$$H_1 = A_1 + \sum_{i=1}^m \sum_{h=1}^{n_i} \left( v^{-e_{(v+1)}} + v^{e_{(v+1)}} \right) u_{i[h]}^{e_{(v+1)}} (G_i[h] \overline{P_i[h]})^{e_{(v+1)}}. \quad \text{(32)}$$

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

$$H_1 = A_1 + \min_{\sum_{j=1}^n \overline{x}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, \quad \sum_{j=1}^n \overline{x}_{ijh} = 1, \quad j = 1, 2, \ldots, n, \overline{x}_{ijh} = 0 \text{ or } 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n} \left( A_P \right. \left. \sum_{j=1}^n \overline{x}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n \right). \quad \text{(33)}$$

where

$$\xi_{ij[h]} = \left\{ \begin{array}{ll}
(v^{-e_{(v+1)}} + v^{e_{(v+1)}}) u_{i[h]}^{e_{(v+1)}} (G_i[h] (M + (1 - M)(n_i))^{e_{(v+1)}}), & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\
(v^{-e_{(v+1)}} + v^{e_{(v+1)}}) u_{i[h]}^{e_{(v+1)}} (G_i[h] k_i (M + (1 - M)(n_i))^{e_{(v+1)}}), & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{array} \right. \quad \text{(34)}$$

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 $P_m|d_{pdn}, DJCR, RMA|Z_1$ can be solved in $O(n^{2m+2})$ time.

Similar to the analysis of problem $P_m|d_{pdn}, DJLR, RMA|\alpha_2 \sum_{i=1}^m TW_i + \delta_2 \sum_{i=1}^m TADW_i + \sum_{i=1}^m \sum_{j=1}^{n_i} G_i[j_i]$, 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:

$$H_2 = A_2 + \sum_{i=1}^m \sum_{h=1}^{n_i} \overline{x}_{ijh} \left( P_i[h] \right)^{e_{(v+1)}} u_{i[h]}^{e_{(v+1)}} + \sum_{i=1}^m \sum_{j=1}^{n_i} G_i[j_i] \quad \text{(35)}$$

where

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

$$\phi_{i[h]} = \left\{ \begin{array}{ll}
\alpha_2 ((n_i - k_i) (1 + \sigma) + k_i h) + \delta_2 (k_i - h) (n_i - k_i) \sigma, & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
\alpha_2 + k_i \delta_2 (n_i - h), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i, \\
\overline{P_i[h]} = \left\{ \begin{array}{ll}
\delta_2 (n_i - k_i) (1 + \sigma), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
\lambda_{i[h]} P_i[h] (M + (1 - M)(n_i))^{e_{(v+1)}), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{array} \right. \right. \quad \text{(36)}$$
Step 1: jobs are scheduled by \((AP_1)\)
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 \(u_{ij}(h), \, i = 1, 2, \ldots, m\) and \(h = 1, 2, \ldots, n_i\), equating the result to zero, and solving it for \(u_{ij}(h), \) we can obtain the optimal resource allocation (denoted by \(u_{ij}^*(h)\)):

\[
\frac{\partial H_2}{\partial u_{ij}(h)} = -v\varphi_{ij}(h)\left(F_{ij}(h)\right)^{v-1}G_{ij}(h) = 0,
\]

\[
u_{ij}^*(h) = \left(\frac{\varphi_{ij}(h)}{G_{ij}(h)}\right)^{1/(v+1)}\left(F_{ij}(h)\right)^{1/(v+1)}.
\]

By substituting \(u_{ij}^*(h)\) into the objective function \(H_2\), we obtain a new unified expression as follows:

\[
H_2 = A_2 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \left(\gamma^{v/(v+1)} + \nu^{1/(v+1)}\right)\varphi_{ij}(h)\left(G_{ij}(h)\right)^{v/(v+1)}.
\]

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

\[
H_2 = A_2 + \min \sum_{i=1}^{m} \sum_{h=1}^{n_i} \eta_{ij}(h)\bar{y}_{ij}(h), \quad \text{s.t.} \quad \sum_{i=1}^{m} \sum_{h=1}^{n_i} \eta_{ij}(h)\bar{y}_{ij}(h) = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,
\]

\[
\sum_{i=1}^{m} \sum_{h=1}^{n_i} \bar{y}_{ij}(h) = 1, \quad j = 1, 2, \ldots, n, \quad \bar{y}_{ij}(h) = 0 \text{ or } 1,
\]

\[
i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, \quad j = 1, 2, \ldots, n, \quad h = 1, 2, \ldots, k_i.
\]

\[
\eta_{ij}(h) = \begin{cases} 
\left(\gamma^{v/(v+1)} + \nu^{1/(v+1)}\right)\varphi_{ij}(h)\left(G_{ij}(h)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\
\left(\gamma^{v/(v+1)} + \nu^{1/(v+1)}\right)\varphi_{ij}(h)\left(G_{ij}(h)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases}
\]

\[
\text{Data Availability}
\]

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. 170–174, 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. 3–11, 2016.
[19] M. Rostami, A. E. Pileeroood, 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.