Journal of Applied Mathematics and Computing
https://doi.org/10.1007/s12190-024-02216-y

ORIGINAL RESEARCH

Single machine group scheduling jobs with resource allocations subject to unrestricted due date assignments

Xuyin Wang[1] **· Weiguo Liu**[1]

Received: 9 July 2024 / Revised: 3 August 2024 / Accepted: 7 August 2024

Abstract
This paper investigates the single machine group scheduling with unrestricted (different) due date assignments and resource allocations (controllable processing times). The resource allocations mean that the actual job processing times are convex decreasing function of their consumption of resources. To solve the general problem of minimizing the weighted sum of earliness, tardiness, due date assignment cost and resource consumption cost (the weights are job-dependent weights), we propose lower and upper bounds to speed up the search process of the branch-and-bound algorithm. To solve this problem quickly and accurately, we also propose a heuristic algorithm. Computational results are tested to evaluate the performance of the algorithms.

Keywords Scheduling · Due date assignment · Resource allocation · Single machine · Group technology

Mathematics Subject Classification 90B35 · 68M20

1 Introduction

Scheduling problems with resource allocations (it is also called controllable processing times) have been the focus of many scholars (Shabtay and Steiner [1], Yedidsiona and Shabtay [2], Sun et al. [3], and Kovalev et al. [4]). In 2021, Zhao [5] considered the flow shop scheduling with resource allocation and learning effects under no-wait setting. For the slack due-window, Zhao [5] proved that some versions of scheduling cost (i.e., weighted sum of earliness-tardiness and due-window assignment) and resource cost can be solved in polynomial time. Lu et al. [6] considered due-date assignment problem with resource allocation and learning effects. Mor et al. [7] addressed single-machine scheduling with resource allocation. For some NP-hard problems, they proposed heuristic algorithms. Tian [8] studied scheduling with

✉ Weiguo Liu
lwgwinterplum@163.com

1 School of Management, Northwest Normal University, Lanzhou 730070, China

resource allocation and common/slack due-window, they sowed that four versions of scheduling cost (i.e., weighted sum of earliness-tardiness, number of early and tardy job, and due-window assignment) are polynomially solvable. Wang and Wang [9] considered single-machine resource allocation scheduling with the time-dependent learning effect. Zhang et al. [10] and Li et al. [11] studied two-agent single machine resource allocation scheduling with deteriorating jobs. Wang et al. [12] investigated the single-machine scheduling with deteriorating jobs and convex resource allocation. A bicriteria analysis on total weighted completion time and resource consumption cost is provided. Qian et al. [13] addressed single-machine due-window assignment scheduling with resource allocations and learning effect. Under delivery times, they proved that some problems are polynomially solvable. Sun et al. [14] studied single machine resource allocation scheduling with slack due window assignment. Zhang et al. [15] considered single machine resource allocation scheduling with exponential time-dependent learning effects.

In addition, some researchers examined the models with group technology (see Potts and Van Wassenhove [16], Webster and Baker [17], Wu and Lee [18], Li et al. [19], Ji et al. [20], Ji et al. [21], and Zhang et al. [22]). In 2019, Huang [23] and Liu et al. [24] considered single machine group scheduling with deterioration effects. Bajwa et al. [25] studied single machine group scheduling with the sequence-independent setup times. For the number of tardy jobs minimization, they proposed a hybrid heuristic and particle swarm optimization meta-heuristics. Xu et al. [26] examined group scheduling with deteriorating effects. Under a nonperiodical maintenance, they proposed some heuristic algorithms. Chen et al. [27] addressed single machine group scheduling with due date assignment. Under three due date methods, the goal is to minimize the cost function including earliness-tardiness, due date assignment and flow time, they proved that the problem can be solved in polynomial time. He et al. [28] considered the flowshop group scheduling with sequence-dependent setup times. For the makespan minimization, they proposed some heuristic algorithms to solve the problem. Wang and Ye [29] delved into group scheduling with random learning effects. They proved that some problems polynomial solvable.

Under many modern industrial process, there has been increasing attention to the scheduling problems involving both group technology and resource allocation (Shabtay et al. [30], Zhu et al. [31], Wang et al. [32], and Lv et al. [33]). In 2023, Yan et al. [34] examined the single machine group problem with learning effects and resource allocation. For the total completion time minimization subject to limited resource availability, they proposed some algorithms. Liu and Wang [35] and He et al. [36] examined the single machine group scheduling with resource allocations and position-dependent weights. Under common and slack due-date assignments, Liu and Wang [35] proved that some special cases can be solved in polynomial time; For a general case of the problem, He et al. [36] proposed some heuristic algorithms and a branch-and-bound. Li et al. [37] considered the single machine group scheduling with convex resource allocation and learning effect. Under common due date (denoted by $\widetilde{con}$) assignment, for the non-regular objection, they proposed the heuristic and branch-and-bound algorithms. Recently, Chen et al. [38] studied the single machine group scheduling with resource allocation. Under the different due dates (denoted by $\widetilde{dif}$) assignment, they proved that a special case of two scheduling problems (i.e., the linear and convex resource consumption functions) can be solved in polynomial time. In light of the significance of group scheduling with resource allocation in real manufacturing environments, in this paper, we continue the study of Chen et al. [38], the purpose is to consider the general case of Chen et al. [38]. Contributions of this study are presented as follows: (i) The general group scheduling with resource allocation and $\widetilde{dif}$ is modeled and studied. (ii) To solve the general problem of Chen et al. [38], the structural properties are derived, and solution algorithms (including a branch-and-bound algorithm and a heuristic algorithm) are proposed. (iii) Numerical tests are presented to evaluate the efficiency of the solution algorithms.

The rest of this paper is organized as follows: In Sect. 2, we give a description of the problem. In Sect. 3, we presents some preliminary properties. In Sect. 4, we proposed the solution algorithms to solve the general problem. In Sect. 5, we present computational study for the algorithms. In Sect. 6, we present the conclusions.

2 Problem assumptions

In this paper, the problem formulation can be described as follows: There are n jobs $J_1, J_2, \ldots, J_n$ grouped into z groups $\widehat{G}_1, \widehat{G}_2, \ldots, \widehat{G}_z$, and these jobs to be processed on a single machine, where there are n_h jobs in the group $\widehat{G}_h$, i.e., $\widehat{G}_h = \{J_{h,1}, J_{h,2}, \ldots, J_{h,n_h}\}$ $(h = 1, 2, \ldots, z)$, $\sum_{h=1}^{z} n_h = n$. Let s_h denote the setup time of $\widehat{G}_h$; $C_{h,j}$ be the completion time of $J_{h,j}$ in $\widehat{G}_h$. For the $\widetilde{dif}$ assignment, the due date of $J_{h,j}$ is $d_{h,j}$. As in Shabtay et al. [30] and Chen et al. [38], the actual processing time of $J_{h,j}$ is

$$p_{h,j}^{Act} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}, h = 1, 2, \ldots, z; j = 1, 2, \ldots, n_h, \tag{1}$$

where $\varpi_{h,j}$ is workload of $J_{h,j}$, $\eta > 0$ is a constant, $u_{h,j}$ is the amount of resource allocated to $J_{h,j}$. The goal is to find a schedule δ, due dates and resource allocations to minimize the following cost function:

$$F(\delta, d_{h,j}, u_{h,j}|_{h=1}^{z},{}_{j=1}^{n_h}) = \sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + v_{h,j}u_{h,j}), \tag{2}$$

where $E_{h,j} = \max\{d_{h,j} - C_{h,j}, 0\}$ (resp. $T_{h,j} = \max\{C_{h,j} - d_{h,j}, 0\}$) is the earliness (resp. tardiness) of $J_{h,j}$ (Yang et al. [39], Geng et al. [40], Lv and Wang [41], and Wang et al. [42]), $\alpha_{h,j}$ (resp. $\beta_{h,j}$, $\gamma_{h,j}$) denotes unit earliness (resp. tardiness, due date) cost of $J_{h,j}$, i.e., the weight $\alpha_{h,j}$ ($\beta_{h,j}$, $\gamma_{h,j}$) is job-dependent weight of $J_{h,j}$, and $v_{h,j}$ is the unit consumption cost of $J_{h,j}$ (i.e., the cost associated with the per unit consumption of resource), i.e.,

$$1\left|\widetilde{gt}, \widetilde{dif}, p_{h,j}^{Act} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}\right|\sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + v_{h,j}u_{h,j}), \tag{3}$$

where 1 denotes a single machine setting, $\widetilde{gt}$ represents group technology, the second field (i.e., $\widetilde{gt}, \widetilde{dif}, p^{Act}_{h,j} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}$) is job characteristics, the third field $\sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + \upsilon_{h,j}u_{h,j})$ refers to the optimal criterion.

3 Preliminary properties

Let $[r]$ denote job (group) scheduled in the rth position in a sequence, from Chen et al. [38], we have

$$\begin{aligned}&\sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + \upsilon_{h,j}u_{h,j})\\ &\quad= \sum_{h=1}^{z}\Psi_{[h]}\left(\sum_{k=1}^{h}s_{[k]}\right) + \left(\eta^{\frac{1}{\eta+1}} + \eta^{\frac{-\eta}{\eta+1}}\right)\\ &\quad\sum_{h=1}^{z}\sum_{j=1}^{n_{[h]}}\theta_{[h],[j]}\left(\sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}},\end{aligned} \tag{4}$$

where $\theta_{[h],[j]} = \left(\varpi_{[h],[j]}\upsilon_{[h],[j]}\right)^{\frac{\eta}{\eta+1}}$.

From Chen et al. [38] and Eq. (4), $1\left|\widetilde{gt}, \widetilde{dif}, p^{Act}_{J_{h,j}} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}\right|\sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + \upsilon_{h,j}u_{h,j})$ reduces to a purely combinatorial optimization of minimizing Eq. (4).

Lemma 1 *For each $h = 1, 2, \ldots, z$, if $\psi_{h,o} \geq \psi_{h,\chi}$ implies $\theta_{h,o} \leq \theta_{h,\chi}$, the optimal job sequence in $\widehat{G}_h$ is in non-decreasing order of $\theta_{h,j}$ (or in non-increasing order of $\psi_{h,j}$), where $\theta_{h,j} = \left(\varpi_{h,j}\upsilon_{h,j}\right)^{\frac{\eta}{\eta+1}}$, $h = 1, 2, \ldots, z$.*

Proof By Eq. (4), for group $\widehat{G}_{[h]}$, we only need to minimize

$$F_{[h]} = \sum_{j=1}^{n_{[h]}}\theta_{[h],[j]}\left(\sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}}. \tag{5}$$

By the adjacent interchange method, let $\delta_{[h]} = [\pi_1, J_{h,o}, J_{h,\chi}, \pi_2]$ and $\delta'_{[h]} = [\pi_1, J_{h,\chi}, J_{h,o}, \pi_2]$, where π_1 and π_2 are partial schedules, and $J_{h,o}$ (resp. $J_{h,\chi}$) is scheduled at λth (resp. $(\lambda+1)$th) position in $\delta_{[h]}$. Let X (resp. Y) be the partial sum of $F_{[h]}$ in π_1 (resp. π_2), we have

$$F_{[h]}(\delta_{[h]}) = X + \theta_{[h],o}\left(\psi_{[h],o} + \psi_{[h],\chi} + \sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=\lambda+2}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}}$$
$$+\theta_{[h],\chi}\left(\psi_{[h],\chi} + \sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=\lambda+2}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}} + Y, \tag{6}$$

and

$$F_{[h]}(\delta'_{[h]}) = X + \theta_{[h],\chi}\left(\psi_{[h],\chi} + \psi_{[h],o} + \sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=\lambda+2}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}}$$
$$+\theta_{[h],o}\left(\psi_{[h],o} + \sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=\lambda+2}^{n_{[h]}}\psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}} + Y. \tag{7}$$

We assume that $\psi_{h,o} \geq \psi_{h,\chi}$, $\theta_{h,o} \leq \theta_{h,\chi}$, $Z = \sum_{k=h+1}^{z}\Psi_{[k]} + \sum_{\xi=\lambda+2}^{n_{[h]}}\psi_{[h],[\xi]}$, from Eqs. (6) and (7), we have

$$\begin{aligned}
F_{[h]}(\delta_{[h]}) - F_{[h]}(\delta'_{[h]}) &= \theta_{[h],o}\left(\psi_{[h],o} + \psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} + \theta_{[h],\chi}\left(\psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} \\
&\quad -\theta_{[h],\chi}\left(\psi_{[h],\chi} + \psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}} - \theta_{[h],o}\left(\psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}} \\
&= \theta_{[h],o}\left(\left(\psi_{[h],o} + \psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} - \left(\psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}}\right) \\
&\quad +\theta_{[h],\chi}\left(\left(\psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} - \left(\psi_{[h],\chi} + \psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}}\right) \\
&\leq \theta_{[h],\chi}\left(\left(\psi_{[h],o} + \psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} - \left(\psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}}\right) \\
&\quad +\theta_{[h],\chi}\left(\left(\psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} - \left(\psi_{[h],\chi} + \psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}}\right) \\
&= \theta_{[h],\chi}\left[\left(\psi_{[h],\chi} + Z\right)^{\frac{1}{\eta+1}} - \left(\psi_{[h],o} + Z\right)^{\frac{1}{\eta+1}}\right] \\
&\leq 0.
\end{aligned} \tag{8}$$

Hence, the optimal job sequence in $\widehat{G}_{[h]}$ is in non-decreasing order of $\theta_{[h],j}$ (or in non-increasing order of $\psi_{[h],j}$). □

Corollary 1 *For each $h = 1, 2, \ldots, z$, if $\psi_{h,\xi} = \psi_h$ for $\xi = 1, 2, \ldots, n_h$, the optimal job sequence in $\widehat{G}_h$ is in non-decreasing order of $\theta_{h,j}$.*

Corollary 2 *For each $h = 1, 2, \ldots, z$, if $\theta_{h,\xi} = \theta_h$ for $\xi = 1, 2, \ldots, n_h$, the optimal job sequence in $\widehat{G}_h$ is in non-increasing order of $\psi_{h,j}$.*

Similarly, we have

Lemma 2 (He et al. [36]). $\sum_{h=1}^{z} \Psi_{[h]} \left(\sum_{k=1}^{h} s_{[k]}\right)$ *is minimized if* $\frac{\Psi_{[1]}}{s_{[1]}} \geq \frac{\Psi_{[2]}}{s_{[2]}} \geq \ldots \geq \frac{\Psi_{[z]}}{s_{[z]}}$.

4 Solution algorithms for the general case

Under a special case (i.e., $n_h = \bar{n}$ and $\psi_{h,\xi} = \bar{\psi}$), Chen et al. [38] proved that

$$1\left|\widetilde{gt}, \widetilde{dif}, p^{Act}_{J_{h,j}} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}, n_h = \bar{n}, \psi_{h,\xi} = \bar{\psi}\right| \sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + \upsilon_{h,j}u_{h,j})$$

can be solved in $O(n^3)$ time. Below we will propose algorithms to solve the general case of

$$1\left|\widetilde{gt}, \widetilde{dif}, p^{Act}_{J_{h,j}} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}\right| \sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j} + \beta_{h,j}T_{h,j} + \gamma_{h,j}d_{h,j} + \upsilon_{h,j}u_{h,j}).$$

4.1 Solution of job sequence within each group

In this subsection, the optimal job sequence δ_h within group $\widehat{G}_h$ will be obtained. For group $\widehat{G}_h$, from Eq. (5) and the proof of Lemma 1, we only need to minimize

$$F_h = \sum_{j=1}^{n_h}\theta_{h,j}\left(\sum_{\xi=j}^{n_h}\psi_{h,\xi}\right)^{\frac{1}{\eta+1}}. \tag{9}$$

Let $\delta_h = (\delta_h^{sp}, \delta_h^{up})$ be a sequence of jobs within group $\widehat{G}_h$, where δ_h^{sp} (resp. δ_h^{up}) is the scheduled (resp. unscheduled part) part, and suppose there are g jobs in δ_h^{sp}, we have

$$F_h(\delta_h^{sp}, \delta_h^{up}) = \sum_{j=1}^{g}\theta_{h,[j]}\left(\sum_{\xi=j}^{n_h}\psi_{h,[\xi]}\right)^{\frac{1}{\eta+1}} + \sum_{j=g+1}^{n_h}\theta_{h,[j]}\left(\sum_{\xi=j}^{n_h}\psi_{h,[\xi]}\right)^{\frac{1}{\eta+1}}. \tag{10}$$

Observe that $\sum_{j=1}^{g}\theta_{h,[j]}\left(\sum_{\xi=j}^{n_h}\psi_{h,[\xi]}\right)^{\frac{1}{\eta+1}}$ in Eq. (10) is known and a lower bound for $F_h(\delta_h^{sp}, \delta_h^{up})$ is obtained by minimizing $\sum_{j=g+1}^{n_h}\theta_{h,[j]}\left(\sum_{\xi=j}^{n_h}\psi_{h,[\xi]}\right)^{\frac{1}{\eta+1}}$. From Lemma 1, we obtain the first lower bound ($\underline{\underline{LB}}$)

$$LB_1(F_h) = \sum_{j=1}^{g} \theta_{h,[j]} \left(\sum_{\xi=j}^{n_h} \psi_{h,[\xi]} \right)^{\frac{1}{\eta+1}} + \sum_{j=g+1}^{n_h} \theta_{h,(j)} \left(\sum_{\xi=j}^{n_h} \psi_{h,<\xi>} \right)^{\frac{1}{\eta+1}}, \quad (11)$$

where $\psi_{h,<g+1>} \geq \psi_{h,<g+2>} \geq \ldots \geq \psi_{h,<n_h>}$, $\theta_{h,(g+1)} \leq \theta_{h,(g+2)} \leq \ldots \leq \theta_{h,(n_h)}$ (note that $\psi_{h,<j>}$ and $\theta_{h,(j)}$ $(j = g+1, g+2, \ldots, n_h)$ do not necessarily correspond to the same job).

Similarly, let $\psi_{h,\min} = \min\{\psi_{h,j} | j \in \delta_h^{up}\}$, we obtain the second $\underline{\underline{LB}}$

$$LB_2(F_h) = \sum_{j=1}^{g} \theta_{h,[j]} \left(\sum_{\xi=j}^{n_h} \psi_{h,[\xi]} \right)^{\frac{1}{\eta+1}} + \sum_{j=g+1}^{n_h} \theta_{h,(j)} \left[(n_h - j + 1)\psi_{h,\min} \right]^{\frac{1}{\eta+1}}, \quad (12)$$

where $\theta_{h,(g+1)} \leq \theta_{h,(g+2)} \leq \ldots \leq \theta_{h,(n_h)}$.

Let $\theta_{h,\min} = \min\{\theta_{h,j} | j \in \delta_h^{up}\}$, we obtain the third $\underline{\underline{LB}}$

$$LB_3(F_h) = \sum_{j=1}^{g} \theta_{h,[j]} \left(\sum_{\xi=j}^{n_h} \psi_{h,[\xi]} \right)^{\frac{1}{\eta+1}} + \sum_{j=g+1}^{n_h} \theta_{h,\min} \left(\sum_{\xi=j}^{n_h} \psi_{h,<\xi>} \right)^{\frac{1}{\eta+1}}, \quad (13)$$

where $\psi_{h,<g+1>} \geq \psi_{h,<g+2>} \geq \ldots \geq \psi_{h,<n_h>}$.

In order to make the $\underline{\underline{LB}}$ tighter, the maximum value of expressions (11), (12) and (13) will be chosen as a $\underline{\underline{LB}}$ for $F_h(\delta_h^{sp}, \delta_h^{up})$, i.e.,

$$\underline{\underline{LB}}(F_h) = \max\{LB_1(F_h), LB_2(F_h), LB_3(F_h)\}. \quad (14)$$

From the above analysis and Framinan and Leisten [43], the following upper bound ($\underline{\underline{UP}}$) algorithm is proposed for sequence δ_h within $\widehat{G}_h$, i.e.,

Algorithm 1 *($\underline{\underline{UP}}$ for sequence δ_h within $\widehat{G}_h$)*

Phase 1

Step 1 Sequence jobs in non-decreasing order of $\theta_{h,j}$.
Step 2 Sequence jobs in non-increasing order of $\psi_{h,j}$.
Step 3 Sequence jobs in non-decreasing order of $\frac{\theta_{h,j}}{\psi_{h,j}}$.
Step 4 Choose the better solution from Steps 1, 2 and 3.

Phase 2

Step i Let δ_h^0 be the job sequence obtained from Phase 1.

Step ii Set $q = 2$. Select the first two jobs from the sorted list and select the better of the two possible sequences.

Step iii Increment q, $q = q + 1$. Select the qth job from the sorted list and insert it into q possible positions of the best partial sequence obtained so far. Among the q sequences, the best q-job partial sequence is selected based on minimum F_h (see Eq. (9)). Next, determine all possible sequences by interchanging jobs in positions x and y of the above partial sequence for all $1 \le x \le q$, $x < y \le q$. Select the best partial sequence among $\frac{q(q-1)}{2}$ sequences having minimum F_h (see Eq. (9)).

Step iv). If $q = n_h$, then STOP; otherwise, go to Step iii).

From $\underline{\underline{LB}}$ (14) and $\underline{\underline{UP}}$ (Algorithm 1), the following branch-and-bound (BB) algorithm is proposed to obtain the sequence δ_h within $\widehat{G}_h$:

Algorithm 2 ($\underline{\underline{BB}}$ *for sequence* δ_h *within* $\widehat{G}_h$, *denoted by* $BB_{\widehat{G}_h}$)

Step 1 (Find $\underline{\underline{UB}}$) Use **Phase 1** of Algorithm 1 to obtain an initial solution for the sub-problem of determining the optimal job sequence δ_h.

Step 2 The bounding and termination are the same as He et al. [36] ($\underline{\underline{LB}}$ is Eq. (14) and objective cost is Eq. (9)).

4.2 Solution of group sequence

From Subsection 4.1, we assume that the optimal job sequences within each group are given. Let $\varrho = (\varrho^{sp}, \varrho^{up})$ be a sequence of groups, where ϱ^{sp} (resp. ϱ^{up}) is scheduled (resp. unscheduled) part, and there are ς groups in ϱ^{sp}, from Eq. (4), one can achieve

$$F(\varrho^{sp}, \varrho^{up}) = \sum_{h=1}^{\varsigma} \Psi_{[h]} \left(\sum_{k=1}^{h} s_{[k]} \right) + \sum_{h=\varsigma+1}^{z} \Psi_{[h]} \left(\sum_{k=1}^{\varsigma} s_{[k]} + \sum_{k=\varsigma+1}^{h} s_{[k]} \right)$$
$$+(\eta^{\frac{1}{\eta+1}} + \eta^{\frac{-\eta}{\eta+1}}) \times \sum_{h=1}^{\varsigma} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{k=h+1}^{z} \Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]} \right)^{\frac{1}{\eta+1}}$$
$$+(\eta^{\frac{1}{\eta+1}} + \eta^{\frac{-\eta}{\eta+1}}) \times \sum_{h=\varsigma+1}^{z} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{k=h+1}^{z} \Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]} \right)^{\frac{1}{\eta+1}}. \quad (15)$$

From (15), $\sum_{k=1}^{\varsigma} s_{[k]}$, $\sum_{h=1}^{\varsigma} \Psi_{[h]} \left(\sum_{k=1}^{h} s_{[k]}\right)$ and $\sum_{h=1}^{\varsigma} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{k=1}^{z} \Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}}$ are constants, $\sum_{h=\varsigma+1}^{z} \Psi_{[h]} \left(\sum_{k=1}^{\varsigma} s_{[k]} + \sum_{k=\varsigma+1}^{h} s_{[k]}\right)$ can be minimized by Lemma 2,

$\sum_{h=\varsigma+1}^{z} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{k=h+1}^{z} \Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}} \ge \sum_{h=\varsigma+1}^{z} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]}\right)^{\frac{1}{\eta+1}}$. Hence, we have the following lower bound:

$$\underline{\underline{LB}} = \sum_{h=1}^{\varsigma} \Psi_{[h]} \left(\sum_{k=1}^{h} s_{[k]} \right) + \sum_{h=\varsigma+1}^{z} \Psi_{<h>} \left(\sum_{k=1}^{\varsigma} s_{[k]} + \sum_{k=\varsigma+1}^{h} s_{<k>} \right)$$
$$+ (\eta^{\frac{1}{\eta+1}} + \eta^{\frac{-\eta}{\eta+1}}) \times \sum_{h=1}^{\varsigma} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{k=h+1}^{z} \Psi_{[k]} + \sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]} \right)^{\frac{1}{\eta+1}}$$
$$+ (\eta^{\frac{1}{\eta+1}} + \eta^{\frac{-\eta}{\eta+1}}) \times \sum_{h=\varsigma+1}^{z} \sum_{j=1}^{n_{[h]}} \theta_{[h],[j]} \left(\sum_{\xi=j}^{n_{[h]}} \psi_{[h],[\xi]} \right)^{\frac{1}{\eta+1}}, \quad (16)$$

where $\frac{\Psi_{<\varsigma+1>}}{s_{<\varsigma+1>}} \geq \frac{\Psi_{<\varsigma+2>}}{s_{<\varsigma+2>}} \geq \ldots \geq \frac{\Psi_{<z>}}{s_{<z>}}$.

Similarly, the following $\underline{\underline{UB}}$ algorithm for group sequence ϱ is:

Algorithm 3 (*$\underline{\underline{UB}}$ for group sequence ϱ*)

Phase 1

Step 1 Sequence groups in non-decreasing order of s_h.
Step 2 Sequence groups in non-increasing order of $\frac{\Psi_h}{s_h}$.
Step 3 Sequence groups in non-increasing order of Ψ_h.
Step 4 Choose the better solution from Steps 1, 2 and 3.

Phase 2

Step i Let ϱ^0 be the group sequence obtained from Phase 1.

Step ii). Set $l = 2$. Select the first two groups from the sorted list and select the better of the two possible sequences.

Step iii). Increment l, $l = l + 1$. Select the lth group from the sorted list and insert it into l possible positions of the best partial sequence obtained so far. Among the l sequences, the best l-job partial sequence is selected based on minimum F (see Eq. (5)). Next, determine all possible sequences by interchanging groups in positions x and y of the above partial sequence for all $1 \leq x \leq l, x < y \leq l$. Select the best partial sequence among $\frac{l(l-1)}{2}$ sequences having minimum F (see Eq. (5)).

Step iv). If $l = z$, then STOP; otherwise, go to Step iii).

From $\underline{\underline{LB}}$ (16) and $\underline{\underline{UB}}$ (Algorithm 3), the following BB algorithm is proposed to obtain the optimal group sequence ϱ:

Algorithm 4 (*BB for group sequence ϱ, denoted by BB_ϱ*)

Step 1. (Find $\underline{\underline{UB}}$) Use **Phase 1** of Algorithm 3 to obtain an initial solution for the sub-problem of determining the optimal group sequence ϱ.

Step 2. The bounding and termination are the same as He et al. [36] ($\underline{\underline{LB}}$ is Eq. (16) and objective cost is Eq. (4)).

4.3 Algorithms

From Subsections 4.1-4.2, and Li et al. [44], the general problem

$$1\left|\widetilde{gt}, \widetilde{dif}, p^{Act}_{J_{h,j}} = \left(\frac{\varpi_{h,j}}{u_{h,j}}\right)^{\eta}\right|\sum_{h=1}^{z}\sum_{j=1}^{n_h}(\alpha_{h,j}E_{h,j}+\beta_{h,j}T_{h,j}+\gamma_{h,j}d_{h,j}+\upsilon_{h,j}u_{h,j})$$

is solved optimally by:

Algorithm 5 (*Exact algorithm based on BB*)
Step 1 For each group $\widehat{G}_h$, calculate the optimal job sequence by using Algorithm 2, $h = 1, 2, \ldots, z$.
Step 2 Calculate the optimal group sequence by using Algorithm 4.
Since Algorithm 5 is based on BB, hence we propose the following heuristic algorithm:

Algorithm 6 (*Heuristic algorithm*)
Step 1 For each group $\widehat{G}_h$, calculate the local optimal job sequence by using Algorithm 1, $h = 1, 2, \ldots, z$.
Step 2 Calculate the local optimal group sequence by using Algorithm 3.

5 Number study

The heuristic (i.e., Algorithm 6) and the exact algorithm (i.e., BB, Algorithm 5) were programmed in C++ (carried out on CPU Interl core i5-8250U 1.4GHz PC with 8.00GB RAM), where $n = 50, 60, 70, 80$ and z= 8, 9, 10, 11, 12, and $n_h \geq 1$. The parameters setting is given as follows:

(1) $s_h, \alpha_{h,j}, \beta_{h,j}, \gamma_{h,j}$ and $\upsilon_{h,j}$ were drawn from a discrete uniform distribution in [1, 49];
(2) $\varpi_{h,j}$ were drawn from a discrete uniform distribution in [1, 49], [50, 99], and [1, 99], i.e., $\varpi_{h,j} \in [1, 49]$, $\varpi_{h,j} \in [50, 99]$, and $\varpi_{h,j} \in [1, 99]$;
(3) $\eta = 1, 1.5, 2, 2.5$.

For simulation accuracy, each random instance was conducted 20 times, and the total number of instances is $4 \times 5 \times 3 \times 4 \times 20$ =4800. The error of Algorithm 6 is calculated as

$$\frac{F(H)}{F^*}, \tag{17}$$

where $F(H)$ (resp. F^*) is the objective value (see Eq. (4)) generated by Algorithm 6 (resp. Algorithm 5).

On the other hand, running time (i.e., ms (millisecond)) of Algorithms 5 and 6 is defined. All of the experimental minimum CPU value, maximum CPU value and average CPU value can easily show that Algorithm 6 is more efficient than Algorithm 5 statistically. From Tables 1, 2 and 3, the maximum error of Algorithm 6 is less than

Table 1 Results for $\varpi_{h,j} \in [1, 49]$ (CPU time is ms)

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
50×8	1	134	190.15	258	3	4.15	7	1.0158	1.0189	1.024
	1.5	95	162.05	235	3	3.75	5	1.0141	1.0177	1.0229
	2	103	179.05	406	3	4	6	1.0118	1.0177	1.0215
	2.5	80	180.75	389	2	3.6	7	1.0107	1.0165	1.0316
50×9	1	306	1028.8	3369	3	4.05	6	1.005	1.0123	1.0209
	1.5	471	1330.25	3435	3	4.2	5	1.0074	1.0115	1.0178
	2	372	956.85	4066	3	4.45	6	1.0068	1.0157	1.0295
	2.5	368	870.1	2128	3	3.95	6	1.0084	1.0118	1.0183
50×10	1	1069	4369.3	9296	3	4.45	6	1.004	1.01	1.017
	1.5	1070	4114.25	9434	3	4.3	6	1.0056	1.0124	1.0183
	2	2028	5001.5	10777	4	4.55	6	1.0069	1.012	1.023
	2.5	780	4673.2	14644	4	4.8	6	1.0068	1.0119	1.0169
50×11	1	7693	19859.45	43868	3	4.65	6	1.0052	1.0087	1.0166
	1.5	3612	32700.7	127196	3	5.2	7	1.0078	1.0113	1.021
	2	4462	40478.85	262085	3	4.6	6	1.0053	1.0102	1.016
	2.5	7691	38886.75	111708	3	5.1	6	1.0057	1.0105	1.0179
50×12	1	6757	72495.1	404048	4	5.55	7	1.0038	1.0088	1.01711
	1.5	28723	172688.9	1002176	4	5.35	7	1.0036	1.0077	1.0123
	2	16331	99993.1	404483	4	5	6	1.0053	1.0084	1.01388
	2.5	16537	140580.7	379872	4	4.95	7	1.0031	1.0085	1.0168
60×8	1	121	230.4	1142	3	3.4	5	1.01345	1.01411	1.0196
	1.5	86	165.65	357	3	4	6	1.012	1.02	1.03
	2	87	600.3	3059	3	4.1	10	1.0086	1.0187	1.0324
	2.5	111	322.55	1338	2	3.8	7	1.0097	1.0191	1.0275

Table 1 continued

		CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
$n \times z$	η	min	avg	max	min	avg	max	min	avg	max
60×9	1	299	971.4	1681	3	4	5	1.0083	1.0133	1.0248
	1.5	761	1514.95	3321	3	3.85	5	1.0097	1.015	1.025
	2	354	1070.9	2403	3	4.45	7	1.0068	1.0125	1.019
	2.5	477	1384.9	2894	3	4.2	6	1.0084	1.0129	1.0238
60×10	1	1750	7364	18892	3	4.45	7	1.0045	1.01	1.0178
	1.5	2164	9238.35	20442	4	5	7	1.0062	1.0124	1.0188
	2	1765	7556.8	21621	4	4.85	7	1.0068	1.0121	1.0199
	2.5	2029	12389.1	52191	3	4.3	6	1.0079	1.0116	1.016
60×11	1	8215	28968.55	101123	4	4.85	7	1.0068	1.01	1.0161
	1.5	8073	38395.3	109120	3	5.15	7	1.0069	1.01	1.0169
	2	4549	53896.4	131455	4	4.85	6	1.0079	1.0132	1.0292
	2.5	8411	38985.35	105403	4	5.1	6	1.006	1.012	1.0187
60×12	1	10053	271025	1187444	5	5.7	7	1.004	1.0089	1.0138
	1.5	15114	260221.2	1149283	5	5.7	7	1.0046	1.0088	1.0161
	2	12460	191216.9	960848	4	5.65	8	1.0048	1.0095	1.0185
	2.5	7251	139583.6	566950	4	5.8	7	1.0047	1.0079	1.0134
70×8	1	166	544.05	1420	3	4.5	6	1.0108	1.0151	1.0195
	1.5	264	11167.05	60016	4	4.3	5	1.011	1.0179	1.031
	2	304	734.25	1903	3	4.2	6	1.011	1.0169	1.0243
	2.5	199	473.35	1051	3	4.05	6	1.0139	1.0171	1.0304
70×9	1	608	2079.7	4339	3	4.6	7	1.0093	1.0129	1.0169
	1.5	917	2057.6	4201	4	4.4	5	1.0072	1.0153	1.0244
	2	717	2526.45	6536	3	4.25	6	1.0076	1.0147	1.0287
	2.5	570	1938.15	4906	4	4.75	7	1.0088	1.0141	1.0227

Table 1 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
70×10	1	3349	13188.6	46480	4	4.9	6	1.0077	1.0126	1.0198
	1.5	2412	11314.65	35631	4	5	7	1.0083	1.0126	1.0226
	2	3928	12420.45	30874	4	4.85	6	1.007	1.0121	1.0173
	2.5	1670	11612.75	25231	3	5.05	8	1.0086	1.0128	1.0195
70×11	1	9106	53797.2	158383	4	5.56	7	1.0065	1.0104	1.0143
	1.5	16944	81038.95	433261	4	5.65	8	1.005	1.0107	1.0155
	2	13147	63877.4	178599	5	5.4	6	1.0061	1.0112	1.0204
	2.5	16973	56586.25	141124	4	5.25	6	1.0038	1.0101	1.0185
70×12	1	17113	250287.5	1048291	5	6.45	8	1.0052	1.0097	1.0163
	1.5	82618	287697.9	576649	5	5.75	7	1.0055	1.0104	1.0163
	2	45528	280837.5	750018	4	6.15	8	1.0056	1.0099	1.0204
	2.5	52218	318374.9	1883449	4	5.9	8	1.0046	1.0088	1.0148
80×8	1	306	1179.6	3670	4	4.6	6	1.0113	1.0196	1.0332
	1.5	461	2823.15	13043	3	5.2	8	1.0079	1.0174	1.0246
	2	387	2046.1	5411	4	4.5	6	1.0105	1.0185	1.03
	2.5	429	5425.45	27229	3	4.5	6	1.0117	1.0186	1.0263
80×9	1	1009	2427.1	4947	3	4.65	6	1.0109	1.016	1.023
	1.5	1094	2967.2	7527	4	5.4	7	1.0097	1.0152	1.0235
	2	930	2267.4	3780	4	4.4	6	1.0085	1.0133	1.0186
	2.5	953	2879.1	6975	3	4.8	7	1.0099	1.0161	1.0287

Table 1 continued

		CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
$n \times z$	η	min	avg	max	min	avg	max	min	avg	max
80×10	1	2105	12631.35	38418	4	4.95	6	1.01	1.0139	1.0215
	1.5	6867	18692.95	77059	4	5.6	7	1.009	1.0144	1.0296
	2	2753	13126.15	32883	4	4.9	6	1.009	1.0142	1.021
	2.5	3619	12462.2	29280	4	5.45	7	1.008	1.013	1.022
80×11	1	20069	101191.2	282346	5	5.85	7	1.0069	1.0117	1.0181
	1.5	27892	75108.25	180941	5	6.1	9	1.0055	1.0112	1.0142
	2	22130	87616.25	206565	5	5.6	7	1.0063	1.0118	1.0164
	2.5	17336	98536.15	365413	5	6	8	1.007	1.0103	1.0152
80×12	1	50713	727708.5	2679963	5	6.25	8	1.0065	1.0105	1.0149
	1.5	165046	577967.9	1316874	5	6.15	7	1.0064	1.0101	1.0165
	2	84254	605795.5	1662153	5	6.45	8	1.0054	1.0092	1.0194
	2.5	82050	544184.3	2549986	5	6.7	8	1.0045	1.0083	1.0135

Table 2 Results for $\varpi_{h,j} \in [50, 99]$ (CPU time is ms)

		CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
$n \times z$	η	min	avg	max	min	avg	*max*	min	avg	max
50×8	1	236	339.2	466	3	3.5	4	1.018	1.022	1.0284
	1.5	209	461.9	879	3	3.6	5	1.015	1.024	1.0376
	2	242	365	757	3	3.8	5	1.011	1.0174	1.0234
	2.5	300	479.9	857	3	3.75	6	1.0133	1.021	1.036
50×9	1	638	2372.6	5393	3	4.2	6	1.01	1.0169	1.0267
	1.5	1742	3387.55	5929	3	3.65	5	1.0122	1.02065	1.031
	2	1120	3437.3	6593	3	4	5	1.0087	1.0163	1.0286
	2.5	1980	3535.65	5974	3	4.15	6	1.01	1.0178	1.0251
50×10	1	2832	21079.55	50746	3	4.4	6	1.0089	1.0147	1.03
	1.5	7377	20559.36	43597	3	4.35	6	1.0064	1.0157	1.024
	2	6409	23244.7	63038	4	4.55	7	1.0084	1.0159	1.0229
	2.5	6151	21708.85	48139	3	4.35	6	1.0069	1.0134	1.022
50×11	1	26486	87049.55	422566	4	5.15	7	1.0071	1.0122	1.019
	1.5	38919	118425	277892	4	4.55	6	1.009	1.0169	1.033
	2	53618	144738.4	633835	3	4.85	6	1.0079	1.0155	1.02559
	2.5	26477	165528.2	366644	3	4.9	6	1.0072	1.0155	1.0233
50×12	1	47658	399162.9	1350157	4	4.9	7	1.0065	1.0123	1.0263
	1.5	134802	983183.9	2654578	4	5.15	6	1.0077	1.0122	1.0163
	2	223340	922976.3	3024068	4	5.15	6	1.0079	1.0135	1.023
	2.5	147538	820849.3	2516306	3	5.1	7	1.0058	1.0125	1.02

Table 2 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
60×8	1	408	713.55	1294	3	4.85	21	1.0162	1.0206	1.0334
	1.5	350	620.05	1421	2	3.75	6	1.0152	1.0236	1.0321
	2	378	721.9	1398	3	3.9	5	1.0133	1.0208	1.0263
	2.5	377	734.1	1215	3	3.6	5	1.0112	1.02	1.03
60×9	1	2025	3838.75	6716	3	4.2	5	1.0098	1.0171	1.0293
	1.5	1748	4660.1	9158	3	4.05	5	1.0097	1.0181	1.0254
	2	2067	5776.15	8555	3	4.15	6	1.0106	1.0196	1.035
	2.5	2167	4724.06	10044	3	4.15	5	1.0112	1.0188	1.03
60×10	1	11078	27338.95	49933	3	4.9	7	1.0085	1.0159	1.0249
	1.5	11068	29975.33	62065	3	4.4	6	1.0093	1.0171	1.0323
	2	6634	33184.93	61801	3	4.667	6	1.0108	1.0159	1.0282
	2.5	17341	32891.4	56783	4	4.73	6	1.0099	1.0161	1.0208
60×11	1	33902	169762.8	375938	4	5.3	7	1.0061	1.0123	1.0222
	1.5	46141	198742.1	442412	4	5.33	7	1.0098	1.0172	1.0273
	2	94611	245046.6	444343	4	4.6	6	1.0076	1.0166	1.0254
	2.5	144104	244836.5	412154	4	4.75	6	1.005	1.0121	1.0182
60×12	1	430414	999665	1794152	5	5.6	6	1.0086	1.0109	1.0117
	1.5	559451	1210922	2007498	6	6.28	7	1.0059	1.01	1.0165
	2	707871	1406702	3828633	5	5.75	7	1.0068	1.0128	1.0291
	2.5	431033	2334331	6124265	5	5.46	7	1.0069	1.0126	1.0239

Table 2 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
70×8	1	468	1057	1921	3	4.4	6	1.0122	1.0229	1.0416
	1.5	717	1523.65	8750	3	4.05	6	1.0185	1.0232	1.0449
	2	531	1183.6	2803	3	4.25	6	1.015	1.024	1.0336
	2.5	447	1217.2	2349	3	4.15	5	1.0136	1.0234	1.0435
70×9	1	2214	6135.45	10106	4	4.75	6	1.012	1.0188	1.0245
	1.5	1958	6102.25	10596	3	4.4	6	1.0124	1.0218	1.0338
	2	3285	7666	13808	3	4.45	6	1.0123	1.0194	1.028
	2.5	1885	5719.35	10044	3	4.4	6	1.0101	1.0182	1.0293
70×10	1	16603	33650.67	51704	4	4.8	7	1.0084	1.0163	1.0247
	1.5	20491	43754.33	96892	4	5.067	7	1.0105	1.0168	1.0283
	2	26353	53224.93	79238	4	5.133	7	1.0115	1.0171	1.0227
	2.5	29321	55588.13	88434	4	5.2	6	1.0101	1.0165	1.0264
70×11	1	48155	221539.2	551882	4	5.33	6	1.0102	1.0157	1.0303
	1.5	86751	268650.6	653949	4	5.53	8	1.008	1.0143	1.0197
	2	103859	396226.5	909673	4	5.3	7	1.0071	1.0151	1.026
	2.5	121172	294530.2	564262	5	5.88	7	1.0063	1.0143	1.0264
70×12	1	519357	1505556	2811045	5	6.5	8	1.0089	1.012	1.0183
	1.5	940305	27992026	5736226	5	6	7	1.0104	1.0135	1.0228
	2	640888	2296009	4739770	5	6.25	8	1.0074	1.0151	1.0259
	2.5	949530	2003256	3001798	5	6.5	7	1.0085	1.0141	1.0164

Table 2 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
80×8	1	600	1609.75	4251	4	4.6	6	1.0184	1.0251	1.0384
	1.5	726	2973.85	15569	4	4.45	6	1.015	1.025	1.047
	2	880	2585.2	7121	4	4.85	7	1.012	1.023	1.0335
	2.5	967	2899.15	8757	4	4.54	6	1.0126	1.0214	1.0348
80×9	1	2326	7489.1	13554	3	5.2	7	1.0144	1.021	1.033
	1.5	3588	7663.52	12459	3	4.8	6	1.0152	1.0225	1.0311
	2	3993	8445.2	14992	4	4.8	7	1.0114	1.0211	1.03
	2.5	4215	7676.75	12188	4	4.85	6	1.0087	1.021	1.0301
80×10	1	16459	51486.35	94849	4	5.2	7	1.012	1.0183	1.0308
	1.5	20424	68806.95	152280	5	5.45	7	1.0095	1.0187	1.0337
	2	25822	63493.5	128761	5	5.55	7	1.0123	1.0178	1.0241
	2.5	18885	53050.3	101402	4	5.05	6	1.0104	1.0183	1.0288
80×11	1	103077	279373.2	720776	5	5.9	7	1.008	1.0151	1.0219
	1.5	110106	403694.6	840158	5	5.85	8	1.009	1.0142	1.0187
	2	144312	388154.4	1226044	5	5.7	7	1.0093	1.0137	1.02
	2.5	152568	443496.6	898563	4	5.8	7	1.008	1.017	1.0243
80×12	1	706839	3474563	8450764	5	6.78	9	1.0084	1.0129	1.019
	1.5	1880313	3624009	8595039	5	6.5	8	1.0134	1.0162	1.0206
	2	918442	3361686	6611007	5	5.6	6	1.0091	1.0124	1.018
	2.5	816915	3680101	17646047	5	6.4	8	1.0057	1.0133	1.0211

Table 3 Results for $\varpi_{h,j} \in [1, 99]$ (CPU time is ms)

		CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
$n \times z$	η	min	avg	max	min	avg	max	min	avg	max
50×8	1	203	254.15	454	3	3.75	6	1.0168	1.0214	1.0315
	1.5	198	304.55	671	3	3.85	6	1.0163	1.0216	1.028
	2	141	343.6	703	3	3.9	5	1.0135	1.0225	1.031
	2.5	156	601	310.7	3	3.6	4	1.0096	1.0129	1.0174
50×9	1	406	1665.2	5232	3	4.05	6	1.0088	1.0178	1.0322
	1.5	893	2429.15	4397	3	4	5	1.0084	1.0181	1.0316
	2	579	2319.9	5544	3	4.4	5	1.0102	1.0189	1.027
	2.5	574	2338.55	7895	3	4.2	10	1.0094	1.0196	1.0419
50×10	1	2639	10203.7	34174	4	4.5	5	1.0055	1.0121	1.0183
	1.5	2509	13144.65	49627	4	4.55	6	1.0078	1.0147	1.0311
	2	3639	15562.15	46460	4	4.6	6	1.0076	1.0162	1.0343
	2.5	4092	12740.75	25529	3	4.65	7	1.0068	1.0152	1.0236
50×11	1	11024	55704.9	321090	4	4.6	6	1.0069	1.0107	1.0153
	1.5	9120	96775.05	406453	4	5.25	7	1.007	1.0137	1.0223
	2	13503	85187.65	328148	4	4.7	6	1.0091	1.0176	1.0433
	2.5	11069	86710.75	248032	3	4.6	6	1.007	1.0134	1.0218
50×12	1	19396	197529.4	876852	4	5.3	6	1.007	1.0108	1.0175
	1.5	68757	398574.4	1154778	4	5.5	7	1.0061	1.0135	1.035
	2	43237	483312.5	1726204	4	5.25	6	1.007	1.0133	1.02
	2.5	68669	499551.1	1844744	4	5.35	6	1.0042	1.0127	1.024
60×8	1	217	565.9	1139	3	3.7	5	1.0116	1.017	1.024
	1.5	291	702.05	1160	3	3.85	5	1.0068	1.0206	1.0381
	2	305	617.35	1127	2	3.35	5	1.0132	1.019	1.0328
	2.5	162	437.8	759	3	4.05	6	1.0112	1.0198	1.0327

Table 3 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
60×9	1	1018	2992.25	9873	3	4	5	1.0096	1.0182	1.0314
	1.5	636	3220.1	5963	3	4.35	6	1.0118	1.0185	1.0326
	2	1386	3156.3	7436	3	3.9	5	1.0092	1.0188	1.0265
	2.5	1000	3357.2	6833	3	4.4	6	1.0089	1.0186	1.0338
60×10	1	3758	13873.3	28251	3	4.75	7	1.0088	1.0154	1.0278
	1.5	6049	18000.65	45507	3	4.45	6	1.0089	1.0147	1.024
	2	6379	23613.75	71327	3	4.5	6	1.009	1.0166	1.0289
	2.5	5929	18637.35	75785	4	4.7	6	1.009	1.0171	1.02636
60×11	1	18040	100656.5	221738	4	5.65	8	1.0098	1.0135	1.0205
	1.5	28661	151533.6	238304	4	5.2	7	1.0069	1.0146	1.0262
	2	22210	119782.9	255309	4	5.15	6	1.0067	1.0157	1.0307
	2.5	35106	116321.6	275801	4	5.23	7	1.0054	1.0121	1.0198
60×12	1	158168	653504.6	2419968	5	6.125	7	1.0066	1.0119	1.0177
	1.5	118569	718514.7	1914416	5	5.75	7	1.0063	1.0126	1.0237
	2	149129	644861.4	1170100	5	6.14	8	1.0107	1.0123	1.0147
	2.5	187475	457226.1	644078	5	5.71	7	1.008	1.0167	1.0287
70×8	1	262	742.95	1316	3	4.2	6	1.012	1.021	1.028
	1.5	327	1077.75	2690	3	4.05	5	1.0147	1.0228	1.0332
	2	503	1601.65	5980	3	4	5	1.0132	1.0216	1.0275
	2.5	472	1803.85	11715	3	4.25	6	1.0173	1.0228	1.0311
70×9	1	1466	3054.4	6417	3	4.3	5	1.0093	1.0175	1.025
	1.5	1810	3939.5	11285	3	4.35	5	1.0105	1.0185	1.0272
	2	1974	4446.95	8814	3	4.3	6	1.01248	1.0208	1.0309
	2.5	1444	3417.2	7018	3	4.3	6	1.0086	1.0211	1.0406

Table 3 continued

		CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
$n \times z$	η	min	avg	max	min	avg	max	min	avg	max
70×10	1	10729	26561.6	60751	4	4.85	6	1.0101	1.0151	1.0247
	1.5	7508	25824.67	52396	4	5.2	7	1.0105	1.0172	1.0262
	2	6946	29022.93	55285	4	4.93	6	1.0095	1.0171	1.0266
	2.5	10283	27891.73	55607	4	5.4	7	1.0101	1.0164	1.0243
70×11	1	46317	120194.5	314601	4	5.3	7	1.0064	1.0128	1.0216
	1.5	31903	183299.1	419051	5	5.53	7	1.008	1.0168	1.0307
	2	47868	252638.3	647458	5	5.3	6	1.0042	1.0131	1.0184
	2.5	25154	246402.3	646908	5	5.67	7	1.0107	1.0177	1.0293
70×12	1	92761	1178181	2068135	6	7.5	19	1.0078	1.0124	1.0218
	1.5	207332	928290.5	2348558	5	5.86	7	1.0085	1.0135	1.0214
	2	368157	1011945	3134826	5	6	7	1.0073	1.0122	1.0166
	2.5	210347	1111415	2251459	5	6.8	8	1.0098	1.013	1.0178
80×8	1	475	1358.7	3114	4	4.6	6	1.0135	1.0236	1.0329
	1.5	994	2675.15	6359	3	4.55	5	1.0124	1.0259	1.0375
	2	730	3996.6	15442	4	4.55	7	1.0121	1.0265	1.0505
	2.5	1163	4360.55	17687	4	4.5	6	1.0129	1.0241	1.0331

Table 3 continued

$n \times z$	η	CPU of Algorithm 5			CPU of Algorithm 6			Error of Algorithm 6		
		min	avg	max	min	avg	max	min	avg	max
80×9	1	1557	4810.45	12269	4	4.6	6	1.0118	1.0201	1.0303
	1.5	2400	4926.05	8579	4	4.7	6	1.0131	1.0201	1.0304
	2	1807	5352.05	10383	4	4.75	6	1.0145	1.0223	1.0364
	2.5	2838	5380.2	15893	4	4.45	6	1.0132	1.0185	1.0235
80×10	1	4080	28649.75	75836	4	5.25	7	1.0088	1.0166	1.0274
	1.5	13655	42239.7	78561	4	5.35	7	1.0095	1.0185	1.0328
	2	17830	33495.85	70261	4	5.2	7	1.0086	1.0169	1.029
	2.5	12890	38265.35	90567	4	5.45	7	1.0098	1.0159	1.0222
80×11	1	72115	188644.1	446228	5	5.8	7	1.0096	1.0142	1.0192
	1.5	56500	239459.5	539906	5	5.54	6	1.0103	1.0139	1.0199
	2	68358	237860.5	684447	5	5.7	7	1.0088	1.0153	1.0395
	2.5	63646	262479.5	780078	5	5.78	7	1.0098	1.0146	1.0208
80×12	1	636557	1843826	3664838	6	7	8	1.0088	1.0132	1.0155
	1.5	676976	1609839	2701869	5	7	10	1.0096	1.0139	1.0174
	2	412298	1330080	3321076	5	6.5	8	1.0084	1.0137	1.0217
	2.5	426513	1807124	4950812	5	6.9	10	1.009	1.0125	1.02049

Table 4 Calculated $t-$values for the hypothesis tests

$n \times z$	η	t
50×12	1	2.713
50×12	1.5	2.848
50×12	2	2.915
50×12	2.5	2.755
60×12	1	2.847
60×12	1.5	2.817
60×12	2	2.882
60×12	2.5	2.831
70×12	1	2.719
70×12	1.5	2.898
70×12	2	2.884
70×12	2.5	2.824
80×12	1	2.724
80×12	1.5	2.808
80×12	2	2.686
80×12	2.5	2.778

1.0505 for $n \times z \leq 80 \times 12$ and the results of $\varpi_{h,j} \in [1, 49]$ is more accurate than $\varpi_{h,j} \in [50, 99]$ and $\varpi_{h,j} \in [1, 99]$.

As the results in Table 1-3 show that Algorithm 6 could be more accurate in the case of $\varpi_{h,j} \in [1, 49]$ than $\varpi_{h,j} \in [50, 99]$ and $\varpi_{h,j} \in [1, 99]$, statistical hypothesis tests are implemented to compare the effectiveness of Algorithm 6 in the case of $Case1 : \varpi_{h,j} \in [1, 49]$ and $Case2 : \varpi_{h,j} \in [50, 99]$ for representativeness in Table 4. For a display, the instances where $\eta = 1, 1.5, 2, 2.5$,$n = 50, 60, 70, 80$ and z= 12 are considered. The $t-$test is used for the tests: $t = \frac{\overline{X_{Case2}} - \overline{X_{Case1}}}{S_w \sqrt{1/m_{Case2} + 1/m_{Case1}}}$, where $S_w^2 = \frac{(m_{Case2}-1)S_{Case2}^2 + (m_{Case1}-1)S_{Case1}^2}{m_{Case2}+m_{Case1}-2}$ and $\overline{X}$ denotes the mean error. The corresponding statistical hypothesis test is configured as $H_0 : \mu_{Case2} > \mu_{Case1}, H_1 : \mu_{Case2} \leq \mu_{Case1}$. Type I error of 1% is used and $t_{critical} = 2.5$. Experiment results in Table 4 show that the hypothesis that $H_0 : \mu_{Case2} > \mu_{Case1}$ with a type I error of 1% cannot be rejected statistically.

6 Conclusions

This paper studied the group scheduling with resource allocation, under single machine and $\widetilde{dif}$ assignment, the goal is to minimize the weighted sum of the earliness-tardiness cost, due date assignment cost and the resource consumption cost. For the general problem, the heuristic and BB algorithms were proposed. The experimental simulations showed that the BB algorithm is able to obtain an optimal solution with less than or equal to 80 × 12 jobs in a reasonable time (maximum CPU time is 17646047 ms), and the error of the heuristic algorithm can be within the reasonable range (maximum error bound is 1.047). Challenging further research can deal with the extensions of this model to the flow shop setting (see Wang and Wang [45], Liu et al. [46], Sun et al. [47], and Lv and Wang [48]), study the $\widetilde{gt}$ scheduling with non-regular objective functions (e.g. due-window assignment, Lin [49], Mao et al. [50], Lv et al. [51], and Zhang et al. [52]), or consider other $\widetilde{gt}$ scheduling with deteriorating jobs (see Gawiejnowicz [53], Lv et al. [54], Mao et al. [55], and Ma et al. [56]).

Acknowledgements This research was supported by the National Natural Science Regional Foundation of China (72061029 and 71861031).

Data Availability The corresponding author will provide the relevant datasets upon request.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Shabtay, D., Steiner, G.: A survey of scheduling with controllable processing times. Discret. Appl. Math. **155**, 1643–1666 (2007)
2. Yedidsiona, L., Shabtay, D.: The resource dependent assignment problem with a convex agent cost function. Eur. J. Oper. Res. **261**, 486–502 (2017)
3. Sun, X.Y., Geng, X.-N., Wang, J.-B., Liu, F.: Convex resource allocation scheduling in the no-wait flowshop with common flow allowance and learning effect. Int. J. Prod. Res. **57**, 1873–1891 (2019)
4. Kovalev, S., Chalamon, I., Bécuwe, A.: Single machine scheduling with resource constraints: equivalence to two-machine flow-shop scheduling for regular objectives. J. Oper. Res. Soc. **75**, 1343–1346 (2023)
5. Zhao, S.: Resource allocation flowshop scheduling with learning effect and slack due window assignment. J. Ind. Manag. Optim. **17**, 2817–2835 (2021)
6. 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. **53**, 1273–1281 (2021)
7. Mor, B., Shabtay, D., Yedidsion, L.: Heuristic algorithms for solving a set of NP-hard single-machine scheduling problems with resource-dependent processing times. Comput. Ind. Eng. **153**, 107024 (2021)
8. Tian, Y.: Single-machine due-window assignment scheduling with resource allocation and generalized earliness/tardiness penalties. Asia-Pacific J. Oper. Res. **39**, 2150041 (2022)
9. Wang, Y.-C., Wang, J.-B.: Study on convex resource allocation scheduling with a time-dependent learning effect. Mathematics **11**, 3179 (2023)
10. Zhang, L.-H., Lv, D.-Y., Wang, J.-B.: Two-agent slack due-date assignment scheduling with resource allocations and deteriorating jobs. Mathematics **11**, 2737 (2023)
11. Li, M.-H., Lv, D.-Y., Lv, Z.-G., Zhang, L.-H., Wang, J.-B.: A two-agent resource allocation scheduling problem with slack due-date assignment and general deterioration function. Comput. Appl. Math. **43**, 229 (2024)
12. 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-Pacific J. Oper. Res. **41**, 2350026 (2024)
13. 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. **70**, 1965–1994 (2024)
14. Sun, X.Y., 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. **20**, 15–35 (2024)
15. Zhang, Y., Sun, X.-Y., 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. (2024). https://doi.org/10.1080/01605682.2024.2371527

16. Potts, C.N., Van Wassenhove, L.N.: Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. J. Oper. Res. Soc. **43**, 395–406 (1992)
17. Webster, S., Baker, K.R.: Scheduling groups of jobs on a single machine. Oper. Res. **43**, 692–703 (1995)
18. Wu, C.-C., Lee, W.-C.: Single-machine group-scheduling problems with deteriorating setup times and job-processing times. Int. J. Prod. Econ. **115**, 128–133 (2008)
19. Li, S.S., Ng, C.T., Yuan, J.J.: Group scheduling and due date assignment on a single machine. Int. J. Prod. Econ. **130**, 230–235 (2011)
20. Ji, M., Chen, K., Ge, J., Cheng, T.C.E.: Group scheduling and job-dependent due window assignment based on a common flow allowance. Comput. Ind. Eng. **68**, 35–41 (2014)
21. Ji, M., Chen, K., Ge, J., Cheng, T.C.E.: Group scheduling with group-dependent multiple due windows assignment. Int. J. Prod. Res. **54**, 1244–1256 (2014)
22. Zhang, X., Liao, L., Zhang, W., Cheng, T.C.E., Tan, Y., Ji, M.: Single-machine group scheduling with new models of position-dependent processing times. Comput. Ind. Eng. **117**, 1–5 (2018)
23. Huang, X.: Bicriterion scheduling with group technology and deterioration effect. J. Appl. Math. Comput. **60**, 455–464 (2019)
24. Liu, F., Yang, J., Lu, Y.-Y.: Solution algorithms for single-machine group scheduling with ready times and deteriorating jobs. Eng. Optim. **51**, 862–874 (2019)
25. Bajwa, N., Melouk, S., Bryant, A.: A hybrid heuristic approach to minimize number of tardy jobs in group technology systems. Int. Trans. Oper. Res. **26**, 1847–1867 (2019)
26. Xu, H.Y., Li, X.P., Ruiz, R., Zhu, H.H.: Group scheduling with nonperiodical maintenance and deteriorating effects. IEEE Trans. Syst. Man Cybernet. Syst. **51**, 2860–2872 (2021)
27. Chen, Y., Xu, Y., Zhang, G., Cheng, Y.: A single machine group scheduling problem with due date assignment and position-dependent costs. Asia-Pacific J. Oper. Res. **40**, 2340007 (2023)
28. He, X., Pan, Q.-K., Gao, L., Neufeld, J.S.: An asymmetric traveling salesman problem based matheuristic algorithm for flowshop group scheduling problem. Eur. J. Oper. Res. **310**, 597–610 (2023)
29. Wang, D.Y., Ye, C.M.: Single machine and group scheduling with random learning rates. AIMS Math. **8**, 19427–19441 (2023)
30. Shabtay, D., Itskovich, Y., Yedidsion, L., Oron, O.: Optimal due date assignment and resource allocation in a group technology scheduling environment. Comput. Oper. Res. **37**, 2218–2228 (2010)
31. Zhu, Z., Sun, L., Chu, F., Liu, M.: Single-machine group scheduling with resource allocation and learning effect. Comput. Ind. Eng. **60**, 148–157 (2011)
32. Wang, D., Huo, Y., Ji, P.: Single-machine group scheduling with deteriorating jobs and allotted resource. Optim. Lett. **8**, 591–605 (2014)
33. Lv, D.-Y., Luo, S.-W., Xue, J., Xu, J.-X., Wang, J.-B.: A note on single machine common flow allowance group scheduling with learning effect and resource allocation. Comput. Ind. Eng. **151**, 106941 (2021)
34. Yan, J.-X., Ren, N., Bei, H.-B., Bao, H., Wang, J.-B.: Study on resource allocation scheduling problem with learning factors and group technology. J. Ind. Manag. Optim. **19**, 3419–3435 (2023)
35. Liu, W.G., Wang, X.Y.: Group technology scheduling with due-date assignment and controllable processing times. Processes **11**, 1271 (2023)
36. He, H., Zhao, Y., Ma, X.J., Lv, Z.-G., Wang, J.-B.: Branch-and-bound and heuristic algorithms for group scheduling with due-date assignment and resource allocation. Mathematics **11**, 4745 (2023)
37. Li, M.-H., Lv, D.-Y., Lu, Y.-Y., Wang, J.-B.: Scheduling with group technology, resource allocation, and learning effect simultaneously. Mathematics **12**, 1029 (2024)
38. Chen, Y., Ma, X., Zhang, F., Cheng, Y.: On optimal due date assignment without restriction and resource allocation in group technology scheduling. J. Comb. Optim. **45**, 64 (2023)
39. Yang, Y., Yin, G., Wang, C., Yin, Y.: Due date assignment and two-agent scheduling under multitasking environment. J. Comb. Optim. **44**, 2207–2223 (2022)
40. Geng, X.-N., Sun, X.Y., Wang, J.Y., Pan, L.: Scheduling on proportionate flow shop with job rejection and common due date assignment. Comput. Ind. Eng. **181**, 109317 (2023)
41. Lv, D.-Y., Wang, J.-B.: No-idle flow shop scheduling with deteriorating jobs and common due date under dominating machines. Asia-Pacific J. Oper. Res. (2024). https://doi.org/10.1142/S0217595924500039
42. Wang, J.-B., Bao, H., Wang, C.: Research on multiple slack due-date assignments scheduling with position-dependent weights. Asia-Pacific J. Oper. Res. (2024). https://doi.org/10.1142/S0217595923500392

43. Framinan, J.M., Leisten, R.: An efficient constructive heuristic for flowtime minimisation in permutation flow shops. Omega **31**, 311–317 (2003)
44. Li, M.-H., Lv, D.-Y., Zhang, L.-H., Wang, J.-B.: Permutation flow shop scheduling with makespan objective and truncated learning effects. J. Appl. Math. Comput. **70**, 2907–2939 (2024)
45. Wang, J.-B., Wang, M.-Z.: Worst-case analysis for flow shop scheduling problems with an exponential learning effect. J. Oper. Res. Soc. **63**, 130–137 (2012)
46. Liu, X., Li, K., Ren, H.: A hybrid algorithm for the permutation flowshop scheduling problem without intermediate buffers. Discret. Dyn. Nat. Soc. **2015**, 548363 (2015)
47. Sun, X.Y., Geng, X.-N., Liu, F.: Flow shop scheduling with general position weighted learning effects to minimise total weighted completion time. J. Oper. Res. Soc. **72**, 2674–2689 (2021)
48. Lv, D.-Y., Wang, J.-B.: Research on two-machine flow shop scheduling problem with release dates and truncated learning effects. Eng. Optim. (2024). https://doi.org/10.1080/0305215X.2024.2372633
49. Lin, S.-S.: Due-window assignment and resource allocation scheduling with truncated learning effect and position-dependent weights. Discret. Dyn. Nat. Soc. **2020**, 9260479 (2020)
50. Mao, R.-R., Wang, Y.-C., Lv, D.-Y., Wang, J.-B., Lu, Y.-Y.: Delivery times scheduling with deterioration effects in due window assignment environments. Mathematics **11**, 3983 (2023)
51. Lv, D.-Y., Xue, J., Wang, J.-B.: Minmax common due-window assignment scheduling with deteriorating jobs. J. Oper. Res. Soc. China **12**, 681–693 (2024)
52. Zhang, L.-H., Geng, X.-N., Xue, J., Wang, J.-B.: Single machine slack due window assignment and deteriorating jobs. J. Ind. Manag. Optim. **20**, 1593–1614 (2024)
53. Gawiejnowicz, S.: Models and algorithms of time-dependent scheduling. Springer, Berlin (2020)
54. Lv, Z.-G., Zhang, L.-H., Wang, X.-Y., Wang, J.-B.: Single machine scheduling proportionally deteriorating jobs with ready times subject to the total weighted completion time minimization. Mathematics **12**, 610 (2024)
55. Mao, R.-R., Lv, D.-Y., Ren, N., Wang, J.-B.: Supply chain scheduling with deteriorating jobs and delivery times. J. Appl. Math. Comput. **70**, 2285–2312 (2024)
56. Ma, R., Meng, L.M., Zhang, Y.Z.: Online scheduling problem of incompatible batch processing with deterioration effect and delivery time in the steel rolling process. J. Oper. Res. Soc. China (2024). https://doi.org/10.1080/01605682.2024.2366537