International Journal of Innovative Computing, Information and Control
Volume **20**, Number **4**, August **2024**

ICIC International ©2024 ISSN 1349-4198
pp. 1169–1183

BICRITERIA DUE-WINDOW ASSIGNMENT AND RESOURCE ALLOCATION SCHEDULING WITH AGING EFFECT AND A DETERIORATING RATE-MODIFYING ACTIVITY

Xiaoli Zhao

School of Science
Shenyang Aerospace University
No. 37, Daoyi South Avenue, Shenbei New Area, Shenyang 110136, P. R. China
zhaoxiaoli@email.sau.edu.cn

Received September 2023; revised January 2024

Abstract. *In some manufacturing systems, it is important to study just-in-time (JIT) scheduling problems that include both earliness and tardiness penalty. This paper investigates single-machine scheduling problems with simultaneous considerations of due-window assignment, position-dependent and convex resource-dependent processing times, and a deteriorating rate-modifying activity. The goal is to determine the optimal position of due-window, resource allocations, job processing sequence, and position of the deteriorating rate-modifying activity that minimize the total schedule penalty cost under the condition that the total weighted resource consumption cost does not exceed a given upper bound, and the inverse version, where the total schedule penalty cost is a weighted sum function of earliness, tardiness, due-window starting time, due-window size, makespan and total completion time. For the three most widely used due-window assignment types, i.e., the common due-window (CONW), the slack due-window (SLKW), and the different due-window (DIFW), we show that all the considered problems can be solved in polynomial time.*

Keywords: Single-machine scheduling, Due-window assignment, Aging effect, Resource allocation, Deteriorating rate-modifying activity

1. **Introduction.** Just-in-time (JIT) scheduling with due-window assignment has become a popular topic in the scheduling literature. In this setting, if a job is completed before its due-window, it will incur earliness penalty such as holding costs. While if a job is completed after its due-window, it will incur tardiness penalties such as a contractual penalty and a loss of customer goodwill. Therefore, the decision-maker has to balance among the earliness penalties, tardiness penalties and other production costs. For scheduling with due-window assignment, the starting time and the size of due-window are decision variables. In this paper, we consider three of the most widely used due-window assignment types, i.e., the common due-window (CONW) assignment where all jobs share a common due-window, see Liman et al. [1], the slack due-window (SLKW) assignment where all jobs have a common flow allowance or a slack, see Mosheiov and Oron [2], and the different due-window (DIFW) assignment where each job has its own due-window, see Shabtay and Steiner [3]. For more on scheduling problems with due-windows, the reader can refer to [4-7].

In the actual industrial production process, the job processing times may be variable due to learning/aging effects and/or deterioration effects and/or controllable processing times (resource allocation) and so on. Mosheiov [8] first introduced the scheduling problems with aging effects. For more research results on scheduling models with learning or aging effects,

DOI: 10.24507/ijicic.20.04.1169

the reader can refer to [9-11]. In addition, the job processing time is variable due to the allocation of additional resources, such as energy, and manpower. There are two resource consumption functions in the literature, namely the linear resource consumption function and the convex resource consumption function. An extensive survey of the scheduling problems with resource-dependent processing time can be found in [12-15].

Recently, many scholars have studied the combination of the learning/aging effect, resource allocation and due-window assignment. Wang and Wang [16] considered a single-machine CONW scheduling problem, in which the processing time of a job is job-dependent learning effect and resource allocation. For a linear and a convex resource allocation, the authors provided a polynomial time algorithm to minimize the total weighted sum of earliness, tardiness, the window location, window size and total weighted resource allocation cost, respectively. Wang et al. [17] studied the constrained objective model of [16] for the convex resource allocation model. They showed that the problem can be solved in optimal time for minimizing the total weighted sum of earliness, tardiness, the window location and window size under the constraint that the total weighted resource allocation cost does not exceed a given upper bound. Yin [18] studied the inverse version of [17] and also provided a polynomial time algorithm to solve it. Li et al. [19] considered a single-machine SLKW scheduling problem where the actual processing time of a job is a function of job-dependent learning effect and resource allocation. For a linear and a convex resource allocation, they provided a polynomial time algorithm to minimize the weighted sum of earliness, tardiness, the window location, window size, makespan and total weighted resource allocation cost, respectively. Liu et al. [20] considered a single-machine scheduling problem with convex resource consumption processing times and due-window assignment. For the SLKW and CONW types, they provided a polynomial time algorithm to minimize the total weighted resource consumption cost under the constraint that the total schedule cost involving earliness, tardiness, window location, window size and makespan does not exceed a given limit, respectively. And they generalized their conclusions to the scheduling with the job-independent learning effect. Liu et al. [21] considered single machine scheduling problems with three due-window assignment types and resource allocation where the actual job processing time is a general function of its position, starting time, and the amount of a resource allocated for the job. For a linear or a convex resource allocation, they provided polynomial time algorithms to solve various combinations of objectives.

Moreover, the scheduling with rate-modifying activities (RMAs) has been extensively studied in the literature. After the rate-modifying activity, the machine will enhance production efficiency. Lee and Leon [22] first studied the single machine scheduling with a rate-modifying activity (RMA). Kubzin and Strusevich [23] first proposed the scheduling with a deteriorating rate-modifying activity. Jia et al. [24] considered SLKW assignment problem with a deteriorating rate-modifying activity.

To the best of our knowledge, there exist a few results concerning scheduling problems with job-dependent learning/aging effects, resource allocation, a deteriorating rate-modifying activity and due-window assignment simultaneously. Ji et al. [25] presented the optimal algorithms on a single-machine with the CONW type, resource allocation, aging effect, and a deteriorating rate-modifying activity. The objective is the linear combination of earliness, tardiness, due-window starting time, due-window size, and resource consumption. Cheng and Cheng [26] considered a similar scheduling problem with the SLKW type and a linear resource consumption. However, the scheduling problem with simultaneous considerations of constrained optimization objective function and the above four characteristics has never been investigated. Research on the scheduling problem with constrained optimization objective has potential impact on the actual production process.

In general, a better schedule for an objective function necessarily results in a worse schedule for the other objective function. Hence, we focus our attention on minimizing the total schedule penalty cost subject to a constraint on the total weighted resource consumption cost, and the inverse version problem in this paper.

The main contributions of this paper are as follows.

1) We consider six constrained optimization scheduling problems with due-window assignment, job-dependent aging effects, resource allocation and a deteriorating rate-modifying activity.

2) We present polynomial time algorithms for the CONW, SLKW and DIFW types of the proposed problem, respectively.

3) The time complexity of the proposed algorithms is analyzed.

Next, we summarize and distinguish some previous research results similar to those in this paper, please see Table 1. The symbols in the table refer to the problem formulation section of the paper.

Table 1. Summary and distinction of some previous research results with this paper (Papers about due-window assignment scheduling with learning/aging effect and convex resource allocation)

Problem	Due-window ass.	Complexity
[16]		
$1\vert p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k \Big\vert \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d^1 + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$	CONW	$O(n^3)$
[17]		
$1\vert p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n} G_j u_j \le U \Big\vert \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d^1 + \delta D)$	CONW	$O(n^3)$
[18]		
$1\vert p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d^1 + \delta D) + \eta C_{\max} \le V \Big\vert \sum_{j=1}^{n} G_j u_j$	CONW	$O(n^3)$
[19]		
$1\vert p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k \Big\vert \sum_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^1 + \delta D_j\right) + \eta C_{\max} + \theta \sum_{j=1}^{n} G_j u_j$	SLKW	$O(n^3)$
[20]		
$1\vert p_j = \left(\frac{\bar{p}_j r^{a}}{u_j}\right)^k, \sum_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^1 + \delta D_j\right) + \eta C_{\max} \le V \Big\vert \sum_{j=1}^{n} G_j u_j$	SLKW/CONW	$O(n \log n)$
[25]		
$1\vert rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k \Big\vert \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d^1 + \delta D) + \theta \sum_{j=1}^{n} G_j u_j$	CONW	$O(n^4)$
This paper		
$1\vert rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n} G_j u_j \le U \Big\vert \sum_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^1 + \delta D_j\right) + \eta C_{\max} + \sigma \sum_{j=1}^{n} C_j$	CONW/SLKW/DIFW	$O(n^4)$
$1\vert rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, \sum_{j=1}^{n} \left(\alpha E_j + \beta T_j + \gamma d_j^1 + \delta D_j\right) + \eta C_{\max} + \sigma \sum_{j=1}^{n} C_j \le V \Big\vert \sum_{j=1}^{n} G_j u_j$	CONW/SLKW/DIFW	$O(n^4)$

The rest of this paper is organized as follows. In Section 2, notations and problem descriptions are given. Section 3 presents preliminary results for three due-window assignment types. In Sections 4, 5 and 6, we propose polynomial time algorithms for the CONW, SLKW and DIFW types, respectively. Section 7 presents some conclusions.

2. **Problem Formulation.**

There is a set of n independent and non-preemptive jobs $J = \{J_1, J_2, \ldots, J_n\}$ to be processed on a single machine. All jobs are available for processing at time zero. The machine can process one job at a time. The actual processing time of job

J_j scheduled in the rth position in a sequence is $p_{jr}(u)=\left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k$, $j,r=1,2,\ldots,n$, where $\bar{p}_j$ is the basic (noncompressed) processing time of job J_j, $a_j\geq 0$ is the position-dependent aging effect, $u_j>0$ is the amount of the resource allocated to job J_j, and k is a positive constant. The machine has at most one RMA. During the RMA period, the machine is idle and no production is performed. The RMA duration is defined $f(t)=b+ct$, where $b>0$ is the basic RMA time, $c\geq 0$ is the RMA deterioration rate, and t is the starting time of the RMA. After the RMA, the machine will revert to its initial condition. If the job at position i is finished and the RMA is performed immediately, then let us say the position of RMA is i. If the job J_j is processed in the rth position after the RMA, then its actual processing time is $p_{jr}(u)=\left(\frac{\lambda_j\bar{p}_j(r-i)^{a_j}}{u_j}\right)^k$, where λ_j $(0<\lambda_j\leq 1)$ is the modifying rate of job J_j.

Let $\left[d_j^1,d_j^2\right]$ be the due-window of job J_j $(j=1,2,\ldots,n)$ such that $d_j^1\leq d_j^2$, and $D_j=d_j^2-d_j^1$ be the due-window size. For a given schedule, let C_j be the completion time of job J_j, $E_j=\max\left\{0,d_j^1-C_j\right\}$ be the earliness, $T_j=\max\left\{0,C_j-d_j^2\right\}$ be the tardiness, $C_{\max}$ be the maximum completion time (makespan), and $\sum_{j=1}^n C_j$ be the total completion time. We define the total schedule penalty cost $f=\sum_{j=1}^n\left(\alpha E_j+\beta T_j+\gamma d_j^1+\delta D_j\right)+\eta C_{\max}+\sigma\sum_{j=1}^n C_j$, where $\alpha(>0)$, $\beta(>0)$, $\gamma(>0)$, $\delta(>0)$, $\eta(>0)$, $\sigma(>0)$ are the per unit time penalties for earliness, tardiness, due-window starting time, due-window size, makespan, and total completion time, respectively.

In this study, we aim to determine the optimal job sequence π^*, the optimal due-window starting time d_j^1, the due-window size D_j, the optimal position i of the RMA, and the optimal resource allocation u_j^* to minimize the following objectives:

i) the total schedule penalty cost f subject to the total weighted resource consumption cost $\sum_{j=1}^n G_j u_j\leq U$, where G_j is the per unit cost of the resource allocated to job J_j and $U>0$ is a given upper bound on the total weighted resource consumption cost;

ii) the total weighted resource consumption cost $\sum_{j=1}^n G_j u_j$ subject to $f\leq V$, where $V>0$ is a given upper bound on the total schedule penalty cost.

In addition, our study is conducted under three of the most familiar due-window assignment types:

- The CONW assignment, where all jobs share a common due-window (Mosheiov and Sarig [27]), that is $d_j^1=d^1$, $d_j^2=d^2$ and $D_j=d^2-d^1=D$ for $j=1,2,\ldots,n$;
- The SLKW assignment, where the due-window for each job is dependent on the job. The starting time of the job-dependent due-window is equal to the sum of the job processing time and a job-independent constant, the finishing time of the due-window is equal to the sum of the job processing time and a larger job-independent constant (Mor and Mosheiov [28]). That is $d_j^1=p_j+q^1$, $d_j^2=p_j+q^2$ and $D_j=q^2-q^1$ for $j=1,2,\ldots,n$;
- The DIFW assignment, where each job can be assigned a different due-window with no restrictions.

Using the three-field notation, our problems can be denoted as follows:

Problem P1: $1\left|rma,p_j=\left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, CONW, \sum\limits_{j=1}^n G_j u_j\leq U\right|f$;

Problem P2: $1\left|rma,p_j=\left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, CONW, f\leq V\right|\sum\limits_{j=1}^n G_j u_j$;

Problem P3: $1\left|rma,p_j=\left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, SLKW, \sum\limits_{j=1}^n G_j u_j\leq U\right|f$;

Problem P4: $1|rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, SLKW, f \le V \left| \sum_{j=1}^{n} G_j u_j \right.$;

Problem P5: $1|rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, DIFW, \sum_{j=1}^{n} G_j u_j \le U|f$;

Problem P6: $1|rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k, DIFW, f \le V \left| \sum_{j=1}^{n} G_j u_j \right.$;

where rma means "a deteriorating rate-modifying activity".

3. **Preliminary Results.** In this section, we provide some main results for further analysis of problems. For any sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$, let $J_{[j]}$, $j = 1, 2, \ldots, n$ denote the job scheduled in the jth position.

Lemma 3.1. *For the problem* $1|rma, p_j = \left(\frac{\bar{p}_j r^{a_j}}{u_j}\right)^k|f$*, there hold the following properties.*

(1) There exists an optimal schedule in which all jobs are processed consecutively without any idle time from time zero.

(2) For the CONW type, there exists an optimal schedule in which the common due-window starting time $d^1 = C_{[h]}$ *and the due-window finishing time* $d^2 = C_{[l]}$ $(l \ge h)$*, where* $h = \max\left\{\left\lceil \frac{n(\delta-\gamma)}{\alpha} \right\rceil, 0\right\}$ *and* $l = \max\left\{\left\lceil \frac{n(\beta-\delta)}{\beta} \right\rceil, 0\right\}$*, here* $\lceil x \rceil$ *denotes the smallest integer greater than or equal to* x*.*

(3) For the SLKW type, if $C_{[j]} \le d^1_{[j]}$*, then* $C_{[j-1]} \le d^1_{[j-1]}$*; if* $C_{[j]} \ge d^1_{[j]}$*, then* $C_{[j+1]} \ge d^1_{[j+1]}$*; the optimal values* $q^1 = C_{[h]}$ *and* $q^2 = C_{[l]}$ $(l \ge h)$*, where* $h = \max\left\{\left\lfloor \frac{n(\delta-\gamma)}{\alpha} \right\rfloor, 0\right\}$ *and* $l = \max\left\{\left\lfloor \frac{n(\beta-\delta)}{\beta} \right\rfloor, 0\right\}$*, here* $\lfloor x \rfloor$ *denotes the largest integer less than or equal to* x*.*

(4) For the DIFW type, if $\gamma \ge \beta$*, then* $d^1_{[j]} = d^2_{[j]} = 0$*, otherwise* $d^1_{[j]} = d^2_{[j]} = C_{[j]}$ $(j = 1, 2, \ldots, n)$ *in an optimal schedule.*

Proof: The correctness of (1) is obvious. For the CONW type, the proof is similar to Lemmas 1-3 of [27]. For the SLKW type, the proof is similar to Properties 1-4 of [2], Properties 1-3 of [28] and Property 2 of Wu et al. [29]. For the DIFW type, the proof is similar to Seidmann et al. [30] and Lemma 4 of [3]. Here we omit the details of the proof.

4. **The Common Due-Window (CONW) Assignment Type.** In this section, we study problems P1 and P2. From Lemma 3.1(2), we know that the values h and l do not depend on the processing times of any jobs and the processing sequence, and they only depend on the per time unit penalty parameters and the total number of jobs n. Hence, the results still hold for our problems P1 and P2. Based on the notations and Lemma 3.1, we suppose that the deteriorating rate-modifying activity is scheduled in position i, and we have the following lemma.

Lemma 4.1. *For the problems P1 and P2, the total schedule penalty cost* f *can be written as*

$$f = \sum_{j=1}^{i} \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k + \sum_{j=i+1}^{n} \omega_j \left(\frac{\lambda_{[j]} \bar{p}_{[j]} (j-i)^{a_{[j]}}}{u_{[j]}}\right)^k + W, \tag{1}$$

where

$$W = \begin{cases} \alpha i b + n\gamma b + \eta b + \sigma(n-i)b, & \text{if } i < h \\ n\delta b + \eta b + \sigma(n-i)b, & \text{if } h \le i < l \\ \beta(n-i)b + \eta b + \sigma(n-i)b, & \text{if } l \le i \le n \end{cases} \tag{2}$$

when $i < h$,

$$\omega_j = \begin{cases} \alpha(j-1) + \alpha ic + n\gamma + n\gamma c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, i \\ \alpha(j-1) + n\gamma + \eta + \sigma(n-j+1), & j = i+1, \ldots, h \\ n\delta + \eta + \sigma(n-j+1), & j = h+1, \ldots, l \\ \beta(n-j+1) + \eta + \sigma(n-j+1), & j = l+1, \ldots, n \end{cases} \quad (3)$$

when $h \le i < l$,

$$\omega_j = \begin{cases} \alpha(j-1) + n\gamma + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, h \\ n\delta + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = h+1, \ldots, i \\ n\delta + \eta + \sigma(n-j+1), & j = i+1, \ldots, l \\ \beta(n-j+1) + \eta + \sigma(n-j+1), & j = l+1, \ldots, n \end{cases} \quad (4)$$

when $l \le i \le n$,

$$\omega_j = \begin{cases} \alpha(j-1) + \beta(n-i)c + n\gamma + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, h \\ \beta(n-i)c + n\delta + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = h+1, \ldots, l \\ \beta(n-j+1) + \beta(n-i)c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = l+1, \ldots, i \\ \beta(n-j+1) + \eta + \sigma(n-j+1), & j = i+1, \ldots, n \end{cases} \quad (5)$$

Here when $i = n$ *it means no RMA.*

Proof: By the position of the RMA, we obtain the RMA duration is $f(t) = b + ct = b + cC_{[i]} = b + c\sum_{j=1}^{i} p_{[j]}$. All jobs before the job $J_{[h]}$ are early and all jobs after the job $J_{[l]}$ are tardy.

(i) If $i < h$, according to Lemma 3.1(2), $d^1 = C_{[h]} = \sum_{j=1}^{h} p_{[j]} + f(t)$, $d^2 = C_{[l]} = \sum_{j=1}^{l} p_{[j]} + f(t)$. The total schedule penalty cost is

$$\begin{aligned} f &= \alpha\left(\sum_{j=1}^{h}(j-1)p_{[j]} + if(t)\right) + \beta\sum_{j=l+1}^{n}(n-j+1)p_{[j]} + n\gamma\left(\sum_{j=1}^{h}p_{[j]} + f(t)\right) \\ &\quad + n\delta\sum_{j=h+1}^{l}p_{[j]} + \eta\left(\sum_{j=1}^{n}p_{[j]} + f(t)\right) + \sigma\left(\sum_{j=1}^{n}(n-j+1)p_{[j]} + (n-i)f(t)\right) \\ &= (\alpha ib + n\gamma b + \eta b + \sigma(n-i)b) \\ &\quad + \sum_{j=1}^{i}(\alpha(j-1) + \alpha ic + n\gamma + n\gamma c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]}j^{a_{[j]}}}{u_{[j]}}\right)^k \\ &\quad + \sum_{j=i+1}^{h}(\alpha(j-1) + n\gamma + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k \\ &\quad + \sum_{j=h+1}^{l}(n\delta + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k \\ &\quad + \sum_{j=l+1}^{n}(\beta(n-j+1) + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k \end{aligned}$$

(ii) If $h \le i < l$, then $d^1 = C_{[h]} = \sum_{j=1}^{h} p_{[j]}$ and $d^2 = C_{[l]} = \sum_{j=1}^{l} p_{[j]} + f(t)$. The total schedule penalty cost is

$$f = \alpha\sum_{j=1}^{h}(j-1)p_{[j]} + \beta\sum_{j=l+1}^{n}(n-j+1)p_{[j]} + n\gamma\sum_{j=1}^{h}p_{[j]}$$

$$
\begin{aligned}
&+ n\delta\left(\sum_{j=h+1}^{l} p_{[j]} + f(t)\right) + \eta\left(\sum_{j=1}^{n} p_{[j]} + f(t)\right) + \sigma\left(\sum_{j=1}^{n}(n-j+1)p_{[j]} + (n-i)f(t)\right) \\
&= (n\delta b + \eta b + \sigma(n-i)b) \\
&+ \sum_{j=1}^{h}(\alpha(j-1) + n\gamma + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=h+1}^{i}(n\delta + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=i+1}^{l}(n\delta + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=l+1}^{n}(\beta(n-j+1) + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k
\end{aligned}
$$

(iii) If $l \le i \le n$, then $d^1 = C_{[h]} = \sum_{j=1}^{h} p_{[j]}$ and $d^2 = C_{[l]} = \sum_{j=1}^{l} p_{[j]}$. The total schedule penalty cost is

$$
\begin{aligned}
f &= (\beta(n-i)b + \eta b + \sigma(n-i)b) \\
&+ \sum_{j=1}^{h}(\alpha(j-1) + \beta(n-i)c + n\gamma + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=h+1}^{l}(\beta(n-i)c + n\delta + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=l+1}^{i}(\beta(n-j+1) + \beta(n-i)c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c)\left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k \\
&+ \sum_{j=i+1}^{n}(\beta(n-j+1) + \eta + \sigma(n-j+1))\left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k
\end{aligned}
$$

Denote W and ω_j as in Equations (2)-(5) in Lemma 4.1, and the conclusion holds.

According to Lemma 4.1, the total schedule penalty cost f is a decreasing continuous function of u_j. For the problem P1, when the constraint $\sum_{j=1}^{n} G_j u_j = U$, the amount of resource u_j is maximized. So the total schedule penalty cost f is minimized. We can obtain an optimal resource allocation for the problem P1 by the following lemma.

Lemma 4.2. *For a given sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$, the optimal resource allocation $u^*(\pi)$ of the problem P1 as a function of the job sequence is given by*

$$
u^*_{[j]} = \begin{cases}
\dfrac{U(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\right)^{-\frac{1}{k+1}}\left(\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}}}{\sum_{j=1}^{i}(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n}(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}}, & j = 1, 2, \ldots, i \\[2ex]
\dfrac{U(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\right)^{-\frac{1}{k+1}}\left(\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}}{\sum_{j=1}^{i}(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n}(\omega_j)^{\frac{1}{k+1}}\left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}}, & j = i+1, \ldots, n
\end{cases} \tag{6}
$$

where ω_j are given by Equations (3)-(5).

Proof: For a given sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$, let ρ be the Lagrangian multiplier, then according to Lemma 4.1, the Lagrange function can be formulated as

$$L(u,\rho) = \sum_{j=1}^{i} \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k + \sum_{j=i+1}^{n} \omega_j \left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k + W + \rho\left(\sum_{j=1}^{n} G_{[j]}u_{[j]} - U\right) \quad (7)$$

Deriving Equation (7) with respect to $u_{[j]}$ and ρ, respectively, we have

$$\frac{\partial L(u,\rho)}{\partial u_{[j]}} = \begin{cases} \rho G_{[j]} - \dfrac{k\omega_j \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^k}{(u_{[j]})^{k+1}} = 0, & j = 1,2,\ldots,i \\ \rho G_{[j]} - \dfrac{k\omega_j \left(\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^k}{(u_{[j]})^{k+1}} = 0, & j = i+1,\ldots,n \end{cases} \quad (8)$$

and

$$\frac{\partial L(u,\rho)}{\partial \rho} = \sum_{j=1}^{n} G_{[j]}u_{[j]} - U = 0 \quad (9)$$

Using Equation (8), we have

$$u_{[j]} = \begin{cases} \left(\dfrac{k\omega_j}{\rho G_{[j]}}\right)^{\frac{1}{k+1}} \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}}, & j = 1,2,\ldots,i \\ \left(\dfrac{k\omega_j}{\rho G_{[j]}}\right)^{\frac{1}{k+1}} \left(\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}, & j = i+1,\ldots,n \end{cases} \quad (10)$$

Then substitute Formula (10) into Formula (9) to obtain the following expression:

$$\rho^{\frac{1}{k+1}} = \frac{\sum_{j=1}^{i}(k\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n}(k\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}}{U} \quad (11)$$

Finally, inserting Equation (11) into Equation (10), we obtain $u^*_{[j]}$ in Equation (6).

According to Lemma 4.2, substituting Equation (6) into the total schedule penalty cost expression given in Equation (1), then we have the following lemma.

Lemma 4.3. *For the problem P1, given that an optimal resource allocation $u^*_{[j]}$ $(j = 1,2,\ldots,n)$ is chosen, then the total schedule penalty cost is given by the following expression:*

$$f = U^{-k}\left(\sum_{j=1}^{i}(\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n}(\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}\right)^{k+1} + W, \quad (12)$$

where W and ω_j are given by Equations (2)-(5) in Lemma 4.1.

Observing the above expression of the total schedule penalty cost, for a given position i of the RMA, the values W, U and k are all constants, and we can see that minimizing the problem P1 is equivalent to minimizing the following objective:

$$Z = \sum_{j=1}^{i}(\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n}(\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}. \quad (13)$$

Let x_{jr} be a 0/1 variable such that $x_{jr} = 1$ if job J_j is scheduled in position r, and $x_{jr} = 0$, otherwise. For a given position i of the RMA, minimizing Z is equivalent to minimizing the following assignment problem:

$$\min Z = \min \sum_{j=1}^{n} \left(\sum_{r=1}^{i} (\omega_r)^{\frac{1}{k+1}} \left(G_j \bar{p}_j r^{a_j}\right)^{\frac{k}{k+1}} x_{jr} + \sum_{r=i+1}^{n} (\omega_r)^{\frac{1}{k+1}} \left(G_j \lambda_j \bar{p}_j (r-i)^{a_j}\right)^{\frac{k}{k+1}} x_{jr} \right) \tag{14}$$

subject to $\sum_{j=1}^{n} x_{jr} = 1$, $r = 1, 2, \ldots, n$; $\sum_{r=1}^{n} x_{jr} = 1$, $j = 1, 2, \ldots, n$; $x_{jr} = 0$ or 1, $j, r = 1, 2, \ldots, n$. The first set of constraints guarantees that each position will be assigned only one job, while the second set of constraints guarantees that each job will be assigned to only one position. The third set of constraints means x_{jr} is a binary variable.

Based on the above analysis, we propose the following polynomial time algorithm to solve optimally the problem P1.

Algorithm 4.1

Step 1: By Lemma 3.1(2), calculate the optimal positions h and l of the common due-window starting time d^1 and finishing time d^2, respectively. Set $i = 0$ and $Z = \infty$.

Step 2: Calculate the weights in front of x_{jr} according to Equations (3)-(5).

Step 3: Solve the assignment problem (14) to obtain the local optimal sequence $\pi(i)$ and the corresponding function value $Z(i)$. If $Z(i) < Z$, then set $Z = Z(i)$.

Step 4: Set $i = i + 1$. If $i > n$, then go to Step 5. Otherwise, go to Step 2.

Step 5: The global optimal sequence is the one with the minimum function value Z, and we denote the optimal sequence as π^* and the optimal position of the RMA as i^*.

Step 6: Calculate the minimum total schedule penalty cost f by Equation (12), the optimal resource allocations by Equation (6) and the corresponding actual job processing times.

Theorem 4.1. *Algorithm 4.1 solves optimally the problem P1 in $O(n^4)$ time.*

Proof: The correctness of the algorithm follows from the above analysis. In step 3, for a given i, solving an assignment problem of size n requires $O(n^3)$ time [31,32]. Furthermore, the position of the RMA is variable and step 4 is executed $(n + 1)$ times. Consequently, the overall time requirement of Algorithm 4.1 is $O(n^4)$.

For the problem P1, we solve the following instance to illustrate Algorithm 4.1.

Example 4.1. *There are $n = 7$ jobs. The unit penalties are $\alpha = 10$, $\beta = 15$, $\gamma = 3$, $\delta = 5$, $\eta = 2$ and $\sigma = 1$, respectively. $k = 2$, $b = 4$, $c = 0.2$ and $U = 200$. The other data are given in Table 2.*

Table 2. The data of Example 4.1

J_j	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$\bar{p}_j$	4	12	13	7	9	16	6
a_j	0.3	0.1	0.15	0.25	0.1	0.2	0.5
λ_j	0.65	0.5	0.7	0.4	0.85	0.6	0.3
G_j	7	10	13	8	4	2	11

According to Lemma 3.1(2), $h = \max\left\{\left\lceil \frac{n(\delta-\gamma)}{\alpha} \right\rceil, 0\right\} = 2$ and $l = \max\left\{\left\lceil \frac{n(\beta-\delta)}{\beta} \right\rceil, 0\right\} = 5$. In the following, we only explain the case that the position for the RMA is $i = 1$ and the other cases are similar. We can compute that $W = 156$, $\omega_1 = 37.8$, $\omega_2 = 39$, $\omega_3 = 42$,

$\omega_4 = 41$, $\omega_5 = 40$, $\omega_6 = 34$, $\omega_7 = 18$ by Equations (2) and (3), respectively. The weight matrix of the assignment problem (14) is given as follows:

$$\begin{pmatrix} 30.9459 & 23.4640 & 27.6272 & 29.7212 & 31.2234 & 30.9268 & 25.9479 \\ 81.6491 & 51.9743 & 55.7938 & 56.8639 & 57.4899 & 55.2745 & 45.2621 \\ 102.5861 & 81.7231 & 89.7792 & 92.7463 & 94.6708 & 91.7022 & 75.5491 \\ 49.1235 & 26.9476 & 31.0042 & 32.9064 & 34.2396 & 33.6630 & 28.0725 \\ 36.5903 & 33.1769 & 35.6150 & 36.2981 & 36.6977 & 35.2835 & 28.8923 \\ 33.8270 & 24.3158 & 27.3372 & 28.6250 & 29.5005 & 28.7888 & 23.8623 \\ 54.8100 & 24.8198 & 32.0531 & 36.3981 & 39.7329 & 40.5440 & 34.8539 \end{pmatrix}$$

We obtain the local optimal job sequence is $\pi(1) = (J_5, J_7, J_1, J_4, J_6, J_2, J_3)$ and the corresponding minimum value of Z is $Z(1) = 282.2678$. The results for the RMA in other positions are summarized as follows:

$$\begin{aligned} &\pi(0) = (J_3, J_7, J_1, J_4, J_6, J_5, J_2),\ Z(0) = 282.4702;\\ &\pi(2) = (J_1, J_5, J_7, J_4, J_6, J_2, J_3),\ Z(2) = 285.7155;\\ &\pi(3) = (J_1, J_6, J_5, J_7, J_4, J_2, J_3),\ Z(3) = 295.9955;\\ &\pi(4) = (J_4, J_1, J_6, J_5, J_7, J_2, J_3),\ Z(4) = 321.7037;\\ &\pi(5) = (J_3, J_4, J_1, J_6, J_5, J_7, J_2),\ Z(5) = 355.9553;\\ &\pi(6) = (J_7, J_4, J_1, J_6, J_5, J_2, J_3),\ Z(6) = 391.8418;\\ &\pi(7) = (J_7, J_4, J_1, J_6, J_5, J_2, J_3),\ Z(7) = 417.7234. \end{aligned}$$

From Algorithm 4.1, the global optimal schedule is $\pi^* = \pi(1) = (J_5, J_7, J_1, J_4, J_6, J_2, J_3)$. By Equation (12), the corresponding minimum total schedule penalty cost is $f = 718.2430$. The corresponding optimal resource allocations are $u^*_{[j]} = (6.4815, 1.5987, 2.7965, 2.9145, 10.4512, 3.9165, 4.1177)$, the corresponding actual processing times are $p_{[j]} = (1.9281, 1.2677, 1.3102, 1.5986, 1.4690, 3.2382, 8.3602)$, the optimal position of the RMA is $i^* = 1$ and the RMA duration is $f(t) = 4.3856$, the starting time and the finishing time of the common due-window are $d^1 = C_{[2]} = C_7 = 7.5814$ and $d^2 = C_{[5]} = C_6 = 11.9592$, respectively, and the due-window size is $D = 4.3778$.

In the following we consider the problem P2. According to Lemma 4.1, the total schedule penalty cost f is a decreasing continuous function of u_j. When the constraint $f = V$, the amount of resource allocation u_j is minimized, so the total weighted resource cost $\sum_{j=1}^n G_j u_j$ is minimized.

Lemma 4.4. *For a given sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$, the optimal resource allocation $u^*(\pi)$ of the problem P2 as a function of the job sequence can be obtained by the following expression, which minimizes the total weighted resource cost $\sum_{j=1}^n G_j u_j$.*

$$u^*_{[j]} = \begin{cases} \left(\dfrac{Z}{V-W}\right)^{\frac{1}{k}} (\omega_j)^{\frac{1}{k+1}} (G_{[j]})^{-\frac{1}{k+1}} \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}}, & j = 1, 2, \ldots, i \\ \left(\dfrac{Z}{V-W}\right)^{\frac{1}{k}} (\omega_j)^{\frac{1}{k+1}} (G_{[j]})^{-\frac{1}{k+1}} \left(\lambda_{[j]} \bar{p}_{[j]} (j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}, & j = i+1, \ldots, n \end{cases} \tag{15}$$

where Z and ω_j are given by Equations (13) and (3)-(5), respectively.

Proof: For a given sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$, let ρ be the Lagrangian multiplier, and then according to Lemma 4.1, the Lagrange function can be formulated as

$$L(u, \rho) = \sum_{j=1}^n G_{[j]} u_{[j]} + \rho \left(\sum_{j=1}^i \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}} \right)^k + \sum_{j=i+1}^n \omega_j \left(\frac{\lambda_{[j]} \bar{p}_{[j]} (j-i)^{a_{[j]}}}{u_{[j]}} \right)^k \right.$$

$$+W - V\Bigg) \tag{16}$$

Deriving Equation (16) with respect to $u_{[j]}$ and ρ, respectively, we have

$$\frac{\partial L(u,\rho)}{\partial u_{[j]}} = \begin{cases} G_{[j]} - \rho \dfrac{k\omega_j \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^k}{\left(u_{[j]}\right)^{k+1}} = 0, & j = 1, 2, \ldots, i \\ G_{[j]} - \rho \dfrac{k\omega_j \left(\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^k}{\left(u_{[j]}\right)^{k+1}} = 0, & j = i+1, \ldots, n \end{cases} \tag{17}$$

and

$$\frac{\partial L(u,\rho)}{\partial \rho} = \sum_{j=1}^{i} \omega_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}}\right)^k + \sum_{j=i+1}^{n} \omega_j \left(\frac{\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}}{u_{[j]}}\right)^k + W - V = 0 \tag{18}$$

Using Equation (17), we have

$$u_{[j]} = \begin{cases} \left(\dfrac{\rho k\omega_j \left(\bar{p}_{[j]} j^{a_{[j]}}\right)^k}{G_{[j]}}\right)^{\frac{1}{k+1}}, & j = 1, 2, \ldots, i \\ \left(\dfrac{\rho k\omega_j \left(\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^k}{G_{[j]}}\right)^{\frac{1}{k+1}}, & j = i+1, \ldots, n \end{cases} \tag{19}$$

Substituting Formula (19) into Formula (18), we can obtain the following expression:

$$(k\rho)^{\frac{1}{k+1}} = \left(\frac{\sum_{j=1}^{i} (\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\bar{p}_{[j]} j^{a_{[j]}}\right)^{\frac{k}{k+1}} + \sum_{j=i+1}^{n} (\omega_j)^{\frac{1}{k+1}} \left(G_{[j]}\lambda_{[j]}\bar{p}_{[j]}(j-i)^{a_{[j]}}\right)^{\frac{k}{k+1}}}{V - W}\right)^{\frac{1}{k}} \tag{20}$$

Inserting Equation (20) into Equation (19), we obtain $u^*_{[j]}$ in Equation (15).

According to Lemma 4.4, substituting Equation (15) into the total weighted resource cost $\sum_{j=1}^{n} G_j u_j$, then we have the following lemma.

Lemma 4.5. *For the problem P2, given that an optimal resource allocation* $u^*_{[j]}$ $(j = 1, 2, \ldots, n)$ *is chosen, then the total weighted resource cost is given by the following expression:*

$$\sum_{j=1}^{n} G_j u_j = (V - W)^{-\frac{1}{k}} (Z)^{\frac{k+1}{k}}, \tag{21}$$

where Z is given by Equation (13).

For a given position i of the RMA, the values V, W and k are all constants. Similar to the problem P1, we obtain that minimizing $\sum_{j=1}^{n} G_j u_j$ is equivalent to minimizing function Z, i.e., minimizing the assignment problem (14). Hence, the problem P2 can be optimally solved by the following algorithm.

Algorithm 4.2

Steps 1-5 are exactly the same as Algorithm 4.1.

Step 6: Calculate the minimum total weighted resource cost $\sum_{j=1}^{n} G_j u_j$ by Equation (21), the optimal resource allocations by Equation (15) and the corresponding actual job processing times.

Theorem 4.2. *Algorithm 4.2 solves optimally the problem P2 in $O(n^4)$ time.*

Proof: The proof is similar to Theorem 4.1.

For the problem P2, the following example illustrates applying Algorithm 4.2 to find the optimal solution.

Example 4.2. *We use the same data as in Example 4.1. Let the limitation on the total schedule penalty cost be* $V = 300$.

From Example 4.1, the global optimal schedule is $\pi^* = \pi(1) = (J_5, J_7, J_1, J_4, J_6, J_2, J_3)$, $Z(1) = 282.2678$. *By Equation (21), the minimum total weighted penalty cost is* $\sum_{j=1}^{n} G_j u_j = 395.1944$. *The corresponding optimal resource allocations are* $u^*_{[j]} = (12.8072, 3.1590, 5.5257, 5.7589, 20.6513, 7.7388, 8.1365)$, *the corresponding actual processing times are* $p_{[j]} = (0.4938, 0.3247, 0.3356, 0.4094, 0.3762, 0.8294, 2.1412)$, *the optimal position of the RMA is* $i^* = 1$ *and the RMA duration is* $f(t) = 4.0988$, *the starting time and the finishing time of the common due-window are* $d^1 = C_{[2]} = C_7 = 4.9173$ *and* $d^2 = C_{[5]} = C_6 = 6.0385$, *respectively, and the due-window size is* $D = 1.1212$.

5. **The Slack Due-Window (SLKW) Assignment Type.** Based on Lemma 3.1(3) and similar to Lemma 4.1, we have the following result.

Lemma 5.1. *For the problems P3 and P4, the total schedule penalty cost can be written as*

$$f = \sum_{j=1}^{i} \varphi_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}} \right)^k + \sum_{j=i+1}^{n} \varphi_j \left(\frac{\lambda_{[j]} \bar{p}_{[j]} (j-i)^{a_{[j]}}}{u_{[j]}} \right)^k + W$$

where W *is given by Equation (2), when* $i < h$,

$$\varphi_j = \begin{cases} \alpha j + \alpha i c + \gamma(n+1) + n\gamma c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, i \\ \alpha j + \gamma(n+1) + \eta + \sigma(n-j+1), & j = i+1, \ldots, h \\ \gamma + n\delta + \eta + \sigma(n-j+1), & j = h+1, \ldots, l \\ \beta(n-j) + \gamma + \eta + \sigma(n-j+1), & j = l+1, \ldots, n \end{cases} \quad (22)$$

when $h \le i < l$,

$$\varphi_j = \begin{cases} \alpha j + \gamma(n+1) + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, h \\ \gamma + n\delta + n\delta c + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = h+1, \ldots, i \\ \gamma + n\delta + \eta + \sigma(n-j+1), & j = i+1, \ldots, l \\ \beta(n-j) + \gamma + \eta + \sigma(n-j+1), & j = l+1, \ldots, n \end{cases} \quad (23)$$

when $l \le i \le n$,

$$\varphi_j = \begin{cases} \alpha j + \beta(n-i)c + \gamma(n+1) + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = 1, 2, \ldots, h \\ \beta(n-i)c + \gamma + n\delta + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = h+1, \ldots, l \\ \beta(n-j) + \beta(n-i)c + \gamma + \eta(1+c) + \sigma(n-j+1) + \sigma(n-i)c, & j = l+1, \ldots, i \\ \beta(n-j) + \gamma + \eta + \sigma(n-j+1), & j = i+1, \ldots, n \end{cases} \quad (24)$$

For problems P3 and P4, the total schedule penalty cost f has the same structure as for problems P1 and P2, except that φ_j is used instead of ω_j. Therefore, Lemma 4.2 and Lemma 4.3 are true for problem P3, Lemma 4.4 and Lemma 4.5 are also true for problem P4, as long as ω_j is replaced by φ_j. Similar to the CONW type, we can use Algorithm 1 and Algorithm 2 to solve problems P3 and P4, respectively, where h and l are calculated according to Lemma 3.1(3), and φ_j is calculated according to Equations (22)-(24). Hence, we obtain the following extended conclusions.

Theorem 5.1. *The problems P3 and P4 can be solved in* $O(n^4)$ *time, respectively.*

6. **The Different Due-Window (DIFW) Assignment Type.** In this section, we consider the problems P5 and P6 with different due-window assignment. Based on Lemma 3.1(4), if $\gamma \geq \beta$, then the total schedule penalty cost $f = \sum_{j=1}^{n} \beta C_j + \eta C_{\max} + \sigma \sum_{j=1}^{n} C_j = (\beta + \sigma) \sum_{j=1}^{n} C_j + \eta C_{\max}$. If $\gamma < \beta$, then the total schedule penalty cost $f = \sum_{j=1}^{n} \gamma C_j + \eta C_{\max} + \sigma \sum_{j=1}^{n} C_j = (\gamma + \sigma) \sum_{j=1}^{n} C_j + \eta C_{\max}$. Hence, for the problems P5 and P6, the total schedule penalty cost can be expressed as follows: $f = \sum_{j=1}^{i} \psi_j \left(\frac{\bar{p}_{[j]} j^{a_{[j]}}}{u_{[j]}} \right)^k + \sum_{j=i+1}^{n} \psi_j \left(\frac{\lambda_{[j]} \bar{p}_{[j]} (j-i)^{a_{[j]}}}{u_{[j]}} \right)^k + W'$, where $W' = (\varepsilon + \sigma)(n - i)b + \eta b$, $\psi_j = \begin{cases} (\varepsilon + \sigma)(n - j + 1) + (\varepsilon + \sigma)(n - i)c + \eta(1 + c), & j = 1, 2, \ldots, i \\ (\varepsilon + \sigma)(n - j + 1) + \eta, & j = i + 1, \ldots, n \end{cases}$ and $\varepsilon = \min\{\beta, \gamma\}$.

Similar to the analysis of the CONW and the SLKW types, using ψ_j instead of ω_j as in Equation (1), W' instead of W as in Equation (2), we obtain the following extensions.

Theorem 6.1. *The problems P5 and P6 can be solved in $O(n^4)$ time, respectively.*

7. **Conclusions.** In this paper, we studied six scheduling problems, where the processing time of the job is affected by the position-dependent aging effects, resource allocation and a deteriorating rate-modifying activity. The objective is to minimize the total schedule penalty cost subject to an upper bound on the total weighted resource cost and to minimize the total weighted resource cost subject to an upper bound on the total schedule penalty cost under three kinds of due-window assignment types. Polynomial-time algorithms were developed for all these presented problems. Moreover, the proposed algorithms can also be used to solve similar problems with the position-dependent learning effects, i.e., $a_j \leq 0$. For future research, it will be interesting to consider other realistic criteria or multiple rate-modifying activities and to design the effective solution algorithms.

Acknowledgment. This work is partially supported by Humanities and Social Science Fund of Ministry of Education of China (No. 18YJA630120), the Young Science and Technology Talents Project Foundation of Liaoning Province Education Department (No. JYT2020128). The author also gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] S. D. Liman, S. S. Panwalkar and S. Thongmee, Common due window size and location determination in a single machine scheduling problem, *Journal of the Operational Research Society*, vol.49, no.9, pp.1007-1010, 1998.

[2] G. Mosheiov and D. Oron, Job dependent due-window assignment based on a common flow allowance, *Foundations of Computing and Decision Sciences*, vol.35, no.3, pp.185-195, 2010.

[3] D. Shabtay and G. Steiner, The single-machine earliness-tardiness scheduling problem with due date assignment and resource-dependent processing times, *Annals of Operations Research*, vol.159, no.1, pp.25-40, 2008.

[4] A. Janiaka, W. A. Janiak, T. Krysiaka and T. Kwiatkowski, A survey on scheduling problems with due windows, *European Journal of Operational Research*, vol.242, no.2, pp.347-357, 2015.

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

[6] J.-B. Wang, Y. Hu and B. Zhang, Common due-window assignment for single-machine scheduling with generalized earliness/tardiness penalties and a rate-modifying activity, *Engineering Optimization*, vol.53, no.3, pp.496-512, 2021.

[7] J.-B. Wang, S.-H. Wang, K. Gao, M. Liu and X. Jia, Due-window assignment methods and scheduling with generalized positional-dependent weights, *Asia-Pacific Journal of Operational Research*, vol.39, no.5, Article ID 2250028, 2022.

[8] G. Mosheiov, Parallel machine scheduling with a learning effect, *Journal of Operational Research Society*, vol.52, no.10, pp.1165-1169, 2001.

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

[10] A. Janiak, T. Krysiak and R. Trela, Scheduling problems with learning and aging effects: A survey, *Decision Making in Manufacturing and Services*, vol.5, nos.1-2, pp.19-36, 2011.

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

[12] A. Janiak, W. Janiak and M. Lichtenstein, Resource management in machine scheduling problems: A survey, *Decision Making in Manufacturing and Services*, vol.1, nos.1-2, pp.59-89, 2007.

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

[14] J.-B. Wang, B. Zhang and H. He, A unified analysis for scheduling problems with variable processing times, *Journal of Industrial and Management Optimization*, vol.18, pp.1063-1077, 2022.

[15] J.-B. Wang, D.-Y. Lv, J. Xu, P. Ji and F. Li, Bicriterion scheduling with truncated learning effects and convex controllable processing times, *International Transactions in Operational Research*, vol.28, no.3, pp.1573-1593, 2021.

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

[17] X. Wang, L. Li and J. Wang, Study on due-window scheduling with controllable processing times and learning effect, *ICIC Express Letters*, vol.10, no.12, pp.2913-2919, 2016.

[18] N. Yin, Single machine due window assignment resource allocation scheduling with job-dependent learning effect, *Journal of Applied Mathematics and Computing*, vol.56, pp.715-725, 2018.

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

[20] L. Liu, J.-J. Wang and X.-Y. Wang, Single machine due-window assignment scheduling with resource-dependent processing times to minimise total resource consumption cost, *International Journal of Production Research*, vol.54, pp.1186-1195, 2016.

[21] 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.

[22] 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.

[23] M. A. Kubzin and V. A. Strusevich, Planning machine maintenance in two-machine shop scheduling, *Operations Research*, vol.54, no.4, pp.789-800, 2006.

[24] X. Jia, D.-Y. Lv, Y. Hu, J.-B. Wang, Z. Wang and E. Wang, Slack due-window assignment scheduling problem with deterioration effects and a deteriorating maintenance activity, *Asia-Pacific Journal of Operational Research*, vol.39, no.6, Article ID 2250005, 2022.

[25] 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.

[26] B. Cheng and L. Cheng, Single machine slack due-window scheduling with linear resource allocation, aging effect, and a deteriorating rate-modifying activity, *International Journal of Applied Mathematics*, vol.30, no.5, pp.375-386, 2017.

[27] G. Mosheiov and A. Sarig, Scheduling a maintenance activity and due-window assignment on a single machine, *Computers & Operations Research*, vol.36, no.9, pp.2541-2545, 2009.

[28] B. Mor and G. Mosheiov, Scheduling a maintenance activity and due-window assignment based on common flow allowance, *International Journal of Production Economics*, vol.135, no.1, pp.222-230, 2012.

[29] Y.-B. Wu, L. Wan and X.-Y. Wang, Study on due-window assignment scheduling based on common flow allowance, *International Journal of Production Economics*, vol.165, pp.155-157, 2015.

[30] A. Seidmann, S. S. Panwalkar and M. L. Smith, Optimal assignment of due dates for a single processor scheduling problem, *International Journal of Production Research*, vol.19, no.4, pp.393-399, 1981.

[31] P. Brucker, *Scheduling Algorithms*, Berlin, Springer, 2007.

[32] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, 1982.

Author Biography

Xiaoli Zhao obtained her Ph.D. degree in System Engineering from Northeastern University, China, 2015.

Dr. Zhao is currently a Lecturer at Shenyang Aerospace University in China. Her research interests include production scheduling and combinational optimization. Her publications include articles in international journals such as European Journal of Operational Research, Asia-Pacific Journal of Operational Research and Mathematical Problems in Engineering.