symmetry

MDPI

Article

Study on Controllable Processing Time and Minmax Group Scheduling with Common Due-Window Assignment

Li-Han Zhang [1,2], Ming-Hui Li [3] and Lin Lin [3,*]

[1] School of Mechatronics Engineering, Shenyang Aerospace University, Shenyang 110136, China; zhanglihan24@stu.sau.edu.cn
[2] Key Laboratory of Rapid Development & Manufacturing Technology for Aircraft (Shenyang Aerospace University), Ministry of Education, Shenyang 110136, China
[3] School of Science, Shenyang Aerospace University, Shenyang 110136, China; liminghui1@stu.sau.edu.cn
* Correspondence: linlin@sau.edu.cn

Abstract

We considerthe single-machine group scheduling problem with controllable processing times (i.e., resource allocation) under a common due-window (*condw*) assignment. The objective is to minimize a total cost composed of earliness, tardiness, due-window-related penalties, and resource consumption. Motivated by realistic production settings such as aerospace component machining and electronics batch assembly, the study addresses the joint optimization of group sequence, job sequence, due-window placement, and resource allocation. For linear and convex resource models, we propose a branch-and-bound ($\widehat{BaB}$) algorithm and efficient heuristics. Numerical experiments show that the $\widehat{BaB}$ algorithm can solve instances with up to 250 jobs and 16 groups. The heuristics ($\widehat{UB}$), including a simulated annealing ($\widehat{SA}$) algorithm, obtain near-optimal solutions with an average error below 0.05% much faster, demonstrating their practical usefulness for real-time scheduling.

Keywords: group scheduling; controllable processing time; minmax; due-window

MSC: 90B35

1. Introduction

Single-machine group scheduling is an important class of production scheduling problems in manufacturing systems, where jobs are partitioned into groups according to their similarities, and setup times are required between different groups. In practical production, the processing times of jobs can often be compressed by allocating additional resources, which leads to the so-called controllable processing time scheduling problem. Meanwhile, to balance customer satisfaction and inventory costs, due-window assignment strategies are widely adopted, among which the common due-window requires all jobs belonging to the same group to share a unified due-window (*condw*). This problem finds broad applications in scenarios such as aerospace component machining, electronics assembly, and batch logistics distribution, and thus possesses significant theoretical value and practical relevance.

With the increasing demands for resource optimization and scheduling in the manufacturing and service industries (see Nowicki and Zdrzalka [1], Shabtay and Steiner [2], Pinedo [3], Oron [4], Lu and Liu [5], Li et al. [6], Liu and Jiang [7], Li et al. [8], Sun et al. [9], Mao et al. [10], Lv and Wang [11], Bosman et al. [12], Zhu et al. [13], Lv and Wang [14], Cohen and Shapira [15], and Sun et al. [16]), controllable processing time (denoted by

check for updates

Academic Editors: Yuankai Li, Zhiguo Shi, Yulian Jiang and Xiaoliang Wang

Received: 14 January 2026
Revised: 5 February 2026
Accepted: 8 February 2026
Published: 14 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

cpt, resource allocations) scheduling problems have been the focus of many scholars. Jiang et al. [17] studied seru production scheduling with the *cpt*. For the total processing cost plus scheduling measures, they showed that some problems are polynomially solvable. Gao et al. [18] considered no-wait job shop scheduling with the *cpt*. Under subcontracting and due date constraints of minimizing the makespan and subcontracting cost, they proposed two mathematical models. Pang and Meng [19] studied robust project scheduling with the *cpt*. Geng and Yuan [20] considered single-machine scheduling with the *cpt*. Under multiple projects, they proved that some problems have NP-hardness. Weng et al. [21] considered scheduling with the multi-resource operations. For the makespan minimization, they proposed a mixed-integer program model. Wang et al. [22] studied single-machine scheduling with the *cpt* and deterioration effects. For a convex resource model, they presented a bicriteria analysis for the total weighted completion time and total resource consumption cost. To solve the problem, they proposed some heuristics and a branch-and-bound algorithm. Zhang et al. [23] investigated single-machine scheduling with the *cpt* and time-dependent learning effects.

Furthermore, group technology (denoted by *gt*) scheduling problems have gradually become a focal point in the fields of operations research and production management (see Liu et al. [24], Huang [25], Bajwa et al. [26], Liao et al. [27]). Recently, Chen et al. [28] considered *gt* scheduling with learning effect and due-window assignment. For the total cost including the due-window penalty and investment, they showed that the problem is polynomially solvable. Yin et al. [29] and Yin and Gao [30] investigated the single-machine *gt* scheduling in which the setup times of groups have linear deterioration effects and processing times of jobs have learning effects. Yin and Gao [30] showed that the makespan minimization is polynomially solvable. For the total weighted completion time minimization, Yin et al. [29] proposed some solution algorithms. Li and Goossens [31] studied the multi-league *gt* scheduling. For two objectives, i.e., the travel distance and capacity violations of venues shared by teams, they proposed a mixed-integer programming model and a two-layer heuristic. Zhao [32] considered the single-machine *gt* scheduling with learning effects and *cpt*. Under a convex resource model of considering the makespan and total resource consumption, they proved that three versions are polynomially solvable. Yuraszeck et al. [33] considered *gt* shop and mixed shop problems of minimizing the makespan. Huang et al. [34] studied the single-machine *gt* scheduling with learning effects and *cpt*. For the convex resource model of minimizing the total resource consumption under the makespan when bounded, they proposed a branch-and-bound and a heuristic.

For many modern industrial processes, logistics and supply chain management, the due-window assignments are very important (see Janiak et al. [35], Geng et al. [36], Wang and Liu [37], Yue and Zhou [38], Tian [39]). In general, there are three types of due-window assignment, i.e., the common due-window (denoted by *condw*, see Wang et al. [40], Hoffmann et al. [41], Qian et al. [42], Bai et al. [43]), the slack due-window (denoted by *slkdw*, see Bai et al. [43], Mor and Mosheiov [44], Mor [45], Sun et al. [46], Qiu and Wang [47]), and the different due-window (denoted by $difdw$, see Lu et al. [48], Sun et al. [49], Wang et al. [50]). Wang et al. [51] studied the *condw* and *slkdw* scheduling with the *cpt*. For the convex resource model of minimizing the general earliness–tardiness cost subject to total resource consumption if bounded, they proposed a branch-and-bound and some heuristics. Sun et al. [52] considered the *condw* and *slkdw* no-wait flow shop scheduling with the *cpt*. The objective is to minimize the sum of general earliness–tardiness cost and total resource consumption, and they proved that the problem is polynomially solvable under some optimal solution properties.

Recently, Liu and Wang [53] and Chen et al. [54] considered *gt* scheduling with due-date assignment and *cpt*. Under linear and convex resource models, they proved

that some special cases of the problem are polynomially solvable. Wang and Liu [55] and Zhang and Yin [56] studied the single-machine gt scheduling with cpt under different due-date assignments aimed at minimizing earliness–tardiness cost, and proposed several solution algorithms. Lv and Wang [57] examined gt scheduling with cpt and $slkdw$, where the objective is to minimize the maximal value of earliness, tardiness, and due-window assignment cost. Under constant processing time, linear, and convex resource models, they presented relevant results. Zhang and Wang [58] investigated single-machine gt scheduling with cpt and $difdw$ under a minmax-type scheduling cost, also considering constant, linear, and convex resource models. Ren and Yang [59] studied the single-machine gt scheduling with cpt and $condw$, with the objective of minimizing the sum of maximal earliness–tardiness cost and total resource allocation cost. However, they focused on a special case (with group-position-dependent penalty, gpp) and proved it to be polynomially solvable.

In this paper, we extend the work of Ren and Yang [59] by studying the general case without the group-position-dependent penalty (gpp) assumption, which is proven to be NP-hard (as indicated in Table 1). Our contribution is threefold. Theoretically, we address this more general and realistic problem setting, thereby filling a gap in the analysis of unrestricted group scheduling. Methodologically, we develop new lower bounds (LB_{linear}, LB_{convex}) and heuristic upper bounds, and propose a dedicated branch-and-bound algorithm along with other heuristic methods, providing a systematic solution framework for this challenging problem. Practically, the model better reflects real-world production environments such as aerospace component machining and electronics batch assembly, and the proposed algorithms can support effective decision-making for real-time scheduling and resource optimization. The remainder of the paper is structured as follows: Section 2 formulates the problem, Section 3 analyzes the linear and convex resource models, Section 4 reports the numerical experiments, and Section 5 concludes.

Table 1. Problems studied.

Problem	Time Complexity	Reference
$1\|gt, slkdw\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}q_1^a+\kappa_{a4}D_a\right\}$	$O(\max(n, m\log m))$	Lv and Wang [57]
$1\|gt, slkdw, linear, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}q_1^{[a]}+\kappa'_{a4}D_{[a]}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Lv and Wang [57]
$1\|gt, slkdw, convex, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}q_1^{[a]}+\kappa'_{a4}D_{[a]}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Lv and Wang [57]
$1\|gt, slkdw, linear\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}q_1^a+\kappa_{a4}D_a\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	NP-hard	Lv and Wang [57]
$1\|gt, slkdw, convex\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}q_1^a+\kappa_{a4}D_a\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	NP-hard	Lv and Wang [57]
$1\|gt, difdw\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a,b}+\kappa_{a4}D_{a,b}\right\}$	$O(\max(n, m\log m))$	Zhang and Wang [58]
$1\|gt, difdw, linear, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}d_1^{a,b}+\kappa'_{a4}D_{a,b}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Zhang and Wang [58]
$1\|gt, difdw, convex, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}d_1^{a,b}+\kappa'_{a4}D_{a,b}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Zhang and Wang [58]
$1\|gt, difdw, linear\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a,b}+\kappa_{a4}D_{a,b}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	NP-hard	Zhang and Wang [58]
$1\|gt, difdw, convex\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a,b}+\kappa_{a4}D_{a,b}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	NP-hard	Zhang and Wang [58]
$1\|gt, condw\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a}+\kappa_{a4}D_{a}\right\}$	$O(\max(n, m\log m))$	Ren and Yang [59]
$1\|gt, condw, linear, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}d_1^{[a]}+\kappa'_{a4}D_{[a]}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Ren and Yang [59]
$1\|gt, condw, convex, gpp\|\sum_{a=1}^{m}\max_{1\le b\le n_{[a]}}\left\{\max\left\{\kappa'_{a1}E_{[a],b},\kappa'_{a2}T_{[a],b}\right\}+\kappa'_{a3}d_1^{[a]}+\kappa'_{a4}D_{[a]}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	$O(\max(mn, m^3))$	Ren and Yang [59]
$1\|gt, condw, linear\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a}+\kappa_{a4}D_{a}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$	NP-hard	This article
$1\|gt, condw, convex\|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^{a}+\kappa_{a4}D_{a}\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b},$	NP-hard	This article

gpp denotes group-position-dependent penalty, κ'_{a1} (κ'_{a2}, κ'_{a3}, κ'_{a4}) denotes the coefficient of *gpp* for $G_{[a]}$; for *slkdw*, $d_1^{a,b}=P_{a,b}^A+q_1^a$, $d_2^{a,b}=P_{a,b}^A+q_2^a$, $D_a=q_2^a-q_1^a$, q_1^a and q_2^a are decision variables; for *difdw*, $d_1^{a,b}$ and $d_2^{a,b}$ are decision variables.

2. Problem Description

There are n jobs to be processed on a single machine, and the jobs are divided into m groups (i.e., groups $G_1, G_2, \ldots, G_m$) based on their similarity characteristics. Let n ($n \geq 1$) be the number of jobs in group $G_a (a = 1, 2, \ldots, m)$, we have $\sum_{a=1}^{m} n_a = n$. During the machining process, there is no idle time for the machine. Furthermore, the machine can only process one job at a time, and it continues to do so until the job is completed. Before group G_a is processed, there is a setup (i.e., preparation) time S_a.

Let $O_{a,b} (b = 1, 2, \ldots, n_a)$ be the bth job in group G_a, and under linear resource model, the actual processing time of $O_{a,b}$ is

$$P_{a,b}^{A} = P_{a,b} - \epsilon_{a,b} u_{a,b},\ 0 \leq u_{a,b} \leq \bar{u}_{a,b} \leq \frac{P_{a,b}}{\epsilon_{a,b}} \tag{1}$$

where $P_{a,b}$ indicates the normal processing time of $O_{a,b}$, $\epsilon_{a,b} (\epsilon_{a,b} \geq 0)$ is its compression factor, $u_{a,b}$ is its resource allocation amount, and $\bar{u}_{a,b}$ is its upper bound of $u_{a,b}$. Under the convex resource model:

$$P_{a,b}^{A} = \left(\frac{w_{a,b}}{u_{a,b}} \right)^{\varrho} \tag{2}$$

where $w_{a,b}$ is job workload of $O_{a,b}$, and $\varrho > 0$ is a constant.

Additionally, let $\left[d_1^a, d_2^a \right]$ be the due-window of $O_{a,b}$, where $d_1^{a,b}$ (resp. $d_2^{a,b}$) is the starting time (resp. completing) time of the due-window, then the due-window size of $O_{a,b}$ is $D_{a,b} = d_2^{a,b} - d_1^{a,b}$. Under common due-window (*condw*) assignment, all jobs belonging to each group G_a $(a = 1, 2, \ldots, m)$ are assigned a unified due-window $\left[d_1^a, d_2^a \right]$ $\left(d_1^a \leq d_2^a \right)$ (i.e., $d_1^{a,b} = d_1^a$ $d_2^{a,b} = d_2^a$), where d_1^a and d_2^a are the starting time and completing time of the due-window, $D_a = d_2^a - d_1^a$ denotes size of the due-window, and d_1^a and d_2^a are decision variables. Let $E_{a,b} = \max\{d_1^a - C_{a,b}, 0\}$ (resp. $T_{a,b} = \max\{0, C_{a,b} - d_2^a\}$) be the earliness (resp. tardiness) cost of job $O_{a,b}$, where $C_{a,b}$ is the completion of $O_{a,b}$. The goal is to find the group-sequence Ψ, the job-sequence ψ_a within G_a, d_1^a, d_2^a, $\left(\widehat{\Psi} = (\Psi, \psi_1, \psi_2, \ldots, \psi_m) \right)$ and the resource allocation amount $u_{a,b}$ for job $O_{a,b}$ such that the following objective cost is minimized, i.e.,

$$\begin{aligned} OC &= \sum_{a=1}^{m} \max_{1 \leq b \leq n_a} \left\{ \max\left\{ \kappa_{a1} E_{a,b} + \kappa_{a3} d_1^a + \kappa_{a4} D_a, \kappa_{a2} T_{a,b} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} \right\} \\ &\quad + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b} \\ &= \sum_{a=1}^{m} \max_{1 \leq b \leq n_a} \left\{ \max\left\{ \kappa_{a1} E_{a,b}, \kappa_{a2} T_{a,b} \right\} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b} \end{aligned}$$

where κ_{a1}, κ_{a2}, κ_{a3}, and κ_{a4}, indicate the unit costs of earliness, tardiness, due-window starting time and size, respectively, and $\tau_{a,b}$ denotes the unit cost of resources consumed by $O_{a,b}$. Adopting the three-field notation (Graham et al. [60], Cai et al. [61] and Chen and Hall [62]), the problems are denoted as follows:

$$1|gt, condw, linear| \sum_{a=1}^{m} \max_{1 \leq b \leq n_a} \left\{ \max\left\{ \kappa_{a1} E_{a,b}, \kappa_{a2} T_{a,b} \right\} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b}$$

and

$$1|gt, condw, convex| \sum_{a=1}^{m} \max_{1 \leq b \leq n_a} \left\{ \max\left\{ \kappa_{a1} E_{a,b}, \kappa_{a2} T_{a,b} \right\} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b}$$

where *linear* (resp. *convex*) denotes the linear (resp. convex) resource model. The problems studied with the *gt* are given in Table 1.

3. Main Results

Mosheiov and Sarig [63] proposed an optimal solution for the problem:

$$1|gt, condw|\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^a+\kappa_{a4}D_a\right\}$$

Based on their conclusions, Lemma 1 is proposed as follows:

Lemma 1. *For group* $G_a(a=1,2,\ldots,m)$*, the objective cost*

$$OC_a=\max_{1\le b\le n_a}\left\{\max\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\}+\kappa_{a3}d_1^a+\kappa_{a4}D_a\right\}$$

can be minimized by the following Table 2:

Table 2. The optimal strategy of *condw*.

	Case	Optimal Due-Window		OC_a
		d_1^*	d_2^*	
1	$\kappa_{a2}\ge\kappa_{a3}$ and $\kappa_{a4}\le\kappa_{a3}$	0	$C_{a,[n_a]}$	$\kappa_{a4}C_{a,[n_a]}$
2	$\kappa_{a2}\ge\kappa_{a3}$, $\kappa_{a4}>\kappa_{a3}$ and $\kappa_{a4}\ge\frac{\kappa_{a2}(\kappa_{a1}+\kappa_{a3})}{\kappa_{a1}+\kappa_{a2}}$	$\frac{\kappa_{a1}C_{a,[1]}+\kappa_{a2}C_{a,[n_a]}}{\kappa_{a1}+\kappa_{a2}}$	$\frac{\kappa_{a1}C_{a,[1]}+\kappa_{a2}C_{a,[n_a]}}{\kappa_{a1}+\kappa_{a2}}$	$\frac{\kappa_{a1}(\kappa_{a3}-\kappa_{a2})}{(\kappa_{a1}+\kappa_{a2})}C_{a,[1]}+\frac{\kappa_{a2}(\kappa_{a1}+\kappa_{a3})}{(\kappa_{a1}+\kappa_{a2})}C_{a,[n_a]}$
3	$\kappa_{a2}\ge\kappa_{a3}$, $\kappa_{a4}>\kappa_{a3}$ and $\kappa_{a4}<\frac{\kappa_{a2}(\kappa_{a1}+\kappa_{a3})}{\kappa_{a1}+\kappa_{a2}}$	$C_{a,[1]}$	$C_{a,[n_a]}$	$(\kappa_{a3}-\kappa_{a4})C_{a,[1]}+\kappa_{a4}C_{a,[n_a]}$
4	$\kappa_{a2}<\kappa_{a3}$ and $\kappa_{a4}\ge\kappa_{a2}$	0	0	$\kappa_{a2}C_{a,[n_a]}$
5	$\kappa_{a2}<\kappa_{a3}$ and $\kappa_{a4}<\kappa_{a2}$	0	$C_{a,[n_a]}$	$\kappa_{a4}C_{a,[n_a]}$

Proof. See Mosheiov and Sarig [63] and Ren and Yang [59]. □

According to Lemma 1, the objective cost OC_a of G_a can be reformulated as follows:

$$OC_a=X_aC_{a,[1]}+Y_aC_{a,[n_a]} \tag{3}$$

where

$$X_a=\begin{cases}0, & \text{Case 1}\\ \frac{\kappa_{a1}(\kappa_{a3}+\kappa_{a2})}{(\kappa_{a1}+\kappa_{a2})}, & \text{Case 2}\\ (\kappa_{a3}-\kappa_{a4}), & \text{Case 3}\\ 0, & \text{Case 4}\\ 0, & \text{Case 5}\end{cases} \tag{4}$$

and

$$Y_a=\begin{cases}\kappa_{a4}, & \text{Case 1}\\ \frac{\kappa_{a2}(\kappa_{a1}+\kappa_{a3})}{(\kappa_{a1}+\kappa_{a2})}, & \text{Case 2}\\ \kappa_{a4}, & \text{Case 3}\\ \kappa_{a2}, & \text{Case 4}\\ \kappa_{a4} & \text{Case 5}\end{cases} \tag{5}$$

3.1. The Linear Model

By the linear resource model (1) and Equation (3), we have

$$\begin{aligned} OC &= \sum_{a=1}^{m} OC_a + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,[b]} u_{a,[b]} \\ &= \sum_{a=1}^{m} \left(X_a C_{a,[1]} + Y_a C_{a,[n_a]} \right) + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,[b]} u_{a,[b]} \\ &= \sum_{a=1}^{m} \left\{ X_a \left[\sum_{r=1}^{a-1} \left(S_r + \tilde{P}_r^A \right) + S_a + P_{a,[1]}^A \right] + Y_a \left[\sum_{r=1}^{a-1} \left(S_r + \tilde{P}_r^A \right) + S_a + \tilde{P}_a^A \right] \right\} + \\ &\quad \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,[b]} u_{a,[b]} \\ &= \sum_{a=1}^{m} \left\{ \left[\sum_{h=a}^{m} (X_h + Y_h) \right] P_{a,[1]}^A + \left[\sum_{h=a+1}^{m} (X_h + Y_h) + Y_a \right] \sum_{j=2}^{n_a} P_{a,[j]}^A \right\} \\ &\quad + \sum_{a=1}^{m} \left[\sum_{h=a}^{m} (X_h + Y_h) \right] S_a + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,[b]} u_{a,[b]} \end{aligned} \tag{6}$$

By substituting Equation (1) into Equation (6), we have

$$OC = \sum_{a=1}^{m} \left[\sum_{h=a}^{m} (X_h + Y_h) \right] S_a + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \frac{\tau_{a,[b]} P_{a,[b]}}{\epsilon_{a,[b]}} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \chi_{a,[b]} P_{a,[b]}^A \tag{7}$$

where

$$\chi_{a,[b]} = \begin{cases} \sum_{h=a}^{m} (X_h + Y_h) - \frac{\tau_{a,[1]}}{\epsilon_{a,[1]}}, & b = 1 \\ \left[\sum_{h=a+1}^{m} (X_h + Y_h) + Y_a \right] - \frac{\tau_{a,[b]}}{\epsilon_{a,[b]}}, & b = 2, 3, \ldots, n_a \end{cases} \quad for\ a = 1, 2, \ldots, m \tag{8}$$

Lemma 2. *For the given job-sequence ψ_a within G_a of the linear model, the optimal resource $u_{a,[b]}^*$ ($a = 1, 2, \ldots, m$ and $b = 1, 2, \ldots, n_a$) is*

$$u_{a,[b]}^* = \begin{cases} 0, & \chi_{a,[b]} < 0 \\ u_{a,[b]} \in [0, \bar{u}_{a,[b]}], & \chi_{a,[b]} = 0 \\ \bar{u}_{a,[b]}, & \chi_{a,[b]} > 0 \end{cases} \quad for\ a = 1, 2, \ldots, m\ and\ b = 1, 2, \ldots, n_a \tag{9}$$

Proof. See Ren and Yang [59]. □

Let

$$\chi_{a,[b]} = \begin{cases} \chi_{a,[b]}^1, & if\ job\ O_{a,[b]}\ is\ in\ the\ first\ position\ of\ \psi_a \\ \chi_{a,[b]}^2, & otherwise \end{cases} \quad for\ b = 1, 2, 3, \ldots, n_a \tag{10}$$

and

$$P_{a,[b]}^A = \begin{cases} P_{a,[b]}, & \chi_{a,[b]} < 0 \\ P_{a,[b]}^A \in \left[P_{a,[b]} - \epsilon_{a,[b]} \bar{u}_{a,[b]}, P_{a,[b]} \right], & \chi_{a,[b]} = 0 \\ P_{a,[b]} - \epsilon_{a,[b]} \bar{u}_{a,[b]}, & \chi_{a,[b]} > 0 \end{cases} \quad for\ b = 1, 2, \ldots, n_a \tag{11}$$

where $\chi_{a,[b]}^1 = \sum_{h=a}^{m} (X_h + Y_h) - \frac{\tau_{a,[1]}}{\epsilon_{a,[1]}}$ and $\chi_{a,[b]}^2 = \left[\sum_{h=a+1}^{m} (X_h + Y_h) + Y_a \right] - \frac{\tau_{a,[b]}}{\epsilon_{a,[b]}}$. As in Ren and Yang [59], for G_a of the given group-sequence, the optimal job-sequence can be obtained in $O(n_a)$ time, which is obtained by Algorithm 1, where $\varpi_{a,[b]} = \chi_{a,[b]}^1 P_{a,[b]}^A + \chi_{a,[b]}^2 \sum_{j=2}^{n_a} P_{a,[b]}^A$ and $\varpi_{a,[b]}$ denotes the minimum value of the term $\sum_{a=1}^{m} \sum_{b=1}^{n_a} \chi_{a,[b]} P_{a,[b]}^A$ resulting from scheduling $O_{a,[b]}$ in the first position in ψ_a. Obviously, the optimal job-sequence of all groups can be determined in $\sum_{a=1}^{m} O(n_a) = O(n)$ time.

Algorithm 1: Optimal job-sequence ψ_a: *linear*

Step 1. Initialization of values.
  Compute X_a and Y_a for $G_a \leftarrow$ Equations (4) and (5)
Step 2. Iterative position analysis.
  Set the initial group counter: $a \leftarrow 1$
  While $a \leq m$ do
    Set the initial position counter: $r \leftarrow 1$
    While $r \leq n_a$ do
      For each job $O_{a,[b]}(b = 1, 2, \ldots, n_a)$
        Compute $\chi_{a,[b]} \leftarrow$ Equation (10)
        Compute $P^A_{a,[b]} \leftarrow$ Equation (11)
      Update the position counter: $r \leftarrow r + 1$
    Update the group counter: $a \leftarrow a + 1$
  Identify the first processed job: $\min\left\{\omega_{a,[b]}, b = 1, 2, \ldots, n_a\right\}$
  Generate the optimal sequence ψ^*_a
Step 3. Optimal resource allocation and due-window.
  Determine optimal resource allocation: $u^*_{a,[b]} \leftarrow$ Equation (9)
  Determine optimal d^*_1 and $d^*_2 \leftarrow$ Lemma 1

3.1.1. An Upper Bound

For the general case, we suspect

$$1|gt, condw, linear| \sum_{a=1}^{m} \max_{1\leq b\leq n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d^a_1 + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}$$

is NP-hard; hence, an exact branch-and-bound algorithm can be proposed. From Equation (7), Zhang and Wang [58], $\sum_{a=1}^{m}[\sum_{h=a}^{m}(X_h + Y_h)]S_a$ is minimized by the descending order of $\frac{X_a+Y_a}{S_a}$. $\sum_{a=1}^{m}\sum_{b=1}^{n_a}\chi_{a,[b]}P^A_{a,[b]}$ is minimized by the descending order of $\frac{X_a+Y_a}{\tilde{P}^A_a}$, where $\tilde{P}^A_a = \sum_{b=1}^{n_a} P^A_{a,[b]}$. Hence, the following heuristic (i.e., Algorithm 2) is proposed as an upper bound $\widehat{UB}$ for the group-sequence of $1|gt, condw, linear|\sum_{a=1}^{m}\max_{1\leq b\leq n_a}$ $\left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d^a_1 + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}$.

3.1.2. A Lower Bound

Let $\Psi = (\Psi_{group-s}, \Psi_{group-u})$ be a group-sequence, where $\Psi_{group-s}(\Psi_{group-u})$ is the set of scheduled (unscheduled) part of groups, and there is g groups in the set $\Psi_{group-s}$. From Equation (7), we have

$$\begin{aligned}
OC = {} & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r + Y_r) + \sum_{r=g+1}^{m}\left(X_{[r]} + Y_{[r]}\right)\right]S_a + \sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}X_{[r]} + Y_{[r]}\right]S_{[a]} \\
& + \sum_{a=1}^{g}\sum_{b=1}^{n_a}\frac{\tau_{a,b}P_{a,b}}{\epsilon_{a,b}} + \sum_{a=g+1}^{m}\sum_{b=1}^{n_a}\frac{\tau_{[a],[b]}P_{[a],[b]}}{\epsilon_{[a],[b]}} + \sum_{a=1}^{g}\left\{\left[\sum_{r=a}^{g}(X_r + Y_r)\right]\right. \\
& \left. + \left[\sum_{r=g+1}^{m}\left(X_{[r]} + Y_{[r]}\right)\right] - \frac{\tau_{a,1}}{\epsilon_{a,1}}\right\}P^A_{a,1} + \sum_{a=g+1}^{m}\left\{\left[\sum_{r=a}^{m}\left(X_{[r]} + Y_{[r]}\right)\right] - \frac{\tau_{[a],[1]}}{\epsilon_{[a],[1]}}\right\}P^A_{[a],[1]} \\
& + \sum_{a=1}^{g}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{g}(X_r + Y_r) + \sum_{r=g+1}^{m}\left(X_{[r]} + Y_{[r]}\right) + Y_a - \frac{\tau_{a,b}}{\epsilon_{a,b}}\right]P^A_{a,b} \\
& + \sum_{a=g+1}^{m}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{m}\left(X_{[r]} + Y_{[r]}\right) + Y_{[a]} - \frac{\tau_{[a],[b]}}{\epsilon_{[a],[b]}}\right]P^A_{[a],[b]}
\end{aligned} \tag{12}$$

Algorithm 2: Upper bound algorithm: $\widehat{UB} - linear$

Step 1. Calculate X_a and $Y_a \leftarrow$ Equations (4) and (5)
  Calculate $\tilde{P}_a^A \leftarrow \tilde{P}_a^A = \sum_{b=1}^{n_a} P_{a,[b]}^A$
Step 2. Determine job-sequence for each group.
  The optimal job-sequence ψ_a of $G_a \leftarrow$ Algorithm 1
Step 3. Determine group-sequence.
  Step 3.1: Arrange groups in descending order of $\frac{X_a+Y_a}{S_a}$
    Compute $OC_A \leftarrow$ Equation (7)
  Step 3.2: Arrange groups in descending order of $\frac{X_a+Y_a}{\tilde{P}_a^A}$
    Compute $OC_B \leftarrow$ Equation (7)
  Step 3.3: Arrange groups in increasing order of S_a
    Compute $OC_C \leftarrow$ Equation (7)
  Step 3.4: Arrange groups in descending order of $X_a + Y_a$
    Compute $OC_D \leftarrow$ Equation (7)
  Step 3.5: Arrange groups in increasing order of $\tilde{P}_a^A$
    Compute $OC_E \leftarrow$ Equation (7)
Step 4. Best group-sequence selection.
  Compute $OC_{best} \leftarrow OC_{best} = \min\{OC_A, OC_B, OC_C, OC_D, OC_E\}$

From (12), $\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)\right]S_a$, $\sum_{a=1}^{g}\sum_{b=1}^{n_a}\frac{\tau_{a,b}P_{a,b}}{\epsilon_{a,b}}$, $\sum_{a=g+1}^{m}\sum_{b=1}^{n_a}\frac{\tau_{[a],[b]}P_{[a],[b]}}{\epsilon_{[a],[b]}}$, $\sum_{a=1}^{g}\left\{\left[\sum_{r=a}^{g}(X_r+Y_r)\right]-\frac{\tau_{a,1}}{\epsilon_{a,1}}\right\}P_{a,1}^A$, and $\sum_{a=1}^{g}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+Y_a-\frac{\tau_{a,b}}{\epsilon_{a,b}}\right]P_{a,b}^A$ are known and constants. Hence, the first lower bound (denoted by LB_{linear}) is obtained as follows:

$$
\begin{aligned}
LB_{linear}^1 = {} & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{<r>}+Y_{<r>})\right]S_a+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}X_{<r>]}+Y_{<r>}\right]S_{min} \\
& +\sum_{a=1}^{g}\sum_{b=1}^{n_a}\frac{\tau_{a,b}P_{a,b}}{\epsilon_{a,b}}+\sum_{a=g+1}^{m}\sum_{b=1}^{n_a}\frac{\tau_{[a],[b]}P_{[a],[b]}}{\epsilon_{[a],[b]}}+\sum_{a=1}^{g}\left\{\left[\sum_{r=a}^{g}(X_r+Y_r)\right]\right. \\
& \left.+\left[\sum_{r=g+1}^{m}\left(X_{(r)}+Y_{(r)}\right)\right]-\frac{\tau_{a,1}}{\epsilon_{a,1}}\right\}P_{a,1}^A+\sum_{a=g+1}^{m}\left\{\left[\sum_{r=a}^{m}\left(X_{(r)}+Y_{(r)}\right)\right]-\frac{\tau_{(a),[1]}}{\epsilon_{(a),[1]}}\right\}P_{(a),[1]}^A \qquad (13) \\
& +\sum_{a=1}^{g}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}\left(X_{(r)}+Y_{(r)}\right)+Y_a-\frac{\tau_{a,b}}{\epsilon_{a,b}}\right]P_{a,b}^A \\
& +\sum_{a=g+1}^{m}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{m}\left(X_{(r)}+Y_{(r)}\right)+Y_{[a]}-\frac{\tau_{(a),[b]}}{\epsilon_{(a),[b]}}\right]P_{(a),[b]}^A
\end{aligned}
$$

where $S_{\min} = \min\{S_a \in \Psi_{group-u}\}$, $X_{<g+1>}+Y_{<g+1>} > X_{<g+2>}+Y_{<g+2>} > \cdots > X_{<m>}+Y_{<m>}$ and $\frac{X_{(g+1)}+Y_{(g+1)}}{\tilde{P}_{(g+1)}^A} > \frac{X_{(g+2)}+Y_{(g+2)}}{\tilde{P}_{(g+2)}^A} > \cdots > \frac{X_{(m)}+Y_{(m)}}{\tilde{P}_{(m)}^A}$. Similarly, the second LB_{linear} is

$$
\begin{aligned}
LB^2_{linear} = & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{<r>}+Y_{<r>})\right]S_a+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}X_{<r>]}+Y_{<r>}\right]S_{<a>} \\
& +\sum_{a=1}^{g}\sum_{b=1}^{n_a}\frac{\tau_{a,b}P_{a,b}}{\epsilon_{a,b}}+\sum_{a=g+1}^{m}\sum_{b=1}^{n_a}\frac{\tau_{[a],[b]}P_{[a],[b]}}{\epsilon_{[a],[b]}}+\sum_{a=1}^{g}\left\{\left[\sum_{r=a}^{g}(X_r+Y_r)\right]\right. \\
& \left.+\left[\sum_{r=g+1}^{m}\left(X_{(r)}+Y_{(r)}\right)\right]-\frac{\tau_{a,1}}{\epsilon_{a,1}}\right\}P^A_{a,1}+\sum_{a=g+1}^{m}\left\{\left[\sum_{r=a}^{m}\left(X_{(r)}+Y_{(r)}\right)\right]-\frac{\tau_{(a),[1]}}{\epsilon_{(a),[1]}}\right\}P^A_{(a),[1]} \\
& +\sum_{a=1}^{g}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}\left(X_{(r)}+Y_{(r)}\right)+Y_a-\frac{\tau_{a,b}}{\epsilon_{a,b}}\right]P^A_{a,b} \\
& +\sum_{a=g+1}^{m}\sum_{b=2}^{n_a}\left[\sum_{r=a+1}^{m}\left(X_{(r)}+Y_{(r)}\right)+Y_{[a]}-\frac{\tau_{(a),[b]}}{\epsilon_{(a),[b]}}\right]P^A_{(a),[b]}
\end{aligned}
\tag{14}
$$

where $\frac{X_{<g+1>}+Y_{<g+1>}}{S_{<g+1>}} > \frac{X_{<g+2>}+Y_{<g+2>}}{S_{<g+2>}} > \cdots > \frac{X_{<m>}+Y_{<m>}}{S_{<m>}}$ and $\frac{X_{(g+1)}+Y_{(g+1)}}{\tilde{P}^A_{(g+1)}} > \frac{X_{(g+2)}+Y_{(g+2)}}{\tilde{P}^A_{(g+2)}} > \cdots > \frac{X_{(m)}+Y_{(m)}}{\tilde{P}^A_{(m)}}$. Note that $\frac{X_{<a>}+Y_{<a>}}{S_{<a>}}$ and $\frac{X_{(a)}+Y_{(a)}}{\tilde{P}^A_{(a)}}(a=g+1,g+2,\ldots,m)$ do not mean the same group. To make the LB_{linear} tighter, we choose the largest value of LB^1_{linear} and LB^2_{linear}, i.e.,

$$
LB_{linear} = \max\left\{LB^1_{linear}, LB^2_{linear}\right\} \tag{15}
$$

3.2. The Convex Model

For the *convex* function of

$$
1|gt,condw,convex|\sum_{a=1}^{m}\max_{1\le b\le n_a}\left\{\max\left\{\kappa_{a1}E_{a,b},\kappa_{a2}T_{a,b}\right\}+\kappa_{a3}d_1^a+\kappa_{a4}D_a\right\}+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,b}u_{a,b}
$$

from Ren and Yang [59], we have

$$
\begin{aligned}
OC = & \sum_{a=1}^{m}\left\{\left[\sum_{h=a}^{m}(X_h+Y_h)\right]\left(\frac{w_{a,[1]}}{u_{a,[1]}}\right)^{\varrho}+\left[\sum_{h=a+1}^{m}(X_h+Y_h)+Y_a\right]\sum_{j=2}^{n_a}\left(\frac{w_{a,[j]}}{u_{a,[j]}}\right)^{\varrho}\right\} \\
& +\sum_{a=1}^{m}\left[\sum_{h=a}^{m}(X_h+Y_h)\right]S_a+\sum_{a=1}^{m}\sum_{b=1}^{n_a}\tau_{a,[b]}u_{a,[b]}
\end{aligned}
\tag{16}
$$

Lemma 3. *For the given job-sequence ψ_a within G_a of convex model, the optimal resource $u^*_{a,[b]}$ $(a=1,2,\ldots,m$ and $b=1,2,\ldots,n_a)$ is*

$$
u^*_{a,[b]}(\Psi,\psi_1,\psi_2,\ldots,\psi_m) =
$$

$$
\begin{cases}
\left[\frac{\varrho\sum_{h=a}^{m}(X_h+Y_h)}{\tau_{a,[1]}}\right]^{\frac{1}{\varrho+1}}\left(w_{a,[1]}\right)^{\frac{\varrho}{\varrho+1}}, & a=1,2,\ldots,m;\ b=1 \\
\left[\frac{\varrho\left(\sum_{h=a+1}^{m}(X_h+Y_h)+Y_a\right)}{\tau_{a,[b]}}\right]^{\frac{1}{\varrho+1}}\left(w_{a,[b]}\right)^{\frac{\varrho}{\varrho+1}}, & a=1,2,\ldots,m;\ b=2,3,\ldots,n_a
\end{cases}
\tag{17}
$$

According Equations (16) and (17), we have

$$
\begin{aligned}
OC = & \sum_{a=1}^{m}\left[\sum_{h=a}^{m}(X_h+Y_h)\right]S_a+\left(\varrho^{-\frac{\varrho}{\varrho+1}}+\varrho^{\frac{1}{\varrho+1}}\right)\sum_{a=1}^{m}\left\{\left[\sum_{h=a}^{m}(X_h+Y_h)\right]^{\frac{1}{\varrho+1}}\left(\tau_{a,[1]}w_{a,[1]}\right)^{\frac{\varrho}{\varrho+1}}\right. \\
& \left.+\left[\sum_{h=a+1}^{m}(X_h+Y_h)+Y_a\right]^{\frac{1}{\varrho+1}}\sum_{j=2}^{n_a}\left(\tau_{a,[j]}w_{a,[j]}\right)^{\frac{\varrho}{\varrho+1}}\right\}
\end{aligned}
\tag{18}
$$

Let $\xi_{a,[b]} = \left(\tau_{a,[b]} w_{a,[b]}\right)^{\frac{\varrho}{\varrho+1}}$, $(a = 1, 2, \ldots, m, b = 1, 2, \ldots, n_a)$, as in Ren and Yang [59], for G_a of the given group-sequence, the optimal job-sequence can be obtained in $O(n_a)$ time, which is obtained by Algorithm 3. Obviously, the optimal job-sequence of all groups can be determined in $\sum_{a=1}^{m} O(n_a) = O(n)$ time.

Algorithm 3: Optimal job-sequence ψ_a: *convex*

Step 1. Initialization of values
  Compute X_a and $Y_a \leftarrow$ Equations (4) and (5)
Step 2. Generate sequence
  Compute $\xi_{a,[b]} \leftarrow \xi_{a,[b]} = \left(\tau_{a,[b]} w_{a,[b]}\right)^{\frac{\varrho}{\varrho+1}}$
  Select the job $O_{a,[1]}$ with largest $\xi_{a,[b]}$ value
  Other jobs in any order
Step 3. Optimal resource allocation and due-window.
  Determine optimal resource allocation: $u^*_{a,[b]} \leftarrow$ Equation (17)
  Determine optimal d_1^* and $d_2^* \leftarrow$ Lemma 1

3.2.1. An Upper Bound

For the general case, we suspect

$$1|gt, condw, convex| \sum_{a=1}^{m} \max_{1 \le b \le n_a} \left\{ \max\{\kappa_{a1} E_{a,b}, \kappa_{a2} T_{a,b}\} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b}$$

is NP-hard; hence, an exact branch-and-bound algorithm can be proposed. Similarly, the following heuristic (Algorithm 4) is proposed as an upper bound for the group-sequence of

$$1|gt, condw, convex| \sum_{a=1}^{m} \max_{1 \le b \le n_a} \left\{ \max\{\kappa_{a1} E_{a,b}, \kappa_{a2} T_{a,b}\} + \kappa_{a3} d_1^a + \kappa_{a4} D_a \right\} + \sum_{a=1}^{m} \sum_{b=1}^{n_a} \tau_{a,b} u_{a,b}.$$

Algorithm 4: Upper bound algorithm: $\widehat{UB} - convex$

Step 1. Calculate X_a and $Y_a \leftarrow$ Equations (4) and (5)
  Calculate $\xi_a \leftarrow \xi_a = \sum_{b=1}^{n_a} \xi_{a,[b]}$
Step 2. Determine job-sequence for each group.
  The optimal job-sequence ψ_a of $G_a \leftarrow$ Algorithm 3
Step 3. Determine group-sequence.
  Step 3.1: Arrange groups in descending order of $\frac{X_a + Y_a}{S_a}$
    Compute $OC_A \leftarrow$ Equation (18)
  Step 3.2: Arrange groups in descending order of $\frac{X_a + Y_a}{\xi_a}$
    Compute $OC_B \leftarrow$ Equation (18)
  Step 3.3: Arrange groups in increasing order of S_a
    Compute $OC_C \leftarrow$ Equation (18)
  Step 3.4: Arrange groups in descending order of $X_a + Y_a$
    Compute $OC_D \leftarrow$ Equation (18)
  Step 3.5: Arrange groups in increasing order of ξ_a
    Compute $OC_E \leftarrow$ Equation (18)
Step 4. Best group-sequence selection.
  Compute $OC^* \leftarrow OC^* = \min\{OC_A, OC_B, OC_C, OC_D, OC_E\}$

3.2.2. A Lower Bound

Similar to Section 3.1.2, from Equation (18), we have

$$\begin{aligned} OC \quad = \quad & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{[r]}+Y_{[r]})\right]S_a+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{[r]}+Y_{[r]})\right]S_{[a]} \\ & +\left(\varrho^{-\frac{\varrho}{\varrho+1}}+\varrho^{\frac{1}{\varrho+1}}\right)\left\{\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{[r]}+Y_{[r]})\right]\xi_{a,[1]}\right. \\ & +\sum_{a=1}^{g}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{[r]}+Y_{[r]})+Y_a\right]\sum_{j=2}^{n_a}\xi_{a,[j]} \\ & \left.+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{[r]}+Y_{[r]})\right]\xi_{[a],[1]}+\sum_{a=g+1}^{m}\left[\sum_{r=a+1}^{m}(X_{[r]}+Y_{[r]})+Y_{[a]}\right]\sum_{j=2}^{n_a}\xi_{[a],[j]}\right\} \end{aligned} \tag{19}$$

From Equation (19), $\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)\right]S_a$, $\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)\right]\xi_{a,[1]}$ and $\sum_{a=1}^{g}\left[\sum_{r=a+1}^{g}(X_r+Y_r)\right]\sum_{j=2}^{n_a}\xi_{a,[j]}$ are constants. Hence, the first lower bound (denoted by LB_{convex}) is obtained as follows:

$$\begin{aligned} & LB^1_{convex} \\ = \quad & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{<r>}+Y_{<r>})\right]S_a+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{<r>}+Y_{<r>})\right]S_{\min} \\ & +\left(\varrho^{-\frac{\varrho}{\varrho+1}}+\varrho^{\frac{1}{\varrho+1}}\right)\left\{\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{(r)}+Y_{(r)})\right]\xi_{a,[1]}\right. \\ & +\sum_{a=1}^{g}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{(r)}+Y_{(r)})+Y_a\right]\sum_{j=2}^{n_a}\xi_{a,[j]} \\ & \left.+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{(r)}+Y_{(r)})\right]\xi_{(a),[1]}+\sum_{a=g+1}^{m}\left[\sum_{r=a+1}^{m}(X_{(r)}+Y_{(r)})+Y_{(a)}\right]\sum_{j=2}^{n_a}\xi_{(a),[j]}\right\} \end{aligned} \tag{20}$$

where $S_{\min}=\min\{S_a\in\Psi_{group-u}\}$, $X_{<g+1>}+Y_{<g+1>}>X_{<g+2>}+Y_{<g+2>}>\cdots>X_{<m>}+Y_{<m>}$ and $\frac{X_{(g+1)}+Y_{(g+1)}}{\xi_{(g+1)}}>\frac{X_{(g+2)}+Y_{(g+2)}}{\xi_{(g+2)}}>\cdots>\frac{X_{(m)}+Y_{(m)}}{\xi_{(m)}}$.

Let $X_{\min}+Y_{\min}=\min\{X_a+Y_a\in\Psi_{group-u}\}$, similarly, the second LB_{convex} is

$$\begin{aligned} & LB^2_{convex} \\ = \quad & \sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{\min}+Y_{\min})\right]S_a+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{\min}+Y_{\min})\right]S_{\min} \\ & +\left(\varrho^{-\frac{\varrho}{\varrho+1}}+\varrho^{\frac{1}{\varrho+1}}\right)\left\{\sum_{a=1}^{g}\left[\sum_{r=a}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{\min}+Y_{\min})\right]\xi_{a,[1]}\right. \\ & +\sum_{a=1}^{g}\left[\sum_{r=a+1}^{g}(X_r+Y_r)+\sum_{r=g+1}^{m}(X_{\min}+Y_{\min})+Y_a\right]\sum_{j=2}^{n_a}\xi_{a,[j]} \\ & \left.+\sum_{a=g+1}^{m}\left[\sum_{r=a}^{m}(X_{\min}+Y_{\min})\right]\xi_{(a),[1]}+\sum_{a=g+1}^{m}\left[\sum_{r=a+1}^{m}(X_{\min}+Y_{\min})+Y_{(a)}\right]\sum_{j=2}^{n_a}\xi_{(a),[j]}\right\} \end{aligned} \tag{21}$$

where $\xi_{(g+1)}<\xi_{(g+2)}<\cdots<\xi_{(m)}$. Note that $\xi_{(a)}$, and $X_{(a)}+Y_{(a)}(a=g+1,g+2,\ldots,m)$ do not mean the same group. Similar to the *linear* model, we choose the largest value of LB^1_{convex} and LB^2_{convex}, i.e.,

$$LB_{convex}=\max\left\{LB^1_{convex},LB^2_{convex}\right\} \tag{22}$$

3.3. Solution Algorithm

3.3.1. Branch-And-Bound

For the *linear* and *convex* models, based on the lower bounds Equations (15) and (22), and upper bounds Algorithms 2 and 4, we propose an exact branch-and-bound ($\widehat{BaB}$) to address the problems

$$1|gt, condw, linear| \sum_{a=1}^{m} \max_{1\le b\le n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d_1^a + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}$$

and

$$1|gt, condw, convex| \sum_{a=1}^{m} \max_{1\le b\le n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d_1^a + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}.$$

The polynomial-time algorithm for the special case cannot be applied to the general case. To address this, we developed a novel Branch-and-Bound ($\widehat{BaB}$) framework and designed tight lower bounds for effective pruning, which is crucial for tackling this NP-hard problem (Algorithm 5).

Algorithm 5: $\widehat{BaB} - linear/convex$

Step 1. Initial upper bound ($\widehat{UB} - linear/\widehat{UB} - convex$) $\Psi^0 \leftarrow$ Algorithm 2/ Algorithm 4
    Calculate $OC(\Psi^*) = OC(\Psi^0) \leftarrow$ Equation (7)/Equation (18)
Step 2. For each node (group) G_a:
        Calculate $LB_{linear}(G_a)/LB_{convex}(G_a) \leftarrow$ Equation (15)/(Equation (22)
        If $LB_{linear}(G_a)/LB_{convex}(G_a) \geq OC(\Psi^*)$:
            Delete node G_a and its subtree
    For each unfathomed node G_j:
        Calculate $LB_{linear}(G_j)/LB_{convex}(G_j) \leftarrow$ Equation (15)/(Equation (22)
        If $LB_{linear}(G_j)/LB_{convex}(G_j) \geq OC(\Psi^*)$:
            Delete node G_j and its subtree
        Else:
            Complete the group-sequence to obtain Ψ
            Compute $OC(\Psi)$
            If $OC(\Psi) < OC(\Psi^*)$:
                Update: $\Psi^* \leftarrow \Psi$
            Else:
                Discard Ψ
Step 3. If all nodes have been searched, terminate the algorithm
    Output the optimal group-sequence Ψ^* and $OC(\Psi^*)$

The branching strategy expands a node by generating child nodes, each of which appends a distinct unscheduled group to the current partial sequence. Nodes are explored in the order they are generated, corresponding to a depth-first traversal of the tree. To enhance efficiency, a pruning rule is applied at each node: if the calculated lower bound for the partial sequence is greater than or equal to the objective value of the current best-known complete solution, the node and its entire subtree are discarded. The algorithm terminates when all nodes in the search tree have been either fully explored or pruned.

3.3.2. Simulated Annealing

As in Kirkpatrick et al. [64], Lv and Wang [65] and Sun et al. [66], the simulated annealing ($\widehat{SA}$) is proposed to solve

$$1|gt, condw, linear| \sum_{a=1}^{m} \max_{1\le b\le n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d_1^a + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}$$

and

$$1|gt, condw, convex| \sum_{a=1}^{m} \max_{1\le b\le n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d_1^a + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}.$$

The $\widehat{SA}$ algorithm (Algorithm 6) is introduced to systematically search for and approximate optimal solutions within acceptable time, bridging the gap between the fast constructive heuristic ($\widehat{UB}$) and the time-consuming exact algorithm ($\widehat{BaB}$).

Algorithm 6: $\widehat{SA} - linear/convex$

Step 1. Initial feasible solution $\Psi^0 \leftarrow$ Algorithm 2/Algorithm 4
  Current solution $\Psi \leftarrow \Psi^0$
  Compute $OC(\Psi^*) = OC(\Psi^0) \leftarrow$ Equations (7) and (18)
  $ST \leftarrow 1.0$ (starting temperature)
  $LTL \leftarrow 1 \times 10^{-16}$ (lower temperature limit)
  $CR \leftarrow 0.99999999$ (cooling rate)
  $IT \leftarrow 1000 \times n$ (iterations)
  $RDF \leftarrow U(0,1)$ (random decision factor)
Step 2. Iterative Optimization
  While $IT > 0$ and $|ST| > LTL$ do:
    Randomly select positions $\hat{R}1, \hat{R}2$
    If $\hat{R}1 = \hat{R}2$:
      $IT \leftarrow IT + 1$, continue
    Else:
      Generate neighbor $\Psi' \leftarrow$ Swap $\hat{R}1$ and $\hat{R}2$ in Ψ
    $\Delta E \leftarrow OC\left(\Psi'\right) - OC(\Psi)$
    If $\Delta E < 0$ or $\exp(-\Delta E/ST) > RDF$:
      Accept $\Psi \leftarrow \Psi'$
      If $OC\left(\Psi'\right) < OC(\Psi^*)$:
        $OC(\Psi^*) \leftarrow OC\left(\Psi'\right)$
        $\Psi^* \leftarrow \Psi'$
    Update temperature: $ST \leftarrow ST \times CR$
    $IT \leftarrow IT - 1$
Step 3. Termination
  Output best group-sequence Ψ^* and $OC(\Psi^*)$

4. Numerical Experiments

To analyze the effectiveness and performance of $\widehat{UB}$ (Algorithm 2 and Algorithm 4), $\widehat{BaB}$ (Algorithm 5) and $\widehat{SA}$ (Algorithm 6), a set of experiments were conducted. Random instances were generated, where n = 50, 100, 150, 200, 250; m = 8, 10, 12, 14, 16; $P_{a,b} \in U[1,100]$; $\epsilon_{a,b} \in U[1,100]$; $w_{a,b} \in U[1,100]$; $S_a \in U[1,10]$; $\kappa_{ai} \in U[1,10]$ ($i = 1,2,3,4$); $\tau_{a,b} \in U[1,10]$, $\varrho = 2$. These algorithms were run on a DELL personal computer with

3.10 GHz CPU, 8 GB RAM, and these algorithms were coded in C++ and compiled with Microsoft Visual Studio 2019.

For the $\widehat{BaB} - linear$ and $\widehat{BaB} - convex$), the average and maximum node numbers and CPU time (in millisecond, ms) were recorded. For the $\widehat{UB} - linear$, $\widehat{UB} - convex$, $\widehat{SA} - linear$ and $\widehat{SA} - convex$, the average and maximum CPU time (in seconds) were recorded. The error of the $\widehat{UB} - linear$, $\widehat{UB} - convex$, $SA - linear$ and $SA - convex$ is calculated as

$$\frac{OC(H) - OC^*}{OC^*} \times 100\%,$$

where $OC(H)$ is the objective cost obtained by the algorithm H, $H \in \{\widehat{UB} - linear, \widehat{UB} - convex, \widehat{SA} - linear, \widehat{SA} - convex\}$ and OC^* is the optimal cost obtained by $\widehat{BaB}$. For each combination of m and n, 20 random examples were generated. From Tables 3 and 4, it can be seen that $\widehat{BaB} - linear$ and $\widehat{BaB} - convex$ can still be solved 250 jobs and 16 groups within the time (i.e., $\leq$) 985967.6343 ms. For the heuristics, the $\widehat{UB} - linear$ and $\widehat{UB} - convex$ have the shortest CPU time than $\widehat{SA} - linear$ and $\widehat{SA} - convex$. For the error of heuristics, from Table 5, it can be seen that the maximum error of all heuristics (i.e., $\widehat{UB} - linear, \widehat{UB} - convex, \widehat{SA} - linear, \widehat{SA} - convex$) is 0.048737%. From Table 5, we also obtain that the average error of $\widehat{SA} - linear$ and $\widehat{SA} - convex$ outperforms $\widehat{UB} - linear$ and $\widehat{UB} - convex$.

A comprehensive analysis of these tables yields the following main conclusions regarding algorithm performance: (1) Efficiency vs. Optimality: The exact $\widehat{BaB}$ algorithm guarantees optimality but exhibits rapidly increasing computation time with problem scale, confirming the problem's inherent complexity. In contrast, both heuristics provide solutions within fractions of a second. (2) Quality of Heuristics: As noted, the $\widehat{SA}$ algorithm consistently delivers lower average errors than $\widehat{UB}$, demonstrating its superior search capability. The superior accuracy of $\widehat{SA}$ stems directly from its iterative refinement mechanism, which allows it to escape local optima and explore a broader solution space than the single-pass, rule-based $\widehat{UB}$. (3) Practical Utility: The results delineate a clear trade-off: $\widehat{UB}$ is the fastest heuristic for a quick, high-quality solution, while $\widehat{SA}$ offers the best balance, providing near-optimal solutions with only a modest increase in runtime.

Table 3. CPU time (ms) of *condw* for linear models.

n	*m*	*UB-Linear*		*SA-Linear*		*BB*		*BB* Nodes	
		Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**
50	8	14.9088	23.208	296.2808	346.56	49.74969	67.7076	1054.35	1452
	10	20.063	33.68	379.8795	512.31	302.095635	495.0918	6759.3	11,131
	12	24.5106	33.588	503.1198	740.412	2237.70766	3162.2199	45,319	67,463
	14	32.1279	47.222	705.8009	1067.066	23,078.83646	44,647.7313	407,422.4	754,785
	16	47.0688	78.912	870.7608	1636.832	195,306.0996	381,843.1225	3,473,117.7	6,588,502
100	8	22.9456	42.76	1058.9224	1806.856	81.30942	127.4241	1172.5	1881
	10	32.2715	103.07	1264.1235	1950.62	547.097055	884.3089	7575.05	12,345
	12	42.3954	135.084	1584.3606	2058.48	4164.310825	6437.5259	56,047.6	85,700
	14	54.3641	125.986	2121.0665	4143.468	32,222.81033	49,109.7604	419,441.7	644,204
	16	59.332	156.176	2420.5336	3440.784	356,116.6568	635,590.2297	4,376,056.7	7,738,105
150	8	28.4788	41.576	2014.8352	2453.64	95.8844	125.4708	1059.3	1345
	10	41.8725	57.38	2657.5895	3432.59	747.119815	1314.9022	7673.45	13,048
	12	58.3632	104.784	3153.7596	3749.568	5291.199145	9228.4548	53,505.95	91,720
	14	69.3007	151.298	4114.9143	5451.152	44,373.66308	96,211.7484	439,917.1	904,255
	16	78.0968	170.192	4819.2376	7734.832	423,181.6796	855,484.0617	3,974,097	7,923,352
200	8	41.396	75.528	3753.4344	4702.888	131.489335	200.906	1128.25	1552
	10	51.3975	74.72	4731.963	5372.66	983.114315	1249.2476	8074.95	10,178
	12	60.4698	83.544	5724.6174	7344.504	7362.903935	11,935.5246	59,416.4	92,948
	14	79.3401	132.79	6659.6999	9273.488	51,553.84987	90,554.1662	428,132.9	743,408
	16	119.2776	286.368	7647.4616	10,240.128	512,735.7826	926,534.6346	4,066,520	7,324,635
250	8	45.482	77.048	5022.452	5814.928	152.116975	240.403	1171.1	1956
	10	65.2595	124.33	6464.126	7764.13	1017.597095	1513.1619	7852.55	11,483
	12	76.275	148.284	8154.108	9426.096	8298.62114	14,924.36	59,375.75	103,418
	14	94.8668	205.324	10,188.8367	12,719.686	74,559.64571	111,269.9429	519,645.8	765,085
	16	102.8472	240.848	11,500.812	14,687.68	581,300.0625	985,967.6343	4,106,525.95	6,954,585

Table 4. CPU time (ms) of *condw* for convex models.

n	*m*	*UB-Convex*		*SA-Convex*		*BB*		*BB* Nodes	
		Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**
50	8	14.0092	25.504	213.8704	309.072	47.2445	81.9703	915.4	1685
	10	16.63	25.85	279.4435	308.62	281.884485	554.474	5383.3	10,601
	12	25.0812	63.804	387.4434	528.768	1995.445185	3384.0362	31,876.75	50,135
	14	41.7207	162.218	574.0336	922.418	19,324.33216	44,301.9899	280,893.2	685,144
	16	45.4176	77.728	610.2368	924.064	154,054.2619	302,089.3153	2,334,002.25	4,609,124
100	8	18.1308	34.68	636.4244	1175.192	78.056675	107.3817	1127.75	1551
	10	22.104	32.38	778.073	980.88	501.678595	812.902	6724.45	10,779
	12	29.031	44.244	1042.2702	1313.892	3896.99359	6146.9746	49441.55	76,893
	14	46.1069	125.636	1343.6241	1807.666	29,904.56491	53,424.1858	360,634.9	644,945
	16	53.2976	116.64	1558.0792	1997.168	303,743.1704	508,514.5931	3,382,350.4	5,727,921
150	8	22.1508	39.712	1158.6384	1499.072	87.604695	139.8171	1065.8	1569
	10	29.144	40.14	1554.333	1927.48	661.687145	1135.0919	7426.2	12,400
	12	50.1834	153.684	1999.341	2300.208	4928.357055	8079.5484	51,338.3	82,531
	14	54.6539	97.79	2491.9146	3623.158	38,734.83223	73,870.4459	393,856.6	756,958
	16	52.9984	85.2	2749.6112	3666.56	355,472.9039	542,691.2738	3,408,258.4	5,164,454
200	8	24.4656	36.848	1846.0644	2136.664	110.255325	172.0278	1148.4	1800
	10	37.2655	77.33	2361.4815	3878.1	798.17429	1045.0738	7808	10,457
	12	42.7518	85.608	2993.4126	4620.96	5872.14315	9883.743	56,480	93,448
	14	56.3367	173.11	3584.9611	4886.35	49,372.92988	81,875.8839	441,629.35	729,433
	16	80.056	153.856	4444.7072	5434.544	442,259.9251	826,716.0385	3,680,805.9	6,892,847
250	8	29.6284	84.144	2746.1744	4058.76	125.26891	211.019	1193.45	1956
	10	38.445	75.74	3660.476	5377.82	846.657065	1114.7384	7648.55	10,338
	12	59.6256	119.952	4462.869	5602.236	7011.789075	10,914.5426	58,498.6	90,152
	14	61.2129	186.466	5241.8205	6640.886	57,949.30358	77,964.78	459,999.1	619,674
	16	74.1832	147.392	6112.5728	7533.376	497,255.3724	864,228.0558	3,845,406.95	6,697,976

Table 5. Error of *condw*.

n	*m*	*UB-Linear*		*UB-Convex*		*SA-Linear*		*SA-Convex*	
		Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**
50	8	0.006997	0.030574	0.001819	0.019022	0.000290	0.003236	0.001670	0.017365
	10	0.007024	0.023686	0.000913	0.018251	0.001356	0.006647	0.000849	0.016983
	12	0.014013	0.048737	0.002965	0.022467	0.001727	0.007394	0.002501	0.019864
	14	0.017983	0.045223	0.007845	0.037220	0.002017	0.008469	0.006726	0.029332
	16	0.016463	0.038894	0.010157	0.030139	0.003160	0.009910	0.009238	0.030139
100	8	0.005491	0.029006	0.000000	0.000000	0.000492	0.003043	0.000000	0.000000
	10	0.007255	0.026654	0.001207	0.015567	0.000204	0.000813	0.000699	0.007010
	12	0.009145	0.035949	0.001432	0.016744	0.000650	0.003516	0.001016	0.011891
	14	0.007649	0.030009	0.004740	0.025903	0.000679	0.002896	0.004395	0.023051
	16	0.008909	0.031258	0.004135	0.028901	0.000650	0.001833	0.003608	0.020566
150	8	0.001950	0.012827	0.000000	0.000000	0.000326	0.002434	0.000000	0.000000
	10	0.003657	0.026714	0.000000	0.000000	0.000282	0.001257	0.000000	0.000000
	12	0.008587	0.021467	0.000000	0.000000	0.000412	0.001509	0.000000	0.000000
	14	0.008086	0.029781	0.001682	0.015398	0.000559	0.001255	0.001531	0.015062
	16	0.009259	0.030160	0.002680	0.016431	0.000458	0.001397	0.002430	0.016431
200	8	0.001939	0.016315	0.000000	0.000000	0.000240	0.001874	0.000000	0.000000
	10	0.004639	0.020718	0.000000	0.000000	0.000280	0.000835	0.000000	0.000000
	12	0.005200	0.020197	0.000000	0.000000	0.000238	0.001079	0.000000	0.000000
	14	0.005225	0.016524	0.000741	0.014817	0.000324	0.001547	0.000741	0.014817
	16	0.007382	0.017377	0.000336	0.006712	0.000356	0.001095	0.000336	0.006712
250	8	0.002052	0.008562	0.000000	0.000000	0.000090	0.000503	0.000000	0.000000
	10	0.002446	0.015276	0.000000	0.000000	0.000168	0.001366	0.000000	0.000000
	12	0.003926	0.030186	0.000000	0.000000	0.000241	0.001338	0.000000	0.000000
	14	0.007543	0.016428	0.000000	0.000000	0.000328	0.001399	0.000000	0.000000
	16	0.005247	0.016696	0.000000	0.000000	0.000275	0.000890	0.000000	0.000000

5. Summary and Future Research

This paper considered *gt* scheduling with the *cpt* and *condw*. Under the *linear* and *convex* models, the goal of optimization is to minimize

$$\sum_{a=1}^{m} \max_{1\le b\le n_a} \left\{ \max\{\kappa_{a1}E_{a,b}, \kappa_{a2}T_{a,b}\} + \kappa_{a3}d_1^a + \kappa_{a4}D_a \right\} + \sum_{a=1}^{m}\sum_{b=1}^{n_a} \tau_{a,b}u_{a,b}.$$

To address this NP-hard optimization problem, we proposed a suite of solution methodologies. A dedicated branch-and-bound (B&B) algorithm was developed, incorporating newly designed lower bounds (LB_{linear}, LB_{convex}) and heuristic-based upper bounds to prune the

search space efficiently. Complementing the exact algorithm, effective heuristic methods, including constructive rules and a simulated annealing metaheuristic, were provided for obtaining high-quality solutions within short computational times. Numerical experiments conducted on a wide range of test instances demonstrated the efficacy of our approaches. The proposed method provides production planners with a practical tool for generating near-optimal schedules in real time, which is crucial for dynamic manufacturing scenarios such as aerospace component machining.

Future research could explore several promising extensions of this work: (1) flow shop scheduling with group technology and controllable processing times, including Sun et al. [67], Marquez and Ribeiro [68], Liu and Shi [69], Sotskov et al. [70], and Fasihi et al. [71]; (2) group scheduling with deterioration effects, as studied by He et al. [72], Sun et al. [73], and Pei et al. [74]; (3) group scheduling with learning effects, investigated by Jiang et al. [75], Zhao [76], and Liu and Wang [77].

Author Contributions: Methodology, L.-H.Z. and M.-H.L.; writing—original draft, L.-H.Z.; writing—review and editing, L.-H.Z., M.-H.L. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the fundamental research funds for the universities of liaoning province (Project No. LJ222510143003).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare there are no conflicts of interest.

References

1. Nowicki, E.; Zdrzałka, S. A survey of results for sequencing problems with controllable processing times. *Discret. Appl. Math.* **1990**, *26*, 271–287. [CrossRef]
2. Shabtay, D.; Steiner, G. Optimal due date assignment and resource allocation to minimize the weighted number of tardy jobs on a single machine. *Manuf. Serv. Oper. Manag.* **2007**, *9*, 332–350. [CrossRef]
3. Pinedo, M.L. *Scheduling Theory, Algorithms, and Systems*, 5th ed.; Springer: New York, NY, USA, 2016.
4. Oron, D. Scheduling controllable processing time jobs with position-dependent workloads. *Int. J. Prod. Econ.* **2016**, *173*, 153–160. [CrossRef]
5. Lu, X.; Liu, Z. An optimal online algorithm for fractional scheduling on uniform machines with three hierarchies. *J. Syst. Sci. Complex.* **2016**, *29*, 1650–1657. [CrossRef]
6. Li, K.; Chen, J.; Fu, H.; Jia, Z.; Fu, W.Z. Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint. *Appl. Soft Comput.* **2019**, *82*, 105585. [CrossRef]
7. Liu, W.W.; Jiang, C. Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation. *Eng. Optim.* **2020**, *52*, 74–89. [CrossRef]
8. Li, F.; Xu, Z.; Li, H. A multi-agent based cooperative approach to decentralized multi-project scheduling and resource allocation. *Comput. Ind. Eng.* **2021**, *151*, 106961. [CrossRef]
9. Sun, X.; Liu, T.; Geng, X.-N.; Hu, Y.; Xu, J.-X. Optimization of scheduling problems with deterioration effects and an optional maintenance activity. *J. Sched.* **2023**, *26*, 251–266. [CrossRef]
10. Mao, R.-R.; Lv, D.-Y.; Ren, N.; Wang, J.-B. Supply chain scheduling with deteriorating jobs and delivery times. *J. Appl. Math. Comput.* **2024**, *70*, 2285–2312. [CrossRef]
11. Lv, D.-Y.; Wang, J.-B. No-idle flow shop scheduling with deteriorating jobs and common due date under dominating machines. *Asia-Pac. J. Oper. Res.* **2024**, *41*, 2450003. [CrossRef]
12. Bosman, T.; van Ee, M.; Ergen, E.; Imreh, C.; Marchetti-Spaccamela, A.; Skutella, M.; Stougie, L. Total completion time scheduling under scenarios. *Theory Comput. Syst.* **2025**, *69*, 33. [CrossRef]
13. Zhu, Z.; Chu, F.; Sun, L.; Liu, M. Single machine scheduling with resource allocation and learning effect considering the rate-modifying activity. *Appl. Math. Model.* **2013**, *37*, 5371–5380. [CrossRef]
14. Lv, D.-Y.; Wang, J.-B. Considering the peak power consumption problem with learning and deterioration effect in flow shop scheduling. *Comput. Ind. Eng.* **2024**, *197*, 110599. [CrossRef]

15. Cohen, E.; Shapira, D. Minimising the makespan on parallel identical machines with log-linear position-dependent processing times. *J. Oper. Res. Soc.* **2025**, *76*, 581–589. [CrossRef]
16. Sun, Y.; Lu, Y.-Y.; Lv, D.-Y.; Wang, J.-B. Single-machine setup time scheduling with general linear deterioration subject to makespan and total completion time minimization. *J. Oper. Res. Soc.* **2025**. [CrossRef]
17. Jiang, Y.J.; Zhang, Z.; Song, X.L.; Yin, Y. Scheduling controllable processing time jobs in seru production system with resource allocation. *J. Oper. Res. Soc.* **2022**, *73*, 2551–2571. [CrossRef]
18. Gao, J.S.; Zhu, X.M.; Bai, K.Y.; Zhang, R.T. New controllable processing time scheduling with subcontracting strategy for no-wait job shop problem. *Int. J. Prod. Res.* **2022**, *60*, 2254–2274. [CrossRef]
19. Pang, N.S.; Meng, Q.C. Resource allocation in robust scheduling. *J. Oper. Res. Soc.* **2023**, *74*, 125–142.
20. Geng, Z.C.; Yuan, J.J. Single-machine scheduling of multiple projects with controllable processing times. *Eur. J. Oper. Res.* **2023**, *308*, 1074–1090. [CrossRef]
21. Weng, W.; Chu, C.; Wu, P. Resource allocation to minimize the makespan with multi-resource operations. *J. Syst. Sci. Complex.* **2024**, *37*, 2054–2070. [CrossRef]
22. Wang, J.-B.; Wang, Y.-C.; Wan, C.; Lv, D.-Y.; Zhang, L. Controllable processing time scheduling with total weighted completion time objective and deteriorating jobs. *Asia-Pac. J. Oper. Res.* **2024**, *41*, 2350026. [CrossRef]
23. Zhang, Y.; Sun, X.; Liu, T.; Wang, J.; Geng, X.-N. Single-machine scheduling simultaneous consideration of resource allocations and exponential time-dependent learning effects. *J. Oper. Res. Soc.* **2025**, *76*, 528–540. [CrossRef]
24. Liu, F.; Yang, J.; Lu, Y. Solution algorithms for single-machine group scheduling with ready times and deteriorating jobs. *Eng. Optim.* **2019**, *51*, 862–874. [CrossRef]
25. Huang, X. Bicriterion scheduling with group technology and deterioration effect. *J. Appl. Math. Comput.* **2019**, *60*, 455–464. [CrossRef]
26. Bajwa, N.; Melouk, S.; Bryant, P. A hybrid heuristic approach to minimize number of tardy jobs in group technology systems. *Int. Trans. Oper. Res.* **2019**, *26*, 1847–1867. [CrossRef]
27. Liao, B.; Wang, X.; Zhu, X.; Yang, S.; Pardalos, P.M. Less is more approach for competing groups scheduling with different learning effects. *J. Comb. Optim.* **2020**, *39*, 33–54. [CrossRef]
28. Chen, K.; Han, S.; Huang, H.; Ji, M. A group-dependent due window assignment scheduling problem with controllable learning effect. *Asia-Pac. J. Oper. Res.* **2023**, *40*, 2250025. [CrossRef]
29. Yin, N.; He, H.; Zhao, Y.; Chang, Y.; Wang, N. Integrating group setup time deterioration effects and job processing time learning effects with group technology in single-machine green scheduling. *Axioms* **2025**, *14*, 480. [CrossRef]
30. Yin, N.; Gao, M. Single-machine group scheduling with general linear deterioration and truncated learning effects. *Comput. Appl. Math.* **2024**, *43*, 386. [CrossRef]
31. Li, M.; Goossens, D. Grouping and scheduling multiple sports leagues: An integrated approach. *J. Oper. Res. Soc.* **2025**, *76*, 739–757. [CrossRef]
32. Zhao, T. Research on group scheduling with effects of learning and resource allocations. *J. Math.* **2025**, *2025*, 4146871. [CrossRef]
33. Yuraszeck, F.; Mejia, G.; Luer-Villagra, A. An adapted constraint-programming formulation of the resource-constrained project scheduling problem applied to the identical parallel machines group shop and mixed shop scheduling problems. *Int. Trans. Oper. Res.* **2025**, *32*, 1422–1441. [CrossRef]
34. Huang, X.; He, H.; Bei, H.; Zhao, Y.; Wang, N.; Chang, Y. Group-scheduling with simultaneous learning effects and convex resource allocations. *Oper. Res. Perspect.* **2025**, *15*, 100370. [CrossRef]
35. Janiak, A.; Janiak, W.A.; Krysiak, T.; Kwiatkowski, T. A survey on scheduling problems with due windows. *Eur. J. Oper. Res.* **2015**, *242*, 347–357. [CrossRef]
36. Geng, X.-N.; Sun, X.Y.; Wang, J.; Pan, L. Scheduling on proportionate flow shop with job rejection and common due date assignment. *Comput. Ind. Eng.* **2023**, *181*, 109317. [CrossRef]
37. Wang, X.; Liu, W. Delivery scheduling with variable processing times and due date assignments. *Bull. Malays. Math. Sci. Soc.* **2025**, *48*, 76. [CrossRef]
38. Yue, Q.; Zhou, S.H. Due-window assignment scheduling problem with stochastic processing times. *Eur. J. Oper. Res.* **2021**, *290*, 453–468. [CrossRef]
39. Tian, Y. Single-machine due-window assignment scheduling with resource allocation and generalized earliness/tardiness penalties. *Asia-Pac. J. Oper. Res.* **2022**, *39*, 2150041. [CrossRef]
40. Wang, J.-B.; Hu, Y.; Zhang, B. Common due-window assignment for single-machine scheduling with generalized earliness/tardiness penalties and a rate-modifying activity. *Eng. Optim.* **2021**, *53*, 496–512. [CrossRef]
41. Hoffmann, J.; Neufeld, J.S.; Buscher, U. Minimizing the earliness–tardiness for the customer order scheduling problem in a dedicated machine environment. *J. Sched.* **2024**, *27*, 525–543. [CrossRef]
42. Qian, J.; Chang, G.; Zhang, X. Single-machine common due-window assignment and scheduling with position-dependent weights, delivery time, learning effect and resource allocations. *J. Appl. Math. Comput.* **2024**, *70*, 1965–1994. [CrossRef]

43. Bai, B.; Wei, C.-M.; He, H.-Y.; Wang, J.-B. Study on single-machine common/slack due-window assignment scheduling with delivery times, variable processing times and outsourcing. *Mathematics* **2024**, *12*, 2833. [CrossRef]
44. Mor, B.; Mosheiov, G. Minsum and minmax scheduling on a proportionate flowshop with common flow-allowance. *Eur. J. Oper. Res.* **2016**, *254*, 360–370. [CrossRef]
45. Mor, B. Minmax common flow-allowance problems with convex resource allocation and position-dependent workloads. *J. Comb. Optim.* **2022**, *43*, 79–97. [CrossRef]
46. Sun, X.; Geng, X.-N.; Wang, J.Y.; Pan, L. Slack due window assignment scheduling in the single-machine with controllable processing times. *J. Ind. Manag. Optim.* **2024**, *20*, 15–35. [CrossRef]
47. Qiu, X.-Y.; Wang, J.-B. Single-machine scheduling with mixed due-windows and deterioration effects. *J. Appl. Math. Comput.* **2025**, *71*, 2527–2542. [CrossRef]
48. Lu, Y.-Y.; Zhang, S.; Tao, J.-Y. Earliness-tardiness scheduling with delivery times and deteriorating jobs. *Asia-Pac. J. Oper. Res.* **2025**, *42*, 2450009. [CrossRef]
49. Sun, X.; Geng, X.-N.; Liu, T. Due-window assignment scheduling in the proportionate flow shop setting. *Ann. Oper. Res.* **2020**, *292*, 113–131. [CrossRef]
50. Wang, J.-B.; Lv, D.-Y.; Wan, C. Proportionate flow shop scheduling with job-dependent due windows and position-dependent weights. *Asia-Pac. J. Oper. Res.* **2025**, *42*, 2450011. [CrossRef]
51. Wang, J.-B.; Sun, Z.-W.; Gao, M. Research on single-machine scheduling with due-window assignment and resource allocation under total resource consumption cost is bounded. *J. Appl. Math. Comput.* **2025**, *71*, 7905–7927. [CrossRef]
52. Sun, Y.; Lv, D.-Y.; Huang, X. Properties for due window assignment scheduling on a two-machine no-wait proportionate flow shop with learning effects and resource allocation. *J. Oper. Res. Soc.* **2026**, *77*, 599–615. [CrossRef]
53. Liu, W.; Wang, X. Group technology scheduling with due-date assignment and controllable processing times. *Processes* **2023**, *11*, 1271. [CrossRef]
54. Chen, Y.; Ma, X.; Zhang, G.; Cheng, Y. On optimal due date assignment without restriction and resource allocation in group technology scheduling. *J. Comb. Optim.* **2023**, *45*, 64. [CrossRef]
55. Wang, X.; Liu, W. Single machine group scheduling jobs with resource allocations subject to unrestricted due date assignments. *J. Appl. Math. Comput.* **2024**, *70*, 6283–6308. [CrossRef]
56. Zhang, L.-H.; Yin, N. Study on single-machine group scheduling with convex resource allocations and different due-date assignments. *Bull. Malays. Math. Sci. Soc.* **2026**, *49*, 33. [CrossRef]
57. Lv, D.-Y.; Wang, J.-B. Single-machine group technology scheduling with resource allocation and slack due window assignment including minmax criterion. *J. Oper. Res. Soc.* **2025**, *76*, 1696–1712. [CrossRef]
58. Zhang, L.-H.; Wang, J.-B. Resource allocation and minmax scheduling under group technology and different due-window assignments. *Axioms* **2025**, *14*, 827. [CrossRef]
59. Ren, J.F.; Yang, Y. Common due-window assignment and minmax scheduling with resource allocation and group technology on a single machine. *Eng. Optim.* **2022**, *54*, 1819–1834. [CrossRef]
60. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* **1979**, *5*, 287–326.
61. Cai, X.Q.; Wu, X.; Zhou, X. *Optimal Stochastic Scheduling*; Springer: New York, NY, USA, 2014.
62. Chen, Z.-L.; Hall, N.G. *Supply Chain Scheduling*; Springer: Cham, Switzerland, 2022.
63. Mosheiov, G.; Sarig, A. Minmax scheduling problems with a common due-window. *Comput. Oper. Res.* **2009**, *36*, 1886–1892. [CrossRef]
64. Kirkpatrick, S.; Gelatt, C.D.J.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
65. Lv, D.-Y.; Wang, J.-B. Research on two-machine flow shop scheduling problem with release dates and truncated learning effects. *Eng. Optim.* **2025**, *57*, 1828–1848. [CrossRef]
66. Sun, Z.-W.; Lv, D.-Y.; Ji, P.; Wang, J.-B. Minimizing total completion time scheduling problem with ready times and linear deterioration functions of processing times. *J. Appl. Math. Comput.* **2026**, *72*, 42. [CrossRef]
67. Sun, X.; Geng, X.-N.; Liu, F. Flow shop scheduling with general position weighted learning effects to minimise total weighted completion time. *J. Oper. Res. Soc.* **2021**, **72**, 2674–2689. [CrossRef]
68. Marquez, C.R.H.; Ribeiro, C.C. Shop scheduling in manufacturing environments: A review. *Int. Trans. Oper. Res.* **2022**, *29*, 3237–3293. [CrossRef]
69. Liu, L.X.; Shi, L.Y. Automatic design of efficient heuristics for two-stage hybrid flow shop scheduling. *Symmetry* **2022**, *14*, 632. [CrossRef]
70. Sotskov, Y.N.; Allahverdi, A.; Lai, T.C. Flowshop scheduling problem to minimize total completion time with random and bounded processing times. *J. Oper. Res. Soc.* **2004**, **55**, 277–286. [CrossRef]
71. Fasihi, M.; Tavakkoli-Moghaddam, R.; Jolai, F. A bi-objective re-entrant permutation flow shop scheduling problem: Minimizing the makespan and maximum tardiness. *Oper. Res.* **2023**, *23*, 29. [CrossRef]

72. He, N.; Qiao, Y.; Wu, N.; Qu, T. Total completion time minimization for scheduling of two-machine flow shop with deterioration jobs and setup time. *Adv. Mech. Eng.* **2017**, *9*,1687814017698887. [CrossRef]
73. Sun, Y.; He, H.; Zhao, Y.; Wang, J.-B. Minimizing makespan scheduling on a single machine with general positional deterioration effects. *Axioms* **2025**, *14*, 290. [CrossRef]
74. Pei, J.; Wang, X.; Fan, W.; Pardalos, P.M.; Liu, X. Scheduling step-deteriorating jobs on bounded parallel-batching machines to maximise the total net revenue. *J. Oper. Res. Soc.* **2019**, *70*, 1830–1847. [CrossRef]
75. Jiang, Z.-Y.; Chen, F.-F.; Zhang, X.-D. Single-machine scheduling with times-based and job-dependent learning effect. *J. Oper. Res. Soc.* **2017**, *68*, 809–815. [CrossRef]
76. Zhao, S. Scheduling jobs with general truncated learning effects including proportional setup times. *Comput. Appl. Math.* **2022**, *41*, 146. [CrossRef]
77. Liu, Z.; Wang, J.-B. Single-machine scheduling with simultaneous learning effects and delivery times. *Mathematics* **2024**, *12*, 2522. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.