mathematics

Article

Resource Allocation Scheduling with Position-Dependent Weights and Generalized Earliness–Tardiness Cost

Yi-Chun Wang, Si-Han Wang and Ji-Bo Wang *

School of Science, Shenyang Aerospace University, Shenyang 110136, China
* Correspondence: wangjibo@sau.edu.cn

Abstract: Under just-in-time production, this paper studies a single machine common due-window (denoted by CONW) assignment scheduling problem with position-dependent weights and resource allocations. A job's actual processing time can be determined by the resource assigned to the job. A resource allocation model is divided into linear and convex resource allocations. Under the linear and convex resource allocation models, our goal is to find an optimal due-window location, job sequence and resource allocation. We prove that the weighted sum of scheduling cost (including general earliness–tardiness penalties with positional-dependent weights) and resource consumption cost minimization is polynomially solvable. In addition, under the convex resource allocation, we show that scheduling (resp. resource consumption) cost minimization is solvable in polynomial time subject to the resource consumption (resp. scheduling) cost being bounded.

Keywords: scheduling; assignment problem; resource allocation; positional-dependent weights; earliness–tardiness

MSC: 90B35

check for updates

Citation: Wang, Y.-C.; Wang, S.-H.; Wang, J.-B. Resource Allocation Scheduling with Position-Dependent Weights and Generalized Earliness–Tardiness Cost. *Mathematics* **2023**, *11*, 222. https://doi.org/10.3390/math11010222

Academic Editors: Aldina Correia, Eliana Costa e Silva, Ana Isabel Borges and Shih-Wei Lin

Received: 3 November 2022
Revised: 14 December 2022
Accepted: 30 December 2022
Published: 2 January 2023

1. Introduction

In scheduling models and problems, a due-window assignment has attracted growing attention, particularly in just-in-time (JIT) philosophy (Yin et al. [1]; Janiak et al. [2]; Liu [3]; Liu et al. [4]; Rolim and Nagano [5]; Qian and Zhan [6]). More recently, Wang et al. [7] studied a single machine production scheduling with the due-window assignment. The objective is to minimize the total cost, which is a weighted sum function of earliness, tardiness, due-window starting time, and due-window size, where the weights only depend on their position in a sequence (i.e., position-dependent weights, Wang et al. [8] and Wang et al. [9]). Sun et al. [10] and Qian and Han [11] investigated a proportionate flow shop with the CONW and slack due-window (denoted by SLKW) assignments. The objective is to minimize the generalized weighted sum of numbers of early and late jobs, earliness–tardiness penalties and due-window assignment cost, where the weights are the position-dependent weights. Yue and Zhou [12] studied single machine scheduling with the due-window assignment and stochastic processing times. Wang [13] considered single machine scheduling with past-sequence-dependent setup times. Under common, slack and different due-date assignments, he proved that a general earliness–tardiness cost minimization can be solved in polynomial time. Shabtay et al. [14] explored the single machine scheduling problem with common due-date/due-window assignments. They proved that a non-regular objective cost minimization can be solved in polynomial time under some conditions. They also showed that the problem is NP-hard when the length of the due-window is bounded. Jiang et al. [15] scrutinized the seru scheduling problem with a learning effect. Under multiple due-window assignments, they showed that some cases of the problem are polynomially solvable. Liu et al. [16] considered the single machine due-window assignment scheduling with past-sequence-dependent setup times. Under three due-window assignment models, the objective is to minimize a non-regular cost function (including the weighted sum of earliness–tardiness, number of early and tardy jobs, due-window assignment cost). They proved that this problem is polynomially solvable.

Recently, Wang et al. [17] explored single-machine problems with due-window assignment and positional-dependent weights. Under common and slack due-window assignments, the goal is to minimize the weighted sum of the number of early and late jobs, earliness–tardiness and due-window assignment costs, where the weights are positional-dependent weights. They demonstrated that some versions of common and slack due-window assignments are polynomially solvable. However, in real-world production processes, job processing times are complicated, which are controlled by allocating additional resources (e.g., fuel and gas, et al., see Shabtay and Steiner [18], Liu et al. [19], Lu et al. [20], Wang et al. [21], Zhao [22], Lu et al. [23], and Yan et al. [24]). Hence, in this investigation, we extend the results of Wang et al. [17] by considering resource allocation that includes the one given in Wang et al. [17] as a special case. We summarize the contributions as follows: (1) We investigate single machine scheduling with resource allocation and CONW due-window assignment; (2) Under linear resource allocation, we provide an analysis for the weighted sum of scheduling cost (including generalized earliness–tardiness penalties with positional-dependent weights) and resource consumption costs; (3) Under convex resource allocation, we give a bicriteria analysis for scheduling costs and resource consumption costs; (4) We derive the structural properties of an optimal solution and demonstrate that the problem remains polynomially solvable. This article is organized as follows. Section 2 presents a description of the problem. In Section 3, we first give two lemmas, then we prove that four problems are polynomially solvable. In Section 4, a case study is given. In Section 5, conclusions are presented.

2. Problem Formulation

There are n independent non-preemptive jobs $J = \{J_1, J_2, \ldots, J_n\}$ that are to be processed on a single-machine, and all jobs are available at time zero. The machine can only process one job at a time. Let p_j^A be the actual processing time of job J_j; under a linear resource allocation, we have

$$p_j^A = \bar{p}_j - b_j u_j,$$

where $\bar{p}_j$ (resp. b_j) is the normal processing time (resp. compression rate) of job J_j, u_j is the resource amount of job J_j, $0 \le u_j \le \bar{u}_j < \frac{\bar{p}_j}{b_j}$ and $\bar{u}_j$ is the upper bound of u_j. For a convex resource allocation, we have

$$p_j^A = \left(\frac{\bar{p}_j}{u_j}\right)^k,$$

where $k > 0$ is a constant. In this paper, all the jobs are subject to a common due-window $[d_1, d_2]$ ($d_1 \le d_2$), where d_1 is the due-window starting time and d_2 is the due-window finishing (completion) time; then, the due-window size is $D = d_2 - d_1$. Let C_j be the completion time of job J_j, the number of early (U_j) and tardy jobs (V_j) that are given as:

$$U_j = \begin{cases} 1, & \text{if } d_1 > C_j \\ 0, & \text{otherwise} \end{cases}$$

and

$$V_j = \begin{cases} 1, & \text{if } d_2 < C_j \\ 0, & \text{otherwise.} \end{cases}$$

Let $E_j = \max\{d_1 - C_j, 0\}$ (resp. $T_j = \max\{C_j - d_2, 0\}$) be the earliness (tardiness) of job J_j ($j = 1, 2, \ldots, n$). Let $[j]$ be the job that is placed in the jth position; the goal is to determine the optimal job sequence π, d_1 and D (such as d_2) that minimizes the total cost. Formally, the resource consumption (resp. scheduling) cost is $\sum_{j=1}^{n} v_j u_j$ ($\sum_{j=1}^{n}(\alpha_j U_{[j]} + \beta_j V_{[j]} + \eta_j E_{[j]} + \delta_j T_{[j]} + \theta d_1 + \lambda D)$). For the linear (resp. convex) resource allocation, the first (second) problem (denoted by P_1 and P_2 respectively) is to

$$Z = \sum_{j=1}^{n}\left(\alpha_j U_{[j]} + \beta_j V_{[j]} + \eta_j E_{[j]} + \delta_j T_{[j]} + \theta d_1 + \lambda D\right) + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]}, \tag{1}$$

where α_j, β_j, η_j and δ_j are positional-dependent weights (penalty factors) for the one-time penalties for earliness, tardiness and the unit time of earliness, tardiness, respectively; θ and λ are the unit time weight (penalty) for the due-window starting time d_1 and size D, ρ is the weight of resource costs and v_j is the unit cost of processing job J_j. For the convex resource allocation, the third (resp. fourth) problem (denoted by P_3 and P_4, respectively) is to minimize scheduling cost $\sum_{j=1}^{n}\left(\alpha_j U_{[j]} + \beta_j V_{[j]} + \eta_j E_{[j]} + \delta_j T_{[j]} + \theta d_1 + \lambda D\right)$ (resource cost $\sum_{j=1}^{n} v_j u_j$) subject to $\sum_{j=1}^{n} v_j u_j \le \widehat{U}$ ($\sum_{j=1}^{n}\left(\alpha_j U_{[j]} + \beta_j V_{[j]} + \eta_j E_{[j]} + \delta_j T_{[j]} + \theta d_1 + \lambda D\right) \le \widehat{V}$), where $\widehat{U}$ ($\widehat{V}$) is a given constant.

Remark 1 (Liu et al. [16])**.** *considered the single machine scheduling with past-sequence-dependent setup times, i.e., for the common due-window assignment; the objective function is to minimize* $\sum_{j=1}^{n}\left(\tilde{\alpha}_j U_j + \tilde{\beta}_j V_j + \tilde{\eta} E_{[j]} + \tilde{\delta} T_{[j]} + \theta d_1 + \lambda D\right)$*, where* $\tilde{\alpha}_j$ ($\tilde{\beta}_j$) *is the weight of job* J_j *(i.e., job-dependent weight), and* $\tilde{\eta}$ *and* $\tilde{\delta}$ *are given constants. They also addressed the slack and unrestricted due-window assignments.*

3. Method

From Wang et al. [17], for the problems P_1–P_4, there exists an optimal sequence π such that all jobs are processed at time 0, and there is no idle time during consecutive processing.

Lemma 1. *For the problems* P_1*-*P_4 *, there exists an optimal sequence* π *such that*
(1) The optimal d_1 *is equal to* $C_{[m]}$ *(i.e.,* $d_1 = C_{[m]}$*) and the optimal* d_2 *is equal to* $C_{[w]}$ *(i.e.,* $d_2 = C_{[w]}$*), where* $1 \le m \le w \le n$*;*
(2) $d_1 = C_{[m]}$*, where* $\sum_{j=1}^{m-1} \eta_j \le n(\lambda - \theta)$*,* $d_2 = C_{[w]}$*, where* $\sum_{j=w+1}^{n} \eta_j \le n\lambda$*.*

Proof. Similar to the proof of Wang et al. [17]. □

3.1. Problem P_1

From Lemma 1, for any given sequence, d_1, d_2 and D can be calculated as follows: $d_1 = C_{[m]} = \sum_{l=1}^{m} p_{[l]}^A$, $d_2 = C_{[w]} = \sum_{l=1}^{n} p_{[l]}^A$, $D = C_{[w]} - C_{[m]} = \sum_{l=m+1}^{w} p_{[l]}^A$. Hence, it follows that

$$\begin{aligned}
Z &= \sum_{j=1}^{n}\left(\alpha_j U_{[j]} + \beta_j V_{[j]} + \eta_j E_{[j]} + \delta_j T_{[j]} + \theta d_1 + \lambda D\right) + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{m-1} \eta_j E_{[j]} + \sum_{j=w+1}^{n} \delta_j T_{[j]} + n\theta d_1 + n\lambda D + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{m-1} \eta_j \left(d_1 - C_{[j]}\right) + \sum_{j=w+1}^{n} \delta_j \left(C_{[j]} - d_2\right) + n\theta d_1 + n\lambda D \\
&\quad + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{m-1} \eta_j \left(\sum_{l=1}^{m} p_{[l]}^A - \sum_{l=1}^{j} p_{[l]}^A\right) + \sum_{j=w+1}^{n} \delta_j \left(\sum_{l=1}^{j} p_{[l]}^A - \sum_{l=1}^{w} p_{[l]}^A\right) \\
&\quad + n\theta \sum_{l=1}^{m} p_{[l]}^A + n\lambda \sum_{l=m+1}^{w} p_{[l]}^A + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]} \\
&= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j p_{[j]}^A + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]},
\end{aligned} \tag{2}$$

where

$$\tilde{\Phi}_j = \begin{cases} \sum_{l=1}^{j-1} \eta_l + n\theta, & j = 1, 2, \ldots, m, \\ n\lambda, & j = m+1, m+2, \ldots, w \\ \sum_{l=j}^{n} \delta_l, & j = w+1, w+2, \ldots, n. \end{cases} \tag{3}$$

When $p_j^A = \bar{p}_j - b_j u_j$, we have

$$\begin{aligned} Z &= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j (\bar{p}_{[j]} - b_{[j]} u_{[j]}) + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]} \\ &= \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j \bar{p}_{[j]} + \sum_{j=1}^{n} \left(\rho v_{[j]} - b_{[j]} \tilde{\Phi}_j \right) u_{[j]} \end{aligned} \tag{4}$$

Let $u_{[j]}^* (1 \le j \le n)$ be the optimal resource allocation of the jth position; we have

$$u_{[j]}^* = \begin{cases} \bar{u}_{[j]}, & \rho v_{[j]} - b_{[j]} \tilde{\Phi}_j < 0 \\ u_{[j]} \in \left[0, \bar{u}_{[j]}\right], & \rho v_{[j]} - b_{[j]} \tilde{\Phi}_j = 0 \\ 0, & \rho v_{[j]} - b_{[j]} \tilde{\Phi}_j > 0 \end{cases} \tag{5}$$

From (2) and (3), if m and w are given, let $x_{jr} = 1$; if job J_j is placed in the rth position and $x_{jr} = 0$ otherwise, then the optimal job sequence π of the problem P_1 can be formulated as the following assignment problem:

$$\min \ Z = \sum_{j=1}^{n} \sum_{r=1}^{n} \tau_{jr} x_{jr} \tag{6}$$

$$s.t. \begin{cases} \sum_{j=1}^{n} x_{jr} = 1, & r = 1, 2, \ldots, n, \\ \sum_{r=1}^{n} x_{jr} = 1, & j = 1, 2, \ldots, n, \\ x_{jr} = 0 \ \text{ or } \ 1, \end{cases} \tag{7}$$

in which

$$\tau_{jr} = \begin{cases} \varpi_r + \chi_r \bar{p}_{[j]}, & \rho v_{[j]} - b_{[j]} \chi_r \ge 0 \\ \varpi_r + \chi_r \bar{p}_{[j]} + \left(\rho v_{[j]} - b_{[j]} \chi_r \right) \bar{u}_{[j]}, & \rho v_{[j]} - b_{[j]} \chi_r < 0 \end{cases} \tag{8}$$

$$\chi_r = \begin{cases} \sum_{l=1}^{r-1} \eta_l + n\theta, & r = 1, 2, \ldots, m \\ n\lambda, & r = m+1, m+2, \ldots, w \\ \sum_{l=r}^{n} \delta_l, & r = w+1, w+2, \ldots, n \end{cases} \tag{9}$$

and

$$\varpi_r = \begin{cases} \alpha_r, & r = 1, 2, \ldots, m-1 \\ 0, & r = m, m+1, \ldots, w \\ \beta_r, & r = w+1, w+2, \ldots, n. \end{cases} \tag{10}$$

Based on the above analysis, the problem P_1 can be solved by following Algorithm 1:

Algorithm 1: The problem P_1.

Input: $\bar{p}_j, b_j, \bar{u}_j, v_j, \alpha_j, \beta_j, \eta_j, \delta_j$ $(j = 1, 2, \ldots, n), n, \theta, \lambda, \rho$
Output: The optimal sequence π^*, Z^*, u_j^*, ${d_1}^*$, D^*
Step 1. From Lemma 1, calculate the range of m and w.
Step 2. For each pair of m and w ($m = 1, 2, \ldots, n$, $w = 1, 2, \ldots, n$, $m \le w$), calculate τ_{jr} (see (8)–(10)), to solve the assignment problems (6) and (7).
Step 3. For each pair of m and w, a suboptimal sequence $\pi(m, w)$ and value $Z(m, w)$ can be obtained.
Step 4. The global optimal sequence π^* is the one with the minimum value $Z^* = \min\{Z(m, w)$, where $1 \le m \le w \le n\}$.
Step 5. Calculate u_j^* by using Equation (5).
Step 6. Calculate $d_1^* = C_{[m]}$ and $D^* = C_{[w]} - C_{[m]}$.

Theorem 1. *The problem P_1 can be solved by Algorithm 1 in $O(n^5)$ time.*

Proof. The correctness of Algorithm 1 follows the above analysis. Steps 1 and 3–6 need $O(n)$ time; in Step 2, for each pair of m and w, solving the assignment problem requires $O(n^3)$ time; m and w are less than n. Hence, the time complexity of Algorithm 1 is $O(n^5)$. □

3.2. Problem P_2

Similar to Section 3.1, for the problem P_2, under $p_j^A = \left(\frac{\bar{p}_j}{u_j}\right)^k$, we have

$$Z = \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j \left(\frac{\bar{p}_{[j]}}{u_{[j]}}\right)^k + \rho \sum_{j=1}^{n} v_{[j]} u_{[j]}. \tag{11}$$

Lemma 2. *The optimal resource allocation of the problem P_2 is:*

$$u_{[j]}^* = \frac{\left(k\tilde{\Phi}_j\right)^{\frac{1}{k+1}} \left(\bar{p}_{[j]}\right)^{\frac{k}{k+1}}}{\left(\rho v_{[j]}\right)^{\frac{k}{k+1}}}, j = 1, 2, \ldots n. \tag{12}$$

Proof. Taking a partial derivative of (11) with respect to $u_{[j]}$ and making it equal to 0, we have

$$\rho v_{[j]} - k\tilde{\Phi}_j \frac{\left(\bar{p}_{[j]}\right)^k}{\left(u_{[j]}\right)^{k+1}} = 0, \tag{13}$$

and the result (12) can be obtained. □

Substituting Equation (12) into Equation (11), we have

$$Z = \left(k^{-\frac{k}{k+1}} + k^{\frac{1}{k+1}}\right) \rho^{\frac{k}{k+1}} \sum_{j=1}^{n} \left(v_{[j]} \bar{p}_{[j]}\right)^{\frac{k}{k+1}} \tilde{\Phi}_j^{\frac{1}{k+1}} + \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j. \tag{14}$$

Similarly, from Equation (14), if m and w are given, then the optimal job sequence π of the problem P_2 can be formulated as the following assignment problem:

$$\min\ Z = \sum_{j=1}^{n} \sum_{r=1}^{n} \tau_{jr} x_{jr} \tag{15}$$

$$s.t. \begin{cases} \sum_{r=1}^{n} x_{jr} = 1, & j = 1, 2, \ldots, n, \\ \sum_{j=1}^{n} x_{jr} = 1, & r = 1, 2, \ldots, n, \\ x_{jr} = 0 \text{ or } 1, \end{cases} \tag{16}$$

in which

$$\tau_{jr} = \begin{cases} \left(k^{-\frac{k}{k+1}} + k^{\frac{1}{k+1}}\right) \rho^{\frac{k}{k+1}} \chi_r^{\frac{1}{k+1}} \left(v_j \bar{p}_j\right)^{\frac{k}{k+1}} + \alpha_r, & r = 1, 2 \ldots m-1 \\ \left(k^{-\frac{k}{k+1}} + k^{\frac{1}{k+1}}\right) \rho^{\frac{k}{k+1}} \chi_r^{\frac{1}{k+1}} \left(v_j \bar{p}_j\right)^{\frac{k}{k+1}}, & r = m, m+1, \ldots, w, \\ \left(k^{-\frac{k}{k+1}} + k^{\frac{1}{k+1}}\right) \rho^{\frac{k}{k+1}} \chi_r^{\frac{1}{k+1}} \left(v_j \bar{p}_j\right)^{\frac{k}{k+1}} + \beta_r, & r = w+1, w+2, \ldots, n, \end{cases} \tag{17}$$

and χ_r is given by Equation (10).

Based on the above analysis, the problem P_2 can be solved by following Algorithm 2:

Algorithm 2: The problem P_2.

Input: $\bar{p}_j, v_j, \alpha_j, \beta_j, \eta_j, \delta_j$ $(j = 1, 2, \ldots, n)$, $n, k, \theta, \lambda, \rho$
Output: The optimal sequence π^*, Z^*, u_j^*, ${d_1}^*$, D^*
Step 1. From Lemma 1, calculate the range of m and w.
Step 2. For each pair of m and w $(m = 1, 2, \ldots, n, w = 1, 2, \ldots, n, m \le w)$, calculate τ_{jr} (see (17)), to solve the assignment problems (15) and (16).
Step 3. For each pair of m and w, a suboptimal sequence $\pi(m, w)$ and value $Z(m, w)$ (see (15) and (16)) can be obtained.
Step 4. The global optimal sequence π^* is the one with the minimum value $Z^* = \min\{Z(m, w)$, where $1 \le m \le w \le n\}$.
Step 5. Calculate u_j^* by using Equation (12).
Step 6. Calculate $d_1^* = C_{[m]}$ and $D^* = C_{[w]} - C_{[m]}$.

Theorem 2. *The problem P_2 can be solved by Algorithm 2 in $O(n^5)$ time.*

3.3. Problem P_3

In this subsection, we consider the problem P_3, i.e.,

$$\min Z = \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j p_{[j]}^A = \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j \left(\frac{\bar{p}_{[j]}}{u_{[j]}} \right)^k \tag{18}$$

subject to $\sum_{j=1}^{n} v_{[j]} u_{[j]} \le \widehat{U}$.

Lemma 3. *The optimal resource allocation of the problem P_3 is:*

$$u_{[j]}^* = \frac{\left(\tilde{\Phi}_j \left(\bar{p}_{[j]} \right)^k \right)^{\frac{1}{k+1}}}{\left(v_{[j]} \right)^{\frac{1}{k+1}} \sum_{j=1}^{n} \left(v_{[j]} \bar{p}_{[j]} \right)^{\frac{k}{k+1}} \left(\tilde{\Phi}_j \right)^{\frac{1}{k+1}}} \times \widehat{U}, j = 1, 2, \ldots, n \tag{19}$$

Proof. For any given job sequence, by the Lagrange multiplier method, we have

$$L\left(u_{[j]}, \mu\right) = \sum_{j=1}^{m-1} \alpha_j + \sum_{j=w+1}^{n} \beta_j + \sum_{j=1}^{n} \tilde{\Phi}_j \left(\frac{\bar{p}_j}{u_j} \right)^k + \mu \left(\sum_{j=1}^{n} v_{[j]} u_{[j]} - \widehat{U} \right), \tag{20}$$

where μ is the Lagrangian multiplier, $\mu \ge 0$. Taking the partial derivative of $u_{[j]}$ and μ, we have

$$\frac{\partial L(u, \mu)}{\partial u_{[j]}} = \mu v_{[j]} - k \tilde{\Phi}_j \frac{\left(\bar{p}_{[j]} \right)^k}{\left(u_{[j]} \right)^{k+1}} = 0 \tag{21}$$

$$u_{[j]} = \frac{\left(k \tilde{\Phi}_j \left(\bar{p}_{[j]} \right)^k \right)^{1/k+1}}{\left(\mu v_{[j]} \right)^{1/k+1}} \tag{22}$$

$$\frac{\partial L(u, \mu)}{\partial \mu} = \sum_{j=1}^{n} v_{[j]} u_{[j]} - \widehat{U} = 0 \tag{23}$$

$$\mu^{1/k+1} = \frac{\sum_{j=1}^{n} \left(k \tilde{\Phi}_j \right)^{1/k+1} \left(v_{[j]} \bar{p}_{[j]} \right)^{k/k+1}}{\widehat{U}}. \tag{24}$$

Substituting Equation (24) into Equation (22), Equation (19) can be obtained. □

Putting $u^*_{[j]}$ (i.e., Equation (19)) in Equation (18), we have

$$Z = \frac{\left(\sum_{j=1}^{n}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\tilde{\Phi}_j^{\frac{1}{k+1}}\right)^{k+1}}{\widehat{U}^k} + \sum_{j=1}^{m-1}\alpha_j + \sum_{j=w+1}^{n}\beta_j. \quad (25)$$

If m and w are given, then $\sum_{j=1}^{m-1}\alpha_j + \sum_{j=w+1}^{n}\beta_j$ is a constant number. Hence, it can be obtained that solving problem P_3 is equal to minimizing $\sum_{j=1}^{n}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\tilde{\Phi}_j^{\frac{1}{k+1}}$ and this term can be minimized by the HLP rule (see Hardy et al. [25]).

Based on the above analysis, the problem P_3 can be solved by following Algorithm 3:

Algorithm 3: The problem P_3.

Input: $\bar{p}_j, v_j, \alpha_j, \beta_j, \eta_j, \delta_j$ $(j = 1, 2, \ldots, n)$, $n, k, \theta, \lambda, \widehat{U}$
Output: The optimal sequence π^*, Z^*, u_j^*, ${d_1}^*$, D^*
Step 1. From Lemma 1, calculate the range of m and w.
Step 2. For each pair of m and w, a suboptimal sequence $\pi(m,w)$ and value $Z(m,w)$ (see Equation (25)) can be obtained by the HLP rule, i.e., by matching the largest $(v_j\bar{p}_j)^{\frac{k}{k+1}}$ with the smallest $\tilde{\Phi}_j^{\frac{1}{k+1}}$, the second largest $(v_j\bar{p}_j)^{\frac{k}{k+1}}$ matches the second smallest $\tilde{\Phi}_j^{\frac{1}{k+1}}$, and so on.
Step 3. The global optimal sequence π^* is the one with the minimum value $Z^* = \min\{Z(m,w), \text{where } 1 \le m \le w \le n\}$.
Step 4. Calculate u_j^* by using Equation (19).
Step 5. Calculate $d_1^* = C_{[m]}$ and $D^* = C_{[w]} - C_{[m]}$.

Theorem 3. *The problem P_3 can be solved by Algorithm 3 in $O(n^5)$ time.*

3.4. Problem P_4

Similar to Section 3.3, we have:

Lemma 4. *The optimal resource allocation of the problem P_4 is:*

$$u^*_{[j]} = \frac{\tilde{\Phi}_j^{\frac{1}{k+1}}\left(\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\left(\sum_{j=1}^{n}(\tilde{\Phi}_j)^{\frac{1}{k+1}}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1}{k}}}{\left(v_{[j]}\right)^{\frac{1}{k+1}}\left(\widehat{V} - \sum_{j=1}^{m-1}\alpha_j - \sum_{j=w+1}^{n}\beta_j\right)^{\frac{1}{k}}}, j = 1, 2, \ldots, n \quad (26)$$

Proof. For any given job sequence, we have:

$$L\left(u_{[j]}, \mu\right) = \sum_{j=1}^{n} v_{[j]}u_{[j]} + \mu\left(\sum_{j=1}^{m-1}\alpha_j + \sum_{j=w+1}^{n}\beta_j + \sum_{j=1}^{n}\tilde{\Phi}_j\left(\frac{\bar{p}_{[j]}}{u_{[j]}}\right)^k - \widehat{V}\right), \quad (27)$$

where μ is the Lagrangian multiplier, $\mu \ge 0$, we have:

$$\frac{\partial L(u_{[j]}, \mu)}{\partial u_{[j]}} = v_{[j]} - \mu k\tilde{\Phi}_j\frac{\left(\bar{p}_{[j]}\right)^k}{\left(u_{[j]}\right)^{k+1}} = 0 \quad (28)$$

$$u_{[j]} = \frac{\left(\mu k\tilde{\Phi}_j\left(\bar{p}_{[j]}\right)^k\right)^{1/k+1}}{\left(v_{[j]}\right)^{\frac{1}{k+1}}} \quad (29)$$

$$\frac{\partial L(u_{[j]},\mu)}{\partial \mu}=\sum_{j=1}^{m-1}\alpha_j+\sum_{j=w+1}^{n}\beta_j+\sum_{j=1}^{n}\tilde{\Phi}_j\left(\frac{\bar{p}_{[j]}}{u_{[j]}}\right)^k-\widehat{V}=0 \quad (30)$$

$$(k\mu)^{\frac{1}{k+1}}=\frac{\left(\sum_{j=1}^{n}(\tilde{\Phi}_j)^{\frac{1}{k+1}}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1}{k}}}{\left(\widehat{V}-\sum_{j=1}^{m-1}\alpha_j-\sum_{j=w+1}^{n}\beta_j\right)^{\frac{1}{k}}}. \quad (31)$$

Substituting Equation (31) into Equation (29), Equation (26) can be obtained. □

Substituting Equation (26) into $\sum_{j=1}^{n} v_j u_j$, we have:

$$\begin{aligned} Z &= \sum_{j=1}^{n} v_{[j]}u_{[j]} \\ &= \sum_{j=1}^{n} v_{[j]}\left[\frac{\tilde{\Phi}_j^{\frac{1}{k+1}}\left(\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\left(\sum_{j=1}^{n}(\tilde{\Phi}_j)^{\frac{1}{k+1}}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1}{k}}}{\left(v_{[j]}\right)^{\frac{1}{k+1}}\left(V-\sum_{j=1}^{m-1}\alpha_j-\sum_{j=w+1}^{n}\beta_j\right)^{\frac{1}{k}}}\right] \\ &= \frac{\left(\sum_{j=1}^{n}(\tilde{\Phi}_j)^{\frac{1}{k+1}}\left(v_{[j]}\bar{p}_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1+k}{k}}}{\left(\widehat{V}-\sum_{j=1}^{m-1}\alpha_j-\sum_{j=w+1}^{n}\beta_j\right)^{\frac{1}{k}}}. \end{aligned} \quad (32)$$

Similar to Section 3.3, the problem P_4 can be solved by following Algorithm 4:

Algorithm 4: The problem P_4.

Input: $\bar{p}_j, v_j, \alpha_j, \beta_j, \eta_j, \delta_j$ $(j=1,2,\ldots,n), n, k, \theta, \lambda, \widehat{V}$
Output: The optimal sequence $\pi^*, Z^*, u_j^*, {d_1}^*, D^*$
Step 1. From Lemma 1, calculate the range of m and w.
Step 2. For each pair of m and w, a suboptimal sequence $\pi(m,w)$ and value $Z(m,w)$ (see Equation (32)) can be obtained by the HLP rule, i.e., by matching the largest $(v_j\bar{p}_j)^{\frac{k}{k+1}}$ with the smallest $\tilde{\Phi}_j^{\frac{1}{k+1}}$, the second largest $(v_j\bar{p}_j)^{\frac{k}{k+1}}$ matches the second smallest $\tilde{\Phi}_j^{\frac{1}{k+1}}$, and so on.
Step 3. The global optimal sequence π^* is the one with the minimum value $Z^*=\min\{Z(m,w), \text{where } 1\le m\le w\le n\}$.
Step 4. Calculate u_j^* by using Equation (26).
Step 5. Calculate $d_1^*=C_{[m]}$ and $D^*=C_{[w]}-C_{[m]}$.

Theorem 4. *The problem P_4 can be solved by Algorithm 4 in $O(n^5)$ time.*

4. A Case Study

In this section, we present a case study to illustrate how the proposed algorithms works.

Example 1. *Consider a five-job problem, where $k=2, \theta=3, \lambda=4, \rho=6$, $\widehat{U}=100, \widehat{V}=1000$; the parameters of job J_j $(j=1,2,\ldots,5)$ are given in Table 1. The parameters of position-dependent weights (i.e., $\alpha_j, \beta_j, \eta_j$ and δ_j, where $j=1,2,\ldots,5$) are given in Table 2.*

Table 1. Values of job-dependent parameters.

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$\bar{p}_j$	13	12	14	15	17
b_j	2	2	3	1	1
$\bar{u}_j$	5	4	3	11	7
v_j	3	5	4	1	6

Table 2. Values of position-dependent weights.

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
α_j	4	8	7	6	5
β_j	8	7	3	2	6
η_j	2	4	6	10	7
δ_j	2	5	4	3	6

For the problem P_1, from Algorithm 1, $m = 1,2$, $w = 3,4,5$, and all results are shown in Table 3. From Table 3, the optimal sequence $\pi^*(2,3)$ is $J_3 \to J_4 \to J_1 \to J_2 \to J_5$, $u_3^* = 3$, $u_4^* = 11$, $u_1^* = 5$, $u_2^* = 4$, $u_5^* = 0$, $Z^*(2,3) = 653$, $d_1^* = C_{[2]} = 9$, and $D^* = C_{[3]} - C_{[2]} = 3$. When $m = 2$ and $w = 3$, the values $\tilde{\Phi}_1 = 15$, $\tilde{\Phi}_2 = 17$, $\tilde{\Phi}_3 = 20$, $\tilde{\Phi}_4 = 9$, $\tilde{\Phi}_5 = 6$ (see Equation (3)), τ_{jr} (see Equation (8)) are given in Table 4, and the optimal solution of the assignment problem is given by Table 4.

Table 3. Results of problem P_1 (bold numbers are the optimal solution).

m	w	$\pi(m,w)$	$Z(m,w)$
1	3	$(J_3, J_4, J_1, J_2, J_5)$	661
1	4	$(J_3, J_2, J_1, J_4, J_5)$	751
1	5	$(J_5, J_1, J_2, J_3, J_4)$	923
2	**3**	$\boldsymbol{(J_3, J_4, J_1, J_2, J_5)}$	**653**
2	4	$(J_3, J_4, J_2, J_1, J_5)$	743
2	5	$(J_5, J_3, J_1, J_4, J_2)$	912

Table 4. Values τ_{jr} for $m = 2$ and $w = 3$ (bold numbers are the optimal solution).

	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
J_1	139	141	**150**	119	84
J_2	184	188	200	**110**	78
J_3	**151**	157	172	119	90
J_4	130	**134**	146	104	96
J_5	259	289	340	155	**108**

Similarly, for the problem P_2 from Algorithm 2, the results are shown in Table 5. From Table 5, the optimal sequence $\pi^*(2,3)$ is $J_3 \to J_1 \to J_4 \to J_2 \to J_5$, $u_3^* = 2.169$, $u_1^* = 2.608$, $u_4^* = 6.230$, $u_2^* = 1.423$, $u_5^* = 1.388$, $Z^*(2,3) = 971.297$, $d_1^* = C_{[2]} = 66.505$ and $D^* = C_{[3]} - C_{[2]} = 5.797$.

Table 5. Results of problem P_2 (bold numbers are the optimal solution).

m	w	$\pi(m,w)$	$Z(m,w)$
1	3	$(J_3, J_4, J_1, J_2, J_5)$	977.570
1	4	$(J_2, J_4, J_1, J_3, J_5)$	1035.169
1	5	$(J_5, J_1, J_3, J_4, J_2)$	1141.332
2	**3**	$\boldsymbol{(J_3, J_1, J_4, J_2, J_5)}$	**971.297**
2	4	$(J_2, J_3, J_1, J_4, J_5)$	1026.095
2	5	$(J_5, J_2, J_1, J_4, J_3)$	1131.643

Similarly, for problem P_3, from Algorithm 3, the results are shown in Table 6. From Table 6, the optimal sequence $\pi^*(2,3)$ is $J_3 \to J_1 \to J_4 \to J_2 \to J_5$, $u_3^* = 5.871$, $u_1^* = 6.412$, $u_4^* = 10.740$, $u_2^* = 4.148$, $u_5^* = 4.301$, $Z^*(2,3) = 375.290$, $d_1^* = C_{[2]} = 9.798$, and $D^* = C_{[3]} - C_{[2]} = 1.951$.

Table 6. Results of problem P_3 (bold numbers are the optimal solution).

m	w	$\pi(m,w)$	$Z(m,w)$
1	3	$(J_3, J_4, J_1, J_2, J_5)$	383.086
1	4	$(J_2, J_4, J_1, J_3, J_5)$	454.595
1	5	$(J_5, J_4, J_1, J_3, J_2)$	611.830
2	**3**	$\mathbf{(J_3, J_1, J_4, J_2, J_5)}$	**375.290**
2	4	$(J_2, J_3, J_4, J_1, J_5)$	441.715
2	5	$(J_5, J_2, J_4, J_1, J_3)$	594.076

Similarly, for problem P_4, from Algorithm 4, the results are shown in Table 7. From Table 7, the optimal sequence is $\pi^*(2,3) = J_3 \to J_1 \to J_4 \to J_2 \to J_5$, $u_3^* = 3.563$, $u_1^* = 3.892$, $u_4^* = 6.519$, $u_2^* = 2.518$, $u_5^* = 2.611$, $Z^*(2,3) = 60.516$, $d_1^* = C_{[2]} = 26.596$, and $D^* = C_{[3]} - C_{[2]} = 5.295$.

Table 7. Results of problem P_4 (bold numbers are the optimal solution).

m	w	$\pi(m,w)$	$Z(m,w)$
1	3	$(J_3, J_4, J_1, J_2, J_5)$	61.491
1	4	$(J_2, J_4, J_1, J_3, J_5)$	67.180
1	5	$(J_5, J_4, J_1, J_3, J_2)$	78.220
2	**3**	$\mathbf{(J_3, J_1, J_4, J_2, J_5)}$	**60.516**
2	4	$(J_2, J_3, J_4, J_1, J_5)$	66.036
2	5	$(J_5, J_2, J_4, J_1, J_3)$	76.971

5. Conclusions

This article explored resource allocation scheduling with general position-dependent weights and common due-window assignment. The resource allocation can be divided into linear and convex resource allocations. Under some combination of general earliness–tardiness cost and resource consumption cost, we showed that four versions of the problem are polynomially solvable. Future research and investigation will consider the scheduling under flow shop settings, or extending our model from resource allocation to general deterioration effects.

Author Contributions: Conceptualization, Y.-C.W., S.-H.W. and J.-B.W.; methodology, Y.-C.W. and S.-H.W.; investigation, J.-B.W.; writing, review and editing, Y.-C.W., S.-H.W. and J.-B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This Work was supported by LiaoNing Revitalization Talents Program (XLYC2002017).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yin, Y.; Cheng, T.C.E.; Wu, C.-C.; Cheng, S.-R. Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time. *J. Oper. Res. Soc.* **2013**, *65*, 1–13. [CrossRef]
2. Janiak, A.; Janiak, W.A.; Krysia, T.; Kwiatkowski, T. A survey on scheduling problems with due windows. *Eur. J. Oper. Res.* **2015**, *242*, 347–357. [CrossRef]
3. Common due-window assignment and group scheduling with position-dependent processing times. *Asia-Pac. J. Oper. Res.* **2015**, *32*, 1550045. [CrossRef]
4. Liu, W.; Hu, X.; Wang, X. Single machine scheduling with slack due dates assignment. *Eng. Optim.* **2017**, *49*, 709–717. [CrossRef]

5. Rolim, G.A.; Nagano, M.S. Structural properties and algorithms for earliness and tardiness scheduling against common due dates and windows: A review. *Comput. Ind. Eng.* **2020**, *149*, 106803. [CrossRef]
6. Qian, J.; Zhan, Y. The due date assignment scheduling problem with delivery times and truncated sum-of-processing-times-based learning effect. *Mathematics* **2021**, *9*, 3085. [CrossRef]
7. Wang, J.-B.; Zhang, B.; Li, L.; Bai, D.; Feng, Y.-B. Due window assignment scheduling problems with position-dependent weights on a single machine. *Eng. Optim.* **2020**, *52*, 185–193. [CrossRef]
8. Wang, J.-B.; Cui, B.; Ji, P.; Liu, W.-W. Research on scheduling with position-dependent weights and past-sequence-dependent delivery times. *J. Comb. Optim.* **2021** *41*, 290–303. [CrossRef]
9. Wang, S.-H.; Lv, D.-Y.; Wang, J.-B. Research on position-dependent weights scheduling with delivery times and truncated sum-of-processing-times-based learning effect. *J. Ind. Manag. Optim.* **2023**, *19*, 2824–2837. [CrossRef]
10. Sun, X.-Y.; Geng, X.-N.; Liu, T. Due-window assignment scheduling in the proportionate flow shop setting. *Ann. Oper. Res.* **2020**, *292*, 113–131. [CrossRef]
11. Qian, J.; Han, H. Improved algorithms for proportionate flow shop scheduling with due-window assignment. *Ann. Oper. Res.* **2022**, *309*, 249–258. [CrossRef]
12. Yue, Q.; Zhou, S. Due-window assignment scheduling problem with stochastic processing times. *Eur. J. Oper. Res.* **2021**, *290*, 453–468. [CrossRef]
13. Wang, W. Single-machine due-date assignment scheduling with generalized earliness/tardiness penalties including proportional setup times. *J. Appl. Math. Comput.* **2022**, *68*, 1013–1031. [CrossRef]
14. Shabtay, D.; Mosheiov, G.; Oron, D. Single machine scheduling with common assignable due date/due window to minimize total weighted early and late work. *Eur. J. Oper. Res.* **2022**, *303*, 66–77. [CrossRef]
15. Jiang, Y.; Zhang, Z.; Song, X.; Yin, Y. Seru scheduling problems with multiple due-windows assignment and learning effect. *J. Syst. Sci. Syst. Eng.* **2022**, *31*, 480–511. [CrossRef]
16. Liu, W.; Wang, X.; Wang, X.; Zhao, P. Due-window assignment scheduling with past-sequence-dependent setup times. *Math. Biosci. Eng.* **2022**, *19*, 3110–3126. [CrossRef]
17. Wang, J.-B.; Wang, S.-H.; Gao, K.; Liu, M.; Jia, X. Due-window assignment methods and scheduling with generalized positional-dependent weights. *Asia-Pac. J. Oper. Res.* **2022**, *39*, 2250028. [CrossRef]
18. Shabtay, D.; Steiner, G. A survey of scheduling with controllable processing times. *Discret. Appl. Math.* **2007**, *155*, 1643–1666. [CrossRef]
19. Liu, L.; Wang, J.-J.; Liu, F. Single machine due window assignment and resource allocation scheduling problems with learning and general positional effects. *J. Manuf. Syst.* **2017**, *43*, 1–14. [CrossRef]
20. Lu, Y.-Y.; Wang, T.-T.; Wang, R.-Q.; Li, Y. A note on due-date assignment scheduling with job-dependent learning effects and convex resource allocation. *Eng. Optim.* **2021**, *53*, 1273–1281. [CrossRef]
21. Wang, J.-B.; Lv, D.-Y.; Xu, J.; Ji, P.; Li, F. Bicriterion scheduling with truncated learning effects and convex controllable processing times. *Int. Trans. Oper. Res.* **2021**, *28*, 1573–1593. [CrossRef]
22. Zhao, S. Resource allocation flowshop scheduling with learning effect and slack due window assignment. *J. Ind. Manag. Optim.* **2021**, *17*, 2817–2835. [CrossRef]
23. Lu, S.; Kong, M.; Zhou, Z.; Liu, X.; Liu, S. A hybrid metaheuristic for a semiconductor production scheduling problem with deterioration efect and resource constraints. *Oper. Res.* **2022**, *22*, 5405–5440.
24. Yan, J.-X.; Ren, N.; Bei, H.-B.; Bao, H.; Wang, J.-B. Scheduling with resource allocation, deteriorating effect and group technology to minimize total completion time. *Mathematics* **2022**, *10*, 2983. [CrossRef]
25. Hardy, G.-H.; Littlewood, J.-E.; Polya, G. *Inequalities*, 2nd ed.; Cambridge University Press: Cambridge, UK, 1967.