Research Article

Results of Parallel-Machine Scheduling Model with Maintenance Activity considering Time-Dependent Deterioration, Delivery Times, and Resource Allocation

Li Sun,1 Xiao-hong Zhang,1 and Lei Ning2

1School of Business, Shanghai Dianji University, Shanghai 201306, China
2School of Business, Shaoxing University, Shaoxing 312000, China

Correspondence should be addressed to Xiao-hong Zhang; yantaizh@163.com

Received 1 September 2020; Revised 13 January 2021; Accepted 11 September 2021; Published 25 October 2021

Academic Editor: You-Jin Park

Copyright © 2021 Li Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates parallel-machine scheduling models with maintenance activity, delivery times, time-dependent deterioration, and resource allocation. We consider two forms of the problem: the first is to minimize the sum of total completion times, total machine loads, the total absolute deviation of job completion times, and the total resource allocation; the second is to minimize the sum of total waiting times, total machine loads, the total absolute deviation of job waiting times, and the total resource allocation. The problems are proved to be solvable in polynomial time.

1. Introduction

Initiated by Vickson [1], scheduling problems with controllable processing times through resource allocation have been studied extensively by researchers since 1980. Wang et al. [2] introduce resource allocation scheduling with learning effect. They analyse the problem with two different processing time functions $p_j = \overline{p}_j r^a - a_j u_j$ and $p_j = (\overline{p}_j r^a/u_j)^k$, $u_j > 0$. They provide a polynomial time algorithm to the problem. Zhu et al. [3] investigate scheduling problems and show that all the presented problems are polynomially solvable. Liu and Feng [4] show two-machine no-wait flow shop scheduling problems. Liu et al. [5] consider a parallel-machine scheduling problem to minimize the sum of resource consumption and outsourcing cost given. Liu et al. [6] study scheduling problems on single machine to determine the optimal job schedule, the optimal due window location, and the optimal resource allocation.

On the other hand, deteriorating scheduling problems have attracting widespread attention since Gupta and Gupta [7] and Browne and Yechiali [8]. More recent papers considering deterioration scheduling problem include Huang and Wang [9], Wang et al. [10], Cheng et al. [11], Cheng et al. [12], Lee et al. [13], Wang and Wang [14], Cheng [15], Wu et al. [16], Yin et al. [17], Zhao and Tang [18], Lai and Lee [19], Miao et al. [20], Yin et al. [21], and Yin et al. [22].

In scheduling, Lee and Leon [23] introduce a new class of scheduling situations with rate-modifying activity in which the machines need to be maintained and become unavailable. Recently, scheduling problem with a rate-modifying activity becomes a research hotspot. Yin et al. [24] consider a single-machine batch delivery scheduling and common due date assignment problem. They provide some properties of the optimal schedule for the problem. Wang and Wei [25] introduce identical parallel-machine scheduling problems and show that the problems remain polynomially solvable under the proposed model. Wang and Wang [26] address SLK due date assignment scheduling problem with a rate-modifying activity. They present a polynomial time solution for this problem. Zhu et al. [27] address a single machine scheduling problem and show that the problems are solvable in $o(n^3)$ time for a linear resource allocation function and are solvable in $o(n^3 \log n)$ time for a convex resource allocation function. Ji et al. [28] consider single-machine common due-window and deteriorating rate-modifying
activity scheduling problem. They provide polynomial solution algorithms for the corresponding problems. Yang and Yang [29] introduce unrelated parallel-machine scheduling problems with multiple rate-modifying activities. Hsu et al. [30] extend an unrelated parallel-machine scheduling environment and present a more efficient algorithm to solve the extended problem.

Jobs’ processing time is impacted greatly by the environment in modern manufacture. There is some extra time to be needed to deal with this thing. This treatment can be performed after the component has been processed by the machine but before it is delivered to the customer so it can be delivered with a guarantee of safety. The extra time to eliminate the adverse effects between the main processing and the delivery of a job is called as a past-sequence-dependent delivery time. Delivery time is firstly introduced into scheduling problem by Koulamas and Kyparisis. Liu et al. [31] consider the single-machine delivery time scheduling problem, which was introduced in Koulamas and Kyparisis [32]. Liu [33] introduce parallel-machine learning scheduling problem, which was introduced in Koulamas and Kyparisis. Liu and the delivery of a job is called as a past-sequence-dependent delivery time. WT_his treatment can be eliminated the adverse effects between the main processing and the delivery of a job. Jobs’ processing completion and its starting time is not affected by the deteriorating maintenance activity. WT_he deteriorating maintenance activity duration is represented by \( f(t) = \beta + \sigma t \), which is longer if it is started later, where \( \beta > 0 \) is the basic maintenance activity time, \( \sigma > 0 \) is the deterioration rate of maintenance activity, and \( t \) is the starting time of the deteriorating maintenance activity. The deteriorating maintenance activity can be scheduled immediately after any job’s processing completion and its starting time is not known in advance. The position of deteriorating maintenance activity on machine \( M_i \) is denoted by \( k_i \), if the deteriorating maintenance activity is scheduled immediately after completing the processing of the \( k \)th job \( J_{ik} \). The machine will revert to its initial condition, and the aging effect will start anew after the maintenance activity.

Let \( n_i \) denote the number of jobs located on machine \( M_i \) and \( k_i \) denote the position of deteriorating maintenance activity on machine \( M_i \). Hence, the deteriorating maintenance activity duration is \( f(t) = \beta + \sigma (\sum_{t=1}^{k_i} p_{i[t]} \). If a job is scheduled on the \( j \)th position of machine \( M_i \) in a sequence, its actual processing time with resource consumption is as follows.

A linear resource consumption function is given by

\[
\begin{align*}
    p_{i[j]} &= a_{i[j]} + bt - \lambda_{i[j]} u_{i[j]} = a_{i[j]} + b \sum_{l=1}^{j-1} p_{i[l]} - \lambda_{i[j]} u_{i[j]}, \quad \text{if } j \leq k_i, \\
    p_{i[j]} &= a_{i[j]} + bt - \lambda_{i[j]} u_{i[j]} = a_{i[j]} + b \sum_{l=k_i}^{j-1} p_{i[l]} - \lambda_{i[j]} u_{i[j]}, \quad \text{if } j > k_i,
\end{align*}
\]  

(1)

where \( \lambda_{i[j]} \) is the weight of the resource which is allocated to job \( J_{i[j]} \). \( u_{i[j]} \) is the amount of the resource allocated to job \( J_{i[j]} \), with \( 0 \leq u_{i[j]} \leq \pi_{i[j]} \leq (a_{i[j]} + b(j - 1)\pi_i/\lambda_{i[j]}), \) if \( j \leq k_i; \)

2. Problem Formulation

There are \( n \) independent and nonpreemptive jobs, which are simultaneously available for processing on \( m \) identical parallel machines. We suppose that \( m < n \) throughout the paper. Let \( a_{ij} \) be the normal processing time of job \( J_{ij} \) and \( p_{ij} \) be the actual processing time of job \( J_{ij} \). Let \( a_{i[k]} \) and \( p_{i[k]} \) be the normal processing time and the actual processing time of the \( k \)th job \( J_{i[k]} \) on machine \( M_i \). \( C_{i[j]} \) and \( W_{i[j]} \) are the completion time and the waiting time of the \( j \)th job on machine \( M_i \) in a sequence.

One deteriorating maintenance activity (denoted by DMA) is allowed on each machine throughout the scheduling to improve machine production efficiency. The deteriorating maintenance duration is represented by \( f(t) = \beta + \sigma t \), which is longer if it is started later, where \( \beta > 0 \) is the basic maintenance activity time, \( \sigma > 0 \) is the deterioration rate of maintenance activity, and \( t \) is the starting time of the deteriorating maintenance activity. The deteriorating maintenance activity can be scheduled immediately after any job’s processing completion and its starting time is not known in advance. The position of deteriorating maintenance activity on machine \( M_i \) is denoted by \( k_i \), if the deteriorating maintenance activity is scheduled immediately after completing the processing of the \( k \)th job \( J_{i[k]} \). The machine will revert to its initial condition, and the aging effect will start anew after the maintenance activity.

Let \( n_i \) denote the number of jobs located on machine \( M_i \) and \( k_i \) denote the position of deteriorating maintenance activity on machine \( M_i \). Hence, the deteriorating maintenance activity duration is \( f(t) = \beta + \sigma (\sum_{t=1}^{k_i} p_{i[t]} \). If a job is scheduled on the \( j \)th position of machine \( M_i \) in a sequence, its actual processing time with resource consumption is as follows.

A linear resource consumption function is given by
where \( v \) is a positive constant, \( u_{ij} \) is the amount of the resource allocated to job \( j \) with \( 0 \leq u_{ij} \leq \bar{u}_{ij} \), and \( \bar{u}_{ij} \) is the maximum resource allocation of job \( j \).

As in the study of Koulamas and Kyparisis [32], the processing of job \( j \) must be followed by the p-s-d delivery time \( q_{ij} \) which can be calculated as \( q_{ij} = 0 \) and \( q_{ij} = W_{ij} + f(t) \), before maintenance activity, and \( q_{ij} = W_{ij} + f(t) \), after maintenance activity, where \( \gamma \geq 0 \) is a normalizing constant.

As usual, we assume that the postprocessing operation of any job \( j \) modelled by its delivery time \( q_{ij} \) is performed off-line. Therefore, it is not affected by the availability of the machine and it can be implemented immediately upon completion of the operation resulting in

\[
C_{i[1]} = p_{[1]} + C_{[1]} = W_{ij} + p_{ij} + q_{ij} = (1 + \gamma)W_{ij} + p_{ij}, \quad j = 2, 3, \ldots, n_i.
\]

Let denote the p-s-d delivery time by \( q_{ps-d} \). In addition, we denote TADC the total absolute deviation of job completion times and TADW the total absolute deviation of job waiting times, i.e.,

\[
\text{TADC} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} |C_{i[j]} - C_{i[i]}|, \quad \text{TADW} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} |W_{ij} - W_{ij[1]}|.
\]

Let TC indicate the job’s total processing times and TW indicate the job’s total waiting times, i.e.,

\[
\text{TC} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_{i[j]}, \quad \text{TW} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} W_{ij}.
\]

We denote the load of machine \( M_i \) by \( L_i \), and the total machine load (TML) is \( \sum_{i=1}^{m} L_i \), i.e.,

\[
\text{TML} = \sum_{i=1}^{m} L_i.
\]

As in the study of Liu and Feng [4], we will try to find the optimal job sequence, the optimal DMA, and the optimal resource consumption on parallel-machine schedule such that the following cost functions are minimized:

\[
\begin{align*}
\delta_1 \text{TML} + \delta_2 \text{TC} + \delta_3 \text{TADC} + \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij}, \\
\delta_1 \text{TML} + \delta_2 \text{TW} + \delta_3 \text{TADW} + \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij},
\end{align*}
\]

where \( \delta_1, \delta_2, \delta_3, \delta_4 \) represent the per unit time contribution for the total machine load, the total processing (waiting) time, and the total absolute deviation of job completion (waiting) times with \( \delta_1 > 0, \delta_2 > 0, \delta_3 > 0, \delta_4 > 0 \). \( G_{ij} \) is the per unit time cost associated with resource allocation. Let LRA denote linear resource allocation and CRA denote convex resource allocation. Using the three-field notation introduced by Graham et al., for scheduling problems, we denote the two versions of the problems as

\[
P_{m|q_{ps-d}, LRA, DMA|z},
\]

\[
P_{m|q_{ps-d}, CRA, DMA|z},
\]

\[
z \in \left\{ \text{\delta}_1 \text{TML} + \text{\delta}_2 \text{TC} + \text{\delta}_3 \text{TADC} + \text{\delta}_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij}, \text{\delta}_1 \text{TML} + \text{\delta}_2 \text{TW} + \text{\delta}_3 \text{TADW} + \text{\delta}_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} \right\}.
\]

We first present some notations and one lemma before the main results. If the number of jobs on machine \( M_i \) is \( n_i \) and the position of the job preceding the deterioration activity is \( k_j \), the jobs’ completion times and waiting times on machine \( M_i \) are as follows.

For the linear case,

\[
C_{i[k_i]} = \left\{ \begin{align*}
\sum_{i=1}^{h-1} (1 + \gamma + b)(1 + b)^{k-1}(a_{i[i]} - \lambda_{i[i]}u_{i[i]}), \\
(1 + \gamma)b + (1 + \gamma)(1 + \sigma) \sum_{i=1}^{k} (1 + b)^{k-1}(a_{i[i]} - \lambda_{i[i]}u_{i[i]}), \\
+ \sum_{i=k+1}^{n_i} (1 + \gamma + b)(1 + b)^{n_i-1}(a_{i[i]} - \lambda_{i[i]}u_{i[i]}), \\
\end{align*} \right\} \text{ for } i = 1, 2, \ldots, m, \ h = 1, 2, \ldots, k_j,
\]

where \( \lambda_{i[i]} = \sum_{j=1}^{i} u_{ij} \) for all \( i \).
\begin{equation}
\begin{aligned}
W_{i[h]} &= \begin{cases}
(a_{i[h-1]} - \lambda_{i[h-1]} u_{i[h-1]}) + \sum_{l=1}^{h-2} (1 + b)^{h-l-1} \left( a_{i[l]} - \lambda_{i[l]} u_{i[l]} \right), \\
&\text{if } i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
\beta + (1 + \sigma) \sum_{l=1}^{k_i} (1 + b)^{h-l} \left( a_{i[l]} - \lambda_{i[l]} u_{i[l]} \right) + \sum_{l=k_i+1}^{h-2} (1 + b)^{h-l-1} \left( a_{i[l]} - \lambda_{i[l]} u_{i[l]} \right) + (a_{i[h-1]} - \lambda_{i[h-1]} u_{i[h-1]}), \\
&\text{if } i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases}
\end{aligned}
\end{equation}

For the convex case,

\begin{equation}
\begin{aligned}
C_{i[h]} &= \begin{cases}
\left( \frac{a_{i[h]}}{u_{i[h]}} \right)^\gamma + \sum_{l=1}^{h-1} (1 + \gamma + b) (1 + b)^{h-l-1} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma, \\
&\text{if } i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
(1 + \gamma) \beta + (1 + \gamma) (1 + \sigma) \sum_{l=1}^{k_i} (1 + b)^{h-l} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma + \sum_{l=k_i+1}^{n_i-1} (1 + \gamma + b) (1 + b)^{n_i-l-1} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma + \left( \frac{a_{i[n_i]}}{u_{i[n_i]}} \right)^\gamma, \\
&\text{if } i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases}
\end{aligned}
\end{equation}

\begin{equation}
\begin{aligned}
W_{i[h]} &= \begin{cases}
\left( \frac{a_{i[h-1]}}{u_{i[h-1]}} \right)^\gamma + \sum_{l=1}^{h-2} (1 + b)^{h-l-1} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma, \\
&\text{if } i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\
\beta + (1 + \sigma) \sum_{l=1}^{k_i} (1 + b)^{h-l} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma + \sum_{l=k_i+1}^{h-2} (1 + b)^{h-l-1} \left( \frac{a_{i[l]}}{u_{i[l]}} \right)^\gamma + \left( \frac{a_{i[h-1]}}{u_{i[h-1]}} \right)^\gamma, \\
&\text{if } i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases}
\end{aligned}
\end{equation}

Let $P(n, m, k) = (n_1, n_2, \ldots, n_m; k_1, k_2, \ldots, k_m)$ denote 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 Ma et al. [36]. \hfill \square

### 3. Cases with Linear Resource Consumption Function

3.1. The Problem $P_m | q_{ps}, d_i, LRA, DMA | \delta_i TML + \delta_i TC + \delta_i TADC + \delta_i \sum_{i=1}^{m} \sum_{j=1}^{n_i} \overline{G_i} u_{i,j}$. In this section, we introduce the problem to minimize the sum of total machine load, total completion times, times with resource consumption on all the machines. From the above analysis, for machine $M_i$, we
calculate the machine load, the total completion times, and the total absolute deviation of job completion times as follows:

\[
L_i = (1 + \gamma)\beta + (1 + \gamma)(1 + \sigma) \sum_{l=1}^{k_i} (1 + b)^{k_i-1}(a_{i[l]} - \lambda_{i[l]}u_{i[l]} + \\
+ \sum_{l=k_i+1}^{n-1} (1 + \gamma + b)(1 + b)^{n-1-l}(a_{i[l]} - \lambda_{i[l]}u_{i[l]}) + (a_{i[n]} - \lambda_{i[n]}u_{i[n]}),
\]

\[
TC_i = \sum_{l=1}^{n} C_{i[l]} = (n_i - k_i) (1 + \gamma)\beta + \sum_{l=1}^{k_i} \left( 1 + (1 + \gamma + b) \frac{(1 + b)^{k_i-1} - 1}{b} + (n_i - k_i) (1 + \gamma) (1 + \sigma)(1 + b)^{k_i-1} \right) \\
+ \sum_{l=k_i+1}^{n-1} \left( 1 + (1 + \gamma + b) \frac{(1 + b)^{n-1-l} - 1}{b} \right)(a_{i[l]} - \lambda_{i[l]}u_{i[l]}),
\]

\[
TADC_i = \sum_{l=1}^{n} (2l - 1 - n_l)C_{i[l]} = \sum_{l=k_i+1}^{n} (2l - 1 - n_l) (1 + \gamma)\beta \\
+ \sum_{l=1}^{k_i} \left( 2l - 1 - n_l \right) + \sum_{h=l+1}^{k_i} \left( 2h - 1 - n_l \right) (1 + \gamma + b)(1 + b)^{h-1} + \sum_{h=k_i+1}^{n} \left( 2h - 1 - n_l \right) (1 + \gamma) (1 + \sigma)(1 + b)^{h-1} \\
+ \sum_{l=k_i+1}^{n-1} \left( 2l - 1 - n_l \right) + \sum_{h=l+1}^{n} \left( 2h - 1 - n_l \right) (1 + \gamma + b)(1 + b)^{h-1} \right) \\
+ (n_i - 1) \left( a_{i[n]} - \lambda_{i[n]}u_{i[n]} \right).
\]

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

\[
\delta_1 TML + \delta_2 TC + \delta_3 TADC + \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n} G_{ij}u_{ij} = \\
(1 + \gamma)\beta \sum_{i=1}^{m} \left( \delta_1 + \delta_2 (n_i - k_i) + \delta_3 \sum_{l=k_i+1}^{n} (2l - 1 - n_l) \right) + \\
\sum_{i=1}^{m} \sum_{l=1}^{k_i} \left( \delta_1 (1 + \gamma) (1 + \sigma)(1 + b)^{k_i-1} + \delta_2 \left( 1 + (1 + \gamma + b) \frac{(1 + b)^{k_i-1} - 1}{b} \right) + \sum_{l=k_i+1}^{n} \left( 2l - 1 - n_l \right) (1 + \gamma) (1 + \sigma)(1 + b)^{k_i-1} \right) \\
+ \delta_3 \left( 2l - 1 - n_l \right) + \sum_{h=l+1}^{k_i} \left( 2h - 1 - n_l \right) (1 + \gamma + b)(1 + b)^{h-1} + \sum_{h=k_i+1}^{n} \left( 2h - 1 - n_l \right) (1 + \gamma) (1 + \sigma)(1 + b)^{h-1} \right) \left( a_{i[l]} - \lambda_{i[l]}u_{i[l]} \right).
\]
\[\begin{align*}
&\quad + \sum_{i=1}^{m} \sum_{h=k_i+1}^{n-1} \left( \delta_i (1 + \gamma + b) (1 + b)^{n_i - 1} + \delta_i \frac{1 + (1 + \gamma + b) (1 + b)^{n_i - l} - 1}{b} \right) \\
&\quad + \delta_i \left( 2l - 1 - n_i \right) + \sum_{h=k_i+1}^{n} (2h - 1 - n_i) (1 + \gamma + b) (1 + b)^{h-1} \right) \left( a_{i[l]} - \lambda_{i[l]} u_{i[l]} \right) \\
&\quad + \sum_{i=1}^{m} \left( \delta_i + \delta_i (n_i - k_i) + \delta_i \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) \right) \\
&\quad \lambda_{i[l]} = \begin{cases} \\
\delta_i (1 + \gamma + b) (1 + b)^{n_i - 1} + \delta_i \frac{1 + (1 + \gamma + b) (1 + b)^{n_i - l} - 1}{b} \\
+ \delta_i \left( 2l - 1 - n_i \right) + \sum_{h=k_i+1}^{n_i} (2h - 1 - n_i) (1 + \gamma + b) (1 + b)^{h-1} \right) \\
\delta_i + \delta_i (n_i - k_i) + \delta_i \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) \end{cases} \\
&\quad i = 1, 2, \ldots, m, l = 1, 2, \ldots, k_i, \\
&\quad i = 1, 2, \ldots, m, l = k_i + 1, k_i + 2, \ldots, n_i - 1, \\
&\quad \delta_i + \delta_i (n_i - k_i) \delta_i, \quad i = 1, 2, \ldots, m, l = n_i.
\end{align*}\]

Thus,

\[\begin{align*}
\delta_i TML + \delta_i TC + \delta_i TADC + \delta_i \sum_{j=1}^{m} G_{i[j]} u_{i[j]} \\
= A_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} u_{i[j]} a_{i[j]} + \sum_{i=1}^{m} \sum_{j=1}^{n_i} (G_{i[i]} - \lambda_{i[i]} u_{i[i]}) u_{i[i]}.
\end{align*}\]  

(11)

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

\[u_{i[i]}^* = \begin{cases} \\
\pi_{i[i]}, \quad \text{if } G_{i[i]} - \lambda_{i[i]} u_{i[i]} < 0, \\
\tau_6 \in [0, \pi_{i[i]}], \quad \text{if } G_{i[i]} - \lambda_{i[i]} u_{i[i]} = 0, \\
0, \quad \text{if } G_{i[i]} - \lambda_{i[i]} u_{i[i]} > 0.
\end{cases}\]  

(12)

From equation (12), we can obtain the optimal resource allocation for any given optimal sequence.

Since \(A_i\) is a constant, when \(n_i\) and \(k_i\) are given, we can express the problem as the following assignment problem:
\[
F_1 = A_1 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{n} \theta_{ijl} x_{ijl},
\]

\[
s.t. \sum_{j=1}^{n} x_{ijl} = 1 \quad i = 1, 2, \ldots, m, \quad l = 1, 2, \ldots, n_i
\]

\[
(AP_1) \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ijl} = 1 \quad j = 1, 2, \ldots, n
\]

\[
x_{ijl} = 0 \text{ or } 1 \quad i = 1, 2, \ldots, m, \quad l = 1, 2, \ldots, n_i, \quad j = 1, 2, \ldots, n
\]

\[\theta_{ijl} = \begin{cases} 
  w_{ij} a_{ij} + (G_{ij} - \lambda_{ij} w_{ij}) n_i, & \text{if } G_{ij} - \lambda_{ij} w_{ij} < 0, \\
  w_{ij} a_{ij}, & \text{if } G_{ij} - \lambda_{ij} w_{ij} \geq 0.
\end{cases} \]

\[L_i = (1 + \gamma)^{\beta} + (1 + \gamma)(1 + \sigma) \sum_{l=1}^{k_i} (1 + b)^{k_i-1} (a_{ij} - \lambda_{ij} u_{ij})
\]

\[+ \sum_{l=k_i+1}^{n_i} (1 + \gamma + b)(1 + b)^{n_i-l-1}(a_{ij} - \lambda_{ij} u_{ij}) + \left( a_{ij} - \lambda_{ij} u_{ij} \right), \]

\[\text{TADW}_i = \sum_{l=1}^{n_i} (2l - 1 - n_i) W_{ijl} = k_i (n_i - k_i \beta)
\]

\[+ \sum_{l=k_i+1}^{n_i} (2l + 1 - n_i) \left( (n_i - k_i - 1) (n_i + k_i + 1) + (1 + \sigma) (1 + b)^{k_i-1} (a_{ij} - \lambda_{ij} u_{ij}) \right)
\]

\[+ \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) (1 + b)^{k_i-1} (a_{ij} - \lambda_{ij} u_{ij}), \]

Hence, the sum of total machine load, total waiting times, and total absolute deviation of job waiting times with resource consumption on all the machines is...
Algorithm 1: Algorithm for the problem to minimize the sum of total completion times, total machine loads, the total absolute deviation of job completion times, and the total resource allocation under linear resource consumption.

\[ \text{Let } A_2 = \sum_{i=1}^{m} \beta(\delta_1(1 + \sigma) + \delta_2(n_i - k_i - 1) + \delta_3k_i(n_i - k_i)), \]

\[ \Phi_{[l]} = \begin{cases} 
\delta_1(1 + \gamma)(1 + \sigma)(1 + b)^{k_i-l-1} + \frac{\delta_2}{b}(1 + b)^{k_i-l-1}(1 + \sigma)(1 + b)^{k_i-l-1} - 1) \\
+ \delta_3 \sum_{h=l}^{k_i-1} ((2h + 1 - n_i)(1 + b)^{h-l} + ((n_i - k_i)(n_i + k_i - 1) - (1 + n_i)(n_i - k_i - 1))(1 + \sigma)(1 + b)^{k_i-h}), \\
i = 1, 2, \ldots, m, l = 1, 2, \ldots, k_i, 
\end{cases} \]

Step 1: jobs are scheduled by \((AP_i)\).
Step 2: optimal job resource allocation is calculated by formula (12).
Thus,\[
\delta_1 TML + \delta_2 TW + \delta_3 TADW + \delta_4 \sum_{j=1}^{m} G_{ij} u_{ij} = A_2 + \sum_{j=1}^{m} \sum_{i=1}^{n} \delta_{ij} a_{ij} + \sum_{j=1}^{m} \sum_{i=1}^{n} \left( G_{ij} - \lambda_{ij} \phi_{ij} \right) u_{ij}.
\] (18)

For any job sequence, the optimal resource allocation for a job depends on the sign of $G_{ij} - \lambda_{ij} \phi_{ij}$. Let $u_{ij}^*$ denote the optimal resource allocation for job $J_{ij}$, where

$$
F_2 = A_2 + \min \sum_{j=1}^{m} \sum_{i=1}^{n} \rho_{ij} y_{ij},
$$

s.t. \[
\sum_{j=1}^{n} y_{ij} = 1 \quad i = 1, 2, \ldots, m, \quad l = 1, 2, \ldots, n_i
\]

$$\quad (a_{ij} + (G_{ij} - \phi_{ij} \lambda_{ij}) \rho_{ij}) \quad \rho_{ij} \begin{cases} \phi_{ij} a_{ij}, & \text{if } G_{ij} - \phi_{ij} \lambda_{ij} < 0, \\ \phi_{ij} a_{ij} - \phi_{ij} a_{ij}, & \text{if } G_{ij} - \phi_{ij} \lambda_{ij} \geq 0. \end{cases}$$

(21)

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

$$z \in \left\{ \delta_1 TML + \delta_2 TC + \delta_3 TADC + \delta_4 \sum_{j=1}^{m} G_{ij} u_{ij}, \delta_1 TML + \delta_2 TW + \delta_3 TADW + \delta_4 \sum_{j=1}^{m} G_{ij} u_{ij} \right\}. \quad \text{(22)}$$

Similar to the analysis of problem $P_1$ with convex resource consumption function, i.e. $P_2$ with convex resource consumption function, i.e. $P_3$ with convex resource consumption function, we can indicate the problem as the following assignment problem:

$$H_1 = \delta_1 TML + \delta_2 TC + \delta_3 TADC + \delta_4 \sum_{j=1}^{m} G_{ij} u_{ij} = A_1 + \sum_{j=1}^{m} \sum_{i=1}^{n_i} \left( a_{ij} u_{ij} \right) + \sum_{j=1}^{m} \sum_{i=1}^{n_i} G_{ij} u_{ij}. \quad \text{\quad (23)}$$

From equation (19), we can get the optimal resource allocation for any given optimal sequence.

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

$$u_{ij} = \begin{cases} \phi_{ij}, & \text{if } G_{ij} - \lambda_{ij} \phi_{ij} < 0, \\ u_0 \in \left[ 0, \phi_{ij} \right], & \text{if } G_{ij} - \lambda_{ij} \phi_{ij} = 0, \\ 0, & \text{if } G_{ij} - \lambda_{ij} \phi_{ij} > 0. \end{cases} \quad \text{(19)}$$

4. Cases with Convex Resource Consumption Function

In this section, we will consider the problems under convex resource consumption function, i.e. $P_1$ with convex resource consumption function, i.e. $P_2$ with convex resource consumption function, i.e. $P_3$ with convex resource consumption function.
Step 1: jobs are scheduled by \((AP_2)\).
Step 2: optimal job resource allocation is calculated by formula (19).

**Algorithm 2:** Algorithm for the problem to minimize the sum of total waiting times, total machine loads, the total absolute deviation of job waiting times, and the total resource allocation under linear resource consumption.

where

\[ A_1 = (1 + \gamma) \beta \sum_{i=1}^{m} \left( \delta_1 + \delta_2 (n_i - k_i) + \delta_3 \sum_{l=k_i+1}^{n_i} (2l - 1 - n_i) \right), \]

\[
\begin{align*}
\delta_1 &= (1 + \gamma) (1 + \sigma) (1 + b)^{h-1} + \delta_2 \left( 1 + (1 + \gamma + b) \frac{(1 + b)^{h-1} - 1}{b} + (n_i - k_i) (1 + \gamma) (1 + \sigma) (1 + b)^{h-1} \right) \\
+ \delta_3 \left( (2l - 1 - n_i) + \sum_{h=l+1}^{k_i} (2h - 1 - n_i) (1 + \gamma + b) (1 + b)^{h-1} + \sum_{h=k_i+1}^{n_i} (2h - 1 - n_i) (1 + \gamma) (1 + \sigma) (1 + b)^{h-1} \right), \\
& \quad i = 1, 2, \ldots, m, l = 1, 2, \ldots, k_i, \\
\end{align*}
\]

\[
\begin{align*}
\omega_i[l] = & \left\{ \begin{array}{ll}
\delta_1 (1 + \gamma + b) \left( (1 + b)^{n_i - l - 1} \right) + \delta_2 \left( 1 + (1 + \gamma + b) \frac{(1 + b)^{n_i - l - 1} - 1}{b} \right) \\
+ \delta_3 \left( (2l - 1 - n_i) + \sum_{h=l+1}^{k_i} (2h - 1 - n_i) (1 + \gamma + b) (1 + b)^{h-1} \right), \\
& i = 1, 2, \ldots, m, l = k_i + 1, k_i + 2, \ldots, n_i - 1, \\
\delta_1 + \delta_2 + (n_i - 1) \delta_3, & i = 1, 2, \ldots, m, l = n_i. \\
\end{array} \right.
\]

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

\[
\frac{\partial H_1}{\partial u_{i[l]}} = -nu_{i[l]} \left( a_{i[l]} \right)^{v/u_{i[l]}} + G_{i[l]} = 0
\]

\[
u_{i[l]}^* = \left( \frac{nu_{i[l]}}{G_{i[l]}} \right)^{1/(v+1)} \left( a_{i[l]} \right)^{v/(v+1)}.
\]

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

\[
\begin{align*}
H_1 &= A_1 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{n_i} \xi_{i[j][l]} x_{i[j][l]} \\
\text{s.t.} \quad \sum_{j=1}^{n} x_{i[j][l]} &= 1 \quad i = 1, 2, \ldots, m, l = 1, 2, \ldots, n_i \\
\left( AP_2 \right) &\sum_{i=1}^{m} \sum_{l=1}^{n_i} x_{i[j][l]} = 1 \quad j = 1, 2, \ldots, n \\
x_{i[j][l]} &= 0 \text{ or } 1 \quad i = 1, 2, \ldots, m, l = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n.
\end{align*}
\]
where

$$\xi_{ij} = (v^{-(v+1)} + v^{(1/v+1)}) n_i^{(1/v+1)}(G_{ij}a_{ij})^{(1/v+1)}, \quad i = 1, 2, \ldots, m, j = 1, 2, \ldots, n.$$  

(28)

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

Together with Lemma 1, we have the following theorem.

**Theorem 3.** The problem $P_m|q_{ps,d}, \text{CRA, DMA}|\delta_1 \text{TML} + \delta_2 \text{TC} + \delta_3 \text{TADC} + \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n} G_{ij}u_{ij}$ can be solved in $O(n^{2m+2})$ time.

$$A_2 = \sum_{i=1}^{m} \beta(\delta_1 (1 + \sigma) + \delta_2 (n_i - k_i - 1) + \delta_3 k_i (n_i - k_i)),$$

$$\varphi_{i[l]} = \begin{cases} 
\delta_1 (1 + \gamma) (1 + b)^{k_i-1} + \delta_2 ((n_i - k_i - 1) (1 + \sigma) (1 + b)^{k_i-1} + (1 + b)^{k_i-1} - 1) \frac{1}{b} \\
+ \delta_3 \sum_{h=1}^{k_i-1} ((2h + 1 - n_i) (1 + b)^{h-1} + ((n_i - k_i) (n_i + k_i + 1) - (1 + n_i) (n_i - k_i - 1)) (1 + \sigma) (1 + b)^{h-1}), \\
\delta_1 (1 + \gamma + b) (1 + b)^{n_i-1} + \delta_2 \frac{(1 + b)^{n_i-1} - 1}{b} + \delta_3 \sum_{h=1}^{n_i-1} (2h + 1 - n_i) (1 + b)^{h-1}, \\
i = 1, 2, \ldots, m, l = 1, 2, \ldots, k_i,
\end{cases}$$

(30)

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

$$\frac{\partial H_2}{\partial u_{i[l]}} = -\varphi_{i[l]}(a_{i[l]})^{-1} + G_{i[l]} = 0$$

$$u_{i[l]}^* = \left(\frac{\varphi_{i[l]}}{G_{i[l]}}\right)^{(1/v+1)+(v+1)}(a_{i[l]})^{(v+1)}.$$

(31)

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

$$H_2 = A_2 + \sum_{i=1}^{m} \sum_{l=1}^{n} (v^{-(v+1)} + v^{(1/v+1)}) \varphi_{i[l]}(G_{i[l]}a_{i[l]})^{(v+1)}.$$

(32)

Therefore, we can formulate the minimum problem as the following assignment problem:
Algorithm 3: Algorithm for the problem to minimize the sum of total completion times, total machine loads, the total absolute deviation of job completion times, and the total resource allocation under the convex resource consumption.

\[
H_2 = A_2 + \min \sum \sum \sum \eta_{ijl}[|F_{ijl}|], \quad \text{s.t.} \quad \sum \eta_{ijl}[|F_{ijl}|] = 1 \quad i = 1, 2, \ldots, m, l = 1, 2, \ldots, n_i
\]

(33)

where \( \eta_{ijl}[|F_{ijl}|] = (v_i - (v_i+1) + \nu_i(1/u_i+1)) \eta_{ijl}[G_{ij}], \nu_i(1/u_i+1) \).

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

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

Theorem 4. The problem \( P_{m|d_{pi}, d, \text{CRA, DMA}|1, TML+ \delta_2 TW + \delta_4 \text{TADW} + \delta_4 \sum_{i=1}^{n_i} \sum_{j=1}^{n_i} G_{ij}H_{ij} \text{ can be solved in } O(n^{2m+2}) \) time.

Example. We consider three jobs assigned to two parallel machines to minimize the sum of the total machine loads, the total completion times, the total absolute deviation of job completion times, and the total resource allocation under linear resource allocation. Using the three-field notation, we denote the problem as \( P_{m|d_{pi}, d, \text{LRA, DMA}|1, TML+ \delta_2 TW + \delta_4 \text{TADW} + \delta_4 \sum_{i=1}^{n_i} \sum_{j=1}^{n_i} G_{ij}H_{ij} \). Thus, \( n_1, n_2 \in \{0, 3\}, (1, 2) \) and \( k_i = 0, 1, 2, i = 1, 2 \). Here, we only consider the situation: \( (n_1, n_2) = (1, 2) \) and \( k_i = 1, i = 1, 2 \). Three jobs are \( J_1, J_2, J_3 \) and two machines are \( M_1, M_2 \). Let \( J_{i[i[j]} \) denote job \( J_i \) assigned the \( r \)th position on machine \( M_j \). When \( (n_1, n_2) = (1, 2) \) and \( k_i = 1 \), there are six schedules:

\( \pi_1 = (J_{1[1]}[1], J_{2[2]}[1], J_{3[2]}[2]), \delta_1 = (J_{1[1]}[1], J_{3[2]}[2], J_{2[2]}[2]), \delta_2 = (J_{2[1]}[1], J_{3[2]}[2], J_{1[2]}[2]), \delta_3 = (J_{3[1]}[1], J_{2[1]}[1], J_{1[2]}[2]), \delta_4 = \{0, 3\} \).

By calculation, in scheduling \( \pi_1 \),

\[
\begin{align*}
\omega_1[1][1] &= 0.3 \times (1 + 0.3)(1 + 2)(1 + 2)^{-1} + 0.5 \times \frac{(1 + 1 + 0.3)(1 + 2)(1 + 2)^{-1}}{2} - (1 - 0)(1 + 0.3)(1 + 2)(1 + 2)^{-1} \\
&= 0.8(2 \cdot 1 - 1 - 1) + (2 \cdot 1 - 1 - 1)(1 + 0.3)(1 + 2)(1 + 2)^{-1} = 0.6873,
\end{align*}
\]

\[
\begin{align*}
\omega_2[2][1] &= 0.3 \times (1 + 0.3)(1 + 2)(1 + 2)^{-1} + 0.5 \times \frac{(1 + 1 + 0.3)(1 + 2)(1 + 2)^{-1}}{2} - (2 - 1)(1 + 0.3)(1 + 2)(1 + 2)^{-1} \\
&= 0.8(2 \cdot 1 - 1 - 2)(1 + 0.3)(1 + 2)(1 + 2)^{-1} = 5.94,
\end{align*}
\]

\[
\begin{align*}
\omega_3[2][2] &= 0.3 + 0.5 + (2 - 1) \times 0.8 = 1.6,
\end{align*}
\]

\[
A_1 = (1 + 0.3) \times 5 \times (0.3 + 0.5)(1 - 1) + 0.8(2 \cdot 1 - 1 - 1) + 0.3 + 0.5(2 - 1) + 0.8(2 \cdot 2 - 1 - 2) = 12.35,
\]

\[
G_1 - \lambda_1 \omega_1[1][1] = 2 - 2 \times 0.6873 = 0.6254 > 0, \quad G_2 - \lambda_2 \omega_2[2][1] = 1 - 1 \times 5.94 = -4.94 < 0,
\]

\[
G_3 - \lambda_3 \omega_3[2][2] = 3 - 5 \times 1.6 = -5 < 0.
\]
Step 1: jobs are scheduled by (AP$_1$).
Step 2: optimal job resource allocation is calculated by formula (31).

**Algorithm 4**: Algorithm for the problem to minimize the sum of total waiting times, total machine loads, the total absolute deviation of job waiting times, and the total resource allocation under convex resource consumption.

| Table 1: Other schedule results. | The minimization value of the objective function | The optimal resource allocation ($u_1, u_2, u_3$) |
|---------------------------------|-----------------------------------------------|-----------------------------------------------|
| $\pi_2$                        | 51.4484                                      | (0, 4, 4)                                     |
| $\pi_3$                        | 42.6595                                      | (4, 0, 4)                                     |
| $\pi_4$                        | 42.6595                                      | (4, 0, 4)                                     |
| $\pi_5$                        | 53.95                                        | (4, 4, 4)                                     |
| $\pi_6$                        | 101.6904                                     | (4, 4, 4)                                     |

Therefore, the optimal resource allocation is $u_1 = 0$, $u_2 = 4$, and $u_3 = 4$, and the objective function value is

$$
\delta_1TML(\pi_1) + \delta_2TC(\pi_1) + \delta_3TADC(\pi_1)
+ \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij} = 12.35 + 8 \times 0.6873 + 15 \times 5.94 + 20 \times 1.6 + 4.94 \times 4 - 5 \times 4 = 99.1884.
$$

By the same method, we have the minimization objective function value and the optimal resource allocation of other schedule as Table 1.

From Table 1, it is easy to obtain that the optimal schedule is ($J_{2[1][1]}, J_{1[2][1]}, J_{3[2][1]}, J_{1[1][2]}$) or ($J_{2[1][1]}, J_{3[2][1]}, J_{3[2][1]}, J_{1[1][2]}$). The optimal resource allocation is $u_1 = 4$, $u_2 = 0$, and $u_3 = 4$.

5. Conclusions

In this paper, parallel-machine scheduling problems with past-sequence-dependent delivery times, time-dependent deterioration, maintenance activity, and resource consumption are considered. We present two versions of the scheduling problems can be solved polynomially. Future research will be worth extending to multiple maintenance activities 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] 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.

[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] 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.

[5] 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.

[6] 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.

[7] J. N. D. Gupta and S. K. Gupta, “Single facility scheduling with nonlinear processing times,” *Computers & Industrial Engineering*, vol. 14, no. 4, pp. 387–393, 1988.

[8] S. Browne and U. Yechiali, “Scheduling deteriorating jobs on a single processor,” *Operations Research*, vol. 38, no. 3, pp. 495–498, 1990.

[9] 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.

[10] X.-R. Wang, X. Huang, and J.-B. Wang, “Single-machine scheduling with linear decreasing deterioration to minimize earliness penalties,” *Applied Mathematical Modelling*, vol. 35, no. 7, pp. 3509–3515, 2011.
[11] T. C. E. Cheng, W.-C. Lee, and C.-C. Wu, “Single-machine scheduling with deteriorating jobs and past-sequence-dependent setup times,” *Applied Mathematical Modelling*, vol. 35, no. 4, pp. 1861–1867, 2011.

[12] T. C. E. Cheng, W.-C. Lee, and C.-C. Wu, “Single-machine scheduling with deteriorating functions for job processing times,” *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 4171–4178, 2010.

[13] W.-C. Lee, J.-B. Lin, and Y.-R. Shiu, “Deteriorating job scheduling to minimize the number of late jobs with setup times,” *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 782–787, 2011.

[14] J.-B. Wang and M.-Z. Wang, “Makespan minimization on three-machine flow shop with deteriorating jobs,” *Asia Pacific Journal of Operational Research*, vol. 30, no. 6, Article ID 1350022, 2013.

[15] S.-R. Cheng, “Scheduling two-agents with a time-dependent deterioration to minimize the minsum earliness measures,” *Asia Pacific Journal of Operational Research*, vol. 31, no. 5, Article ID 1450040, 2014.

[16] C. C. Wu, W. C. Lee, and Y. R. Shiu, “A single-machine deteriorating job scheduling problem with a non-regular criterion,” *Asia Pacific Journal of Operational Research*, vol. 28, no. 3, pp. 349–359, 2011.

[17] Y. Q. Yin, T. C. E. Cheng, and C.-C. Wu, “Scheduling with time-dependent processing times,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 201421, 2 pages, 2014.

[18] C. Zhao and H. Tang, “A note to due-window assignment and single machine scheduling with deteriorating jobs and a rate-modifying activity,” *Computers & Operations Research*, vol. 39, no. 6, pp. 1300–1303, 2012.

[19] P.-J. Lai and W.-C. Lee, “Single-machine scheduling with a nonlinear deterioration function,” *Information Processing Letters*, vol. 110, no. 11, pp. 455–459, 2010.

[20] C. Miao, Y. Zhang, and Z. Cao, “Bounded parallel-batch scheduling on single and multi machines for deteriorating jobs,” *Information Processing Letters*, vol. 111, no. 16, pp. 798–803, 2011.

[21] Y. Q. Yin, T. C. E. Cheng, and C.-C. Wu, “Scheduling with time-dependent processing times 2015,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 367585, 2 pages, 2015.

[22] Y. Q. Yin, S.-R. Cheng, J. Y. Chiang, J. C. H. Chen, X. Mao, and C.-C. Wu, “Scheduling problems with due date assignment,” *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 683269, 2 pages, 2015.

[23] 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.

[24] Y. Yin, T. C. E. Cheng, D. Xu, and C.-C. Wu, “Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost,” *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 223–234, 2012.

[25] J.-B. Wang and C.-M. Wei, “Parallel machine scheduling with a deteriorating maintenance activity and total absolute differences penalties,” *Applied Mathematics and Computation*, vol. 217, no. 20, pp. 8093–8099, 2011.

[26] X.-Y. Wang and M.-Z. Wang, “Single machine common flow allowance scheduling with a rate-modifying activity,” *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 898–902, 2010.

[27] 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.

[28] 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.

[29] 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.

[30] C.-J. Hsu, T. C. E. Cheng, and D.-L. Yang, “Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time,” *Information Sciences*, vol. 181, no. 20, pp. 4799–4803, 2011.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] L. Sun, B. Wu, and L. Ning, “Parallel-machine scheduling with DeJong learning effect, delivery times, rate-modifying activity, and resource allocation,” *Shock and Vibration*, vol. 2021, Article ID 6687525, 10 pages, 2021.

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