Journal of Industrial and Management Optimization

JIMO, 22(9): 4151–4175.
DOI: 10.3934/jimo.2026146
Received: 05 May 2026
Revised: 15 July 2026
Accepted: 28 July 2026
Published: 06 August 2026

https://www.aimspress.com/journal/jimo

Research article

Research on the single-machine group scheduling problem with general deteriorating setup times and learning effects

Heng Yin, Jia-Ming Gu and Ji-Bo Wang*

School of Economics and Management, Shenyang Aerospace University, Shenyang 110136, P. R. China

* **Correspondence:** Email: wangjibo75@163.com.

Abstract: This paper considers a single-machine group scheduling problem with general deteriorating setup times and learning effects characterized by truncated exponential functions. The goal is makespan minimization. We establish that the optimal ordering of jobs inside any group adheres to the shortest processing time (SPT) rule, and further identify a sufficient condition that determines the optimal sequence of groups. We solve the problem using four approaches: a branch-and-bound algorithm, a heuristic, a mixed-integer linear programming (MILP) model, and a mixed-integer nonlinear programming (MINLP) model. The effectiveness and efficiency of the proposed methods are examined via numerical experimentation.

Keywords: scheduling; learning effect; single-machine; group technology; deterioration effect
Mathematics Subject Classification: 90B35

1. Introduction

In actual production environments, job processing times are rarely static; they often exhibit dynamic changes. Two common patterns are learning and deterioration. However, the traditional scheduling theory typically assumes constant processing times. The learning effect describes a gradual reduction in processing time as workers gain experience from repeated operations. Conversely, the deterioration effect captures an increase in processing time when a job's start time is delayed. For instance, in plastic molding, mold temperature drops over time. This requires additional heating for subsequent jobs, thus extending their processing times (see Luo [1]).

Azzouz et al. [2] provided a systematic review of scheduling problems with learning effects. The first study on scheduling with position-dependent processing times can be traced back to Gawiejnowicz [3], while Mosheiov [4] was the first to directly analyze parallel machine scheduling with position-dependent deterioration (then referred to as the "aging effect"). Subsequently, Gawiejnowicz [5] provided a comprehensive review of scheduling problems with deterioration effects. In recent years, research on scheduling with learning or deterioration effects has increasingly expanded to areas such as multi-machine environments, stochastic factors, and multi-objective optimization (see Bai et al. [6], Zhang [7], Muştu and Eren [8], Jemmali and Hidri [9], Liu and Wang [10], Sun et al. [11], Qiu and Wang [12], Wang et al. [13], Meng et al. [14], Sun and Geng [15], Lv and Wang [16], Wang [17], Wu et al. [18], Kong et al. [19], Sun et al. [20], and Liu et al. [21]). In addition, Rudek [22] established a generic optimization framework to schedule problems that involve both deterioration effects and maintenance activities.

Meanwhile, some scholars have attempted to incorporate both learning effects and deterioration effects into a single model. For instance, Gao et al. [23] studied a single-machine scheduling problem that considers both DeJong's learning effect and deteriorating maintenance activities. Li et al. [24] considered single-machine scheduling with learning effects and deteriorating delivery times. Luo [1] further introduced both types of effects into a single-machine problem in a stochastic environment. Parichehreh et al. [25] addressed parallel machine scheduling considering both job deterioration and operator learning and designed a heuristic algorithm. Song et al. [26] investigated scheduling with multiple due-windows assignment incorporating learning and deteriorating effects. Lu and Li [27] studied resource allocation scheduling with learning and deteriorating effects. While these studies cover various forms of learning and deterioration effects, their models are primarily designed for general single-machine or flow-shop settings. However, they do not address a more complex scenario, namely the coexistence of both effects within group technology (GT). In this context, intra-group processing times are subject to learning, while inter-group setup times are affected by deterioration.

The advancement of flexible manufacturing systems has greatly promoted GT. GT classifies jobs with similar designs or processes into groups, where jobs within a group are processed consecutively, while switching between different groups incurs additional setup time. This approach improves the manufacturing efficiency and perfectly aligns with flexible manufacturing systems, which must adapt to small-batch, high-mix production requirements (see Wang et al. [28] and Qian and Zhan [29]). To date, scholars have conducted extensive research on group scheduling problems (see Liu et al. [30], Chen and Cheng [31], Qian [32], Keshavarz et al. [33], Zhang et al. [34], and Chen et al. [35]). Recently, Wang and Liu [36] and Zhang and Yin [37] considered a single-machine scheduling problem that integrates GT, resource allocation, and different due-date assignments. Subsequently, Wang and Liu [38] further considered the general case with unrestricted due-date assignments and proposed a branch-and-bound (B&B) algorithm and a heuristic algorithm. Li et al. [39] investigated the single-machine scheduling problem that simultaneously considers GT, convex resource allocation, and learning effects.

In a broader research context, Lv and Wang [40] addressed the single-machine group scheduling problem with resource allocation (via linear and convex functions) and slack due-date window assignments, for which they adopted a minimax criterion and developed the corresponding solution algorithms. Zhang and Wang [41] and Zhang et al. [42] studied single-machine group scheduling problems with resource allocation and due-date window assignments. Huang et al. [43] studied the single-machine group scheduling problem with convex resource allocation and multiple position-dependent learning effects. Yin et al. [44] studied single-machine green scheduling by considering the

deteriorating effect of setup times between groups and the learning effect of job processing times, and proved the problem to be NP-hard and provided solution algorithms.

In more complex production environments, Li et al. [45] proposed a Q-learning-assisted cooperative multi-objective evolutionary algorithm (QACMOEA) to efficiently solve the distributed flowshop group scheduling problem with blocking constraints and sequence-dependent setup times, which simultaneously enhances the solution set quality and achieves a Pareto-optimal trade-off between makespan and total energy consumption. Hu et al. [46] proposed the contrastive shaking-based variable neighborhood search integrated with an iterated greedy (CSVNS-IG) strategy to solve parallel machine group scheduling with time-dependent setup and combined deterioration-learning effects, thereby providing a polynomial optimal solution for single-machine cases and significantly improving the performance for parallel machines.

Although these studies are comprehensive and consider many practical factors, they mostly model learning and deterioration using linear or other specific functions, a limitation that motivates an extension toward a more general framework based on truncated exponential functions. Recently, Yin and Gao [47] considered single-machine group scheduling in which the group setup times have deteriorating effects and the jobs in each group have learning effects. Inspired by their research, this paper extends the learning effect model to a broader class to enhance its applicability and practical relevance. Considering that, in practice, both the increment in processing time due to the deterioration effect and the reduction due to the learning effect exhibit saturation phenomena, we incorporate the truncated exponential function into the learning effect and simultaneously modify the deterioration model to a truncated exponential form.

To address this problem, we develop a diversified solution framework with the objective of minimizing the makespan (i.e., the maximum completion time). We conjecture that this problem is NP-hard in the general case, but also analyze special cases where the optimal sequence can be obtained. For the general case, a B&B algorithm and mathematical models are proposed to solve the problem. The results related to GT scheduling problems are presented in Table 1. The remainder of this paper is organized as follows: Section 2 describes the problem; Section 3 presents basic results; Section 4 proposes the B&B algorithm; Section 5 presents two mathematical models; Section 6 provides numerical experimental results; and Section 7 concludes the paper.

Table 1. Results of GT.

Reference	Objective Function	Setup Time	Job-Processing Time	Solution Method
Kuo and Yang [48]	Makespan	Constant number	$p_{okr}^{A}=p_{ok}\left(1+\sum_{l=1}^{r-1}p_{o[l]}\right)^{\alpha_o}$	Polynomial time algorithm
	Total completion time	Constant number		Heuristic algorithm
Wu and Lee [49]	Makespan	$S_o^A=\vartheta_o+\delta t_o$	$p_{okr}^{A}=p_{ok}+bt$	Polynomial time algorithm
	Total completion time			Heuristic algorithm
Yang and Yang [50]	Makespan	$S_o^A=\delta_o t_o$	$p_{okr}^{A}=p_{ok}r^{\alpha_o}$	Polynomial time algorithm
	Total completion time			Polynomial time algorithm Under a special condition
	Makespan	$S_o^A=\delta_o t_o$	$p_{okr}^{A}=p_{ok}\left(1+\sum_{l=1}^{r-1}p_{o[l]}\right)^{\alpha_o}$	Polynomial time algorithm
	Total completion time			Polynomial time algorithm Under a special condition
Kuo [51]	Makespan	$S_{or}^A=\vartheta_o r^{\alpha}$	$p_{okr}^{A}=p_{ok}\left(1+\sum_{l=1}^{r-1}p_{o[l]}\right)^{\alpha_o}$	Polynomial time algorithm
	Total completion time			Polynomial time algorithm Under a special condition
He and Sun [52]	Makespan	$S_o^A=\vartheta_o+\delta_o t_o$	$p_{okr}^{A}=\left(p_{ok}+b_o t\right)r^{\alpha}$	Polynomial time algorithm
	Total completion time			Polynomial time algorithm Under a special condition
Wang et.al. [53]	Makespan	Constant number	$p_{ok}^{A}=p_{ok}\left(1-bt\right)$	Heuristic algorithm branch-and-bound algorithm
Ning and Sun [54]	Makespan	$S_o^A=\vartheta_o+\delta_o t_o$	$p_{okr}^{A}=$ $p_{ok}\left[Af_o\left(\sum_{l=1}^{r-1}p_{o[l]}\right)+Bh_o(r)\right]$	Polynomial time algorithm
	Total (weighted) completion time			Polynomial time algorithm Under a special condition
Yin et.al. [44]	Total weighted completion time	$S_o^A=\vartheta_o+\delta_o t_o$	$p_{okr}^{A}=$ $p_{ok}\left[Af_o\left(\sum_{l=1}^{r-1}p_{o[l]}\right)+Bh_o(r)\right]$	Heuristic algorithm, simulated annealing, branch-and-bound algorithm
Yin and Gao [47]	Makespan	$S_o^A=\vartheta_o+\delta_o t_o$	$p_{okr}^{A}=$ $p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]})\right)h_o(r),\varphi_o\right\}$	Polynomial time algorithm Heuristic algorithm branch-and-bound algorithm
This paper	Makespan	$S_o^A=$ $\vartheta_o+\delta_o\min\{t_o,\varpi\}$	$p_{okr}^{A}=$ $p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\}$	Heuristic algorithm branch-and-bound algorithm MILP and MINLP

2. Problem description

This section begins by defining the symbols used throughout the paper (see Table 2), and then presents a general description of the problem.

Table 2. List of symbols.

Symbol	Meaning
n	Total number of jobs
e	Number of groups
G_o	Group o, $o=1,2,3,\cdots,e$
n_o	Number of jobs that belong to G_o, $o=1,2,3,\cdots,e$
J_{ok}	Job k in G_o, $o=1,2,3,\cdots,e$, $k=1,2,3,\cdots,n_o$
p_{ok}	Normal processing time of J_{ok}, $o=1,2,3,\cdots,e$, $k=1,2,3,\cdots,n_o$
$p_{o[k]}$	Normal processing time of J_{ok} when placed in the k-th position of G_o, $o=1,2,3,\cdots,e$, $k=1,2,3,\cdots,n_o$
S_o^A	Actual setup time of G_o, $o=1,2,3,\cdots,e$
p_{okr}^A	Actual processing time of job J_{ok} when placed in the r-th position of G_o, $o=1,2,3,\cdots,e$, $k=1,2,3,\cdots,n_o$
ϑ_o	Normal setup time of G_o, $o=1,2,3,\cdots,e$
t_o	Start setup time of G_o, $o=1,2,3,\cdots,e$
δ_o	Setup deterioration rate of G_o, $o=1,2,3,\cdots,e$
ϖ	Truncation parameter of S_o, $o=1,2,3,\cdots,e$
φ_o	Truncation parameter of G_o, $o=1,2,3,\cdots,e$
α_o	Learning index of G_o, $o=1,2,3,\cdots,e$
β_o	Learning index of G_o, $o=1,2,3,\cdots,e$
ζ	A job schedule of n jobs
$C_{ok}=C_{ok}(\pi)$	Completion time for job J_{ok} in π
$C_{\max}$	Makespan, $C_{\max}=\max\{C_{ok}\mid o=1,2,3,\cdots,e;k=1,2,3,\cdots,n_o\}$

Suppose that n jobs are partitioned into e groups and processed on a single-machine. The machine can process at most one job at a time. All jobs are available at time zero, and jobs within the same group must be processed consecutively. Group G_o contains n_o jobs, $o=1,2,3,\cdots,e$, and $\sum_{o=1}^{e} n_o = n$. Due to the learning effect, when job J_{ok} is scheduled in the r-th position within its group, its actual processing time is as follows:

$$p_{okr}^A = p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_o\right\}, o=1,2,3,\ldots,e; r,k=1,2,3,\ldots,n_o, \tag{1}$$

where p_{ok} denotes the normal processing time of job J_{ok}. The function $f_o(x,y)$ satisfies that $\frac{\partial f_o(x,y)}{\partial x}\le 0$, $\frac{\partial^2 f_o(x,y)}{\partial x^2}\ge 0$, and $f_o(0,0)=1$. The function $g_o(x)$ satisfies that $\frac{\partial^2 g_o(x)}{\partial x^2}\le 0$, $g_o(0)=0$, and $\sum_{l=1}^{0} g_o(p_{o[l]})=0$. The function $h_o(r)$ is a non-increasing function with $h_o(1)=1$, where r denotes the position within the group. The truncated parameter of group G_o is φ_o with $0<\varphi_o\le 1$.

Within a group, the learning effect is captured by the actual processing time p_{okr}^{A}. This value decreases as the cumulative normal processing time of earlier jobs (expressed as $\sum_{l=1}^{r-1} g_o(p_{o[l]})$) increases and as the job's position r advances (via $h_o(r)$). The parameter φ_o prevents the actual processing time from becoming excessively small.

The actual setup time of group G_o is as follows:

$$S_o^A = \vartheta_o + \delta_o \min\{t_o, \varpi\}, o = 1,2,3,\ldots,e, \tag{2}$$

where t_o is the time when the setup of group G_o begins, $\delta_o > 0$ is the deterioration rate, ϖ is the truncation parameter for the setup time between groups, and ϑ_o is the normal setup time of group G_o. For groups, the deterioration effect is reflected in the group setup time S_o^A, which increases with the group's start time t_o, but is constrained by a truncation parameter ϖ (via the term $\min\{t_o, \varpi\}$) to prevent excessive growth. These two effects jointly determine the makespan.

For a given sequence ζ, let $C_{ok}(\zeta)$ denote the completion time of job J_{ok} in group G_o. The goal is to determine an optimal sequence to minimize the makespan, defined as $C_{\max} = \max\{C_{ok} \mid o = 1,\cdots,e; k = 1,\cdots,n_o\}$. For a sequence $\zeta = \{C_{11},\cdots,C_{1n_1},\cdots,C_{e1},\cdots,C_{en_e}\}$, let $\tilde{p}_o = \sum_{r=1}^{n_o} p_{o[r]} \max\left\{f_o\left(\sum_{l=1}^{r-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_o\right\}$; then, we have the following:

$$\begin{aligned}
&C_{1[1]} = \vartheta_1 + p_{1[1]}\\
&C_{1[2]} = \vartheta_1 + p_{1[1]} + p_{1[2]} \max\left\{f_o\left(\sum_{l=1}^{2-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_1\right\}\\
&\vdots\\
&C_{1[n_1]} = \vartheta_1 + \tilde{p}_1\\
&C_{2[1]} = C_{1[n_1]} + \vartheta_2 + \delta_2 \min\left\{C_{1[n_1]}, \varpi\right\} + p_{2[1]}\\
&C_{2[2]} = C_{1[n_1]} + \vartheta_2 + \delta_2 \min\left\{C_{1[n_1]}, \varpi\right\} + p_{2[1]} + p_{2[2]} \max\left\{f_o\left(\sum_{l=1}^{2-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_2\right\}\\
&\vdots\\
&C_{2[n_2]} = C_{1[n_1]} + \vartheta_2 + \delta_2 \min\left\{C_{1[n_1]}, \varpi\right\} + \tilde{p}_2\\
&\vdots\\
&C_{e[n_e]} = C_{e-1[n_{e-1}]} + \vartheta_e + \delta_e \min\left\{C_{e-1[n_{e-1}]}, \varpi\right\} + \tilde{p}_e
\end{aligned}$$

$$C_{\max} = \sum_{o=1}^{e} \tilde{p}_o + \sum_{o=1}^{e} \vartheta_o + \sum_{o=2}^{e} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\}. \tag{3}$$

This problem is expressed using the three-field notation (Graham et al. [55], Wang et al. [56]) as follows:

$$1\left|S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\},p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\},GT\right|C_{\max}, \tag{4}$$

where 1 denotes a single-machine environment, $S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\}$ indicates that group setup times deteriorate with the start time and are truncated by parameter ϖ, $p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\}$ indicates that job processing times within a group are subject to learning effects and truncated by parameter φ_o, GT denotes the group technology, and $C_{\max}$ denotes the objective of minimizing the makespan.

3. Basic results

Lemma 1 $R(x)=\dfrac{f_o(U+g_o(x),h_o(r+1))-f_o(U,h_o(r))}{x}$ is non-decreasing with x, and $U>0$, $\dfrac{\partial f_o(x,y)}{\partial x}\le 0$, $\dfrac{\partial^2 f_o(x,y)}{\partial x^2}\ge 0$, $\dfrac{\partial^2 g_o(x)}{\partial x^2}\le 0$, $h_o(r)$ is a non-increasing function, $g_o(0)=0$, $h_o(1)=1$, and $x\ge 0$.

Proof. To analyze the monotonicity of $R(x)$, we have the following:

$$R'(x)=\frac{\frac{\partial f_o}{\partial x}(U+g_o(x),h_o(r+1))\cdot g_o'(x)\cdot x-f_o(U+g_o(x),h_o(r+1))+f_o(U,h_o(r))}{x^2},$$

set $H(x)=\dfrac{\partial f_o}{\partial x}(U+g_o(x),h_o(r+1))\cdot g_o'(x)\cdot x-f_o(U+g_o(x),h_o(r+1))+f_o(U,h_o(r))$,

$H'(x)=\dfrac{\partial^2 f_o}{\partial x^2}(U+g_o(x),h_o(r+1))\cdot(g_o'(x))^2\cdot x+\dfrac{\partial f_o}{\partial x}(U+g_o(x),h_o(r+1))\cdot g_o''(x)\cdot x$. Obviously, if $\dfrac{\partial f_o(x,y)}{\partial x}\le 0$, $\dfrac{\partial^2 f_o(x,y)}{\partial x^2}\ge 0$, and $g''(x)\le 0$, then $H'(x)\ge 0$; consequently, $H(x)\ge H(0)=0$, and we have $R'(x)\ge 0$. Thus, $R(x)$ is a non-decreasing function in x.

Lemma 2 For the problem $1\left|S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\},p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\},GT\right|C_{\max}$ within a group, the non-decreasing order of p_{ok} gives the optimal job sequence (i.e., $p_{o1}\le p_{o2}\le,\dots,\le p_{on_o}$, $o=1,2,3,\cdots,e$).

Proof. For group G_o, let $\pi_o=[Z_1,J_{ok},J_{oi},Z_2]$, $\pi_o'=[Z_1,J_{oi},J_{ok},Z_2]$, and $U=\sum_{l=1}^{r-1}g_o(p_{o[l]})$, where Z_1 and Z_2 are partially sorted, and $p_{ok}\le p_{oi}$. Suppose there are $r-1$ jobs in Z_1. Then, we have the following:

$$\begin{aligned}C_{oi}(\pi_o)=Q&+p_{ok}\max\left\{f_o\left(U,h_o\left(r\right)\right),\varphi_o\right\}\\&+p_{oi}\max\left\{f_o\left(U+g_o(p_{ok}),h_o\left(r+1\right)\right),\varphi_o\right\},\end{aligned} \tag{5}$$

$$\begin{aligned} C_{ok}(\pi_o') = Q &+ p_{oi} \max\left\{f_o\left(U, h_o(r)\right), \varphi_o\right\} \\ &+ p_{ok} \max\left\{f_o\left(U + g_o(p_{oi}), h_o(r+1)\right), \varphi_o\right\}, \end{aligned} \tag{6}$$

where Q denotes the completion time of the final job in Z_1. Subtracting Equation (5) from Equation (6) yields the following:

$$\begin{aligned} C_{ok}(\pi_o') - C_{oi}(\pi_o) = &\left(p_{oi} - p_{ok}\right) \max\left\{f_o\left(U, h_o(r)\right), \varphi_o\right\} \\ &+ p_{ok} \max\left\{f_o\left(U + g_o(p_{oi}), h_o(r+1)\right), \varphi_o\right\} \\ &- p_{oi} \max\left\{f_o\left(U + g_o(p_{ok}), h_o(r+1)\right), \varphi_o\right\}. \end{aligned} \tag{7}$$

We consider the following cases:

Case 1: If $f_o\left(U, h_o(r)\right) \le \varphi_o$, then from Equation (7), we obtain

$$C_{ok}(\pi_o') - C_{oi}(\pi_o) = \left(p_{oi} - p_{ok}\right)\varphi_o + p_{ok}\varphi_o - p_{oi}\varphi_o = 0. \tag{8}$$

Case 2: If $f_o\left(U, h_o(r)\right) > \varphi_o \ge f_o\left(U + g_o(p_{ok}), h_o(r+1)\right)$, then from Equation (7), we obtain

$$\begin{aligned} C_{ok}(\pi_o') - C_{oi}(\pi_o) &= \left(p_{oi} - p_{ok}\right) f_o\left(U, h_o(r)\right) + p_{ok}\varphi_o - p_{oi}\varphi_o \\ &= \left(p_{oi} - p_{ok}\right)\left[f_o\left(U, h_o(r)\right) - \varphi_o\right] \\ &\ge 0. \end{aligned} \tag{9}$$

Case 3: If $f_o\left(U + g_o(p_{ok}), h_o(r+1)\right) > \varphi_o \ge f_o\left(U + g_o(p_{oi}), h_o(r+1)\right)$, then from Equation (7), we obtain

$$\begin{aligned} C_{ok}(\pi_o') - C_{oi}(\pi_o) &= \left(p_{oi} - p_{ok}\right) f_o\left(U, h_o(r)\right) \\ &\quad + p_{ok}\varphi_o - p_{oi} f_o\left(U + g_o(p_{ok}), h_o(r+1)\right) \\ &\ge \left(p_{oi} - p_{ok}\right) f_o\left(U, h_o(r)\right) \\ &\quad + p_{ok} f_o\left(U + g_o(p_{oi}), h_o(r+1)\right) \\ &\quad - p_{oi} f_o\left(U + g_o(p_{ok}), h_o(r+1)\right) \\ &= p_{ok} p_{oi}\left[\frac{f_o\left(U + g_o(p_{oi}), h_o(r+1)\right) - f_o\left(U, h_o(r)\right)}{p_{oi}}\right. \\ &\quad \left. - \frac{f_o\left(U + g_o(p_{ok}), h_o(r+1)\right) - f_o\left(U, h_o(r)\right)}{p_{ok}}\right]. \end{aligned} \tag{10}$$

Let p_{ok} be x_1, and p_{oi} be x_2, $x_1 \le x_2$. Then, by Lemma 1, we obtain Equation (10) ≥ 0.

Case 4: If $f_o\left(U+g_o(p_{oi}),h_o\left(r+1\right)\right)>\varphi_o$, then from Equation (7), we obtain

$$\begin{aligned} C_{ok}(\pi_o')-C_{oi}(\pi_o)&=\left(p_{oi}-p_{ok}\right)f_o\left(U,h_o\left(r\right)\right)\\ &\quad+p_{ok}f_o\left(U+g_o(p_{oi}),h_o\left(r+1\right)\right)\\ &\quad-p_{oi}f_o\left(U+g_o(p_{ok}),h_o\left(r+1\right)\right). \end{aligned} \tag{11}$$

Similar to Case 3, we obtain Equation (11) ≥ 0.

Lemma 3 For the problem $1\left|S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\},p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\},GT\right|C_{\max}$, the SPT order determines the optimal sequence for jobs belonging to a group (i.e., non-decreasing order of p_{ok}). For the ordering among groups, let $\tilde{p}_o=\sum_{r=1}^{n_o}p_{o[r]}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\}$. Then, if $\delta_i\leq\delta_k$ and only if $\frac{\delta_i}{\vartheta_i+\tilde{p}_i}\leq\frac{\delta_k}{\vartheta_k+\tilde{p}_k}$, then, the optimal ordering among groups is given by a non-increasing sequence of δ_o.

Proof. Let $\pi=[Z_1,G_k,G_i,Z_2]$ and $\pi'=[Z_1,G_i,G_k,Z_2]$ denote a complete sequence, $\tilde{p}_o=\sum_{r=1}^{n_o}p_{o[r]}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\}$. Let T represent the finishing time of the final job in Z_1; then, we have the following:

$$\begin{aligned} C_{in_i}(\pi)&=T+\vartheta_k+\delta_k\min\{T,\varpi\}+\tilde{p}_k\\ &\quad+\vartheta_i+\delta_i\min\left\{T+\vartheta_k+\delta_k\min\{T,\varpi\}+\tilde{p}_k,\varpi\right\}+\tilde{p}_i, \end{aligned} \tag{12}$$

$$\begin{aligned} C_{kn_k}(\pi')&=T+\vartheta_i+\delta_i\min\{T,\varpi\}+\tilde{p}_i\\ &\quad+\vartheta_k+\delta_k\min\left\{T+\vartheta_i+\delta_i\min\{T,\varpi\}+\tilde{p}_i,\varpi\right\}+\tilde{p}_k, \end{aligned} \tag{13}$$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi)&=\delta_i\min\{T,\varpi\}-\delta_k\min\{T,\varpi\}\\ &\quad+\delta_k\min\left\{C_{in_i}(\pi'),\varpi\right\}-\delta_i\min\left\{C_{kn_k}(\pi),\varpi\right\}, \end{aligned} \tag{14}$$

where $C_{kn_k}(\pi)=T+\vartheta_k+\delta_k\min\{T,\varpi\}+\tilde{p}_k$, $C_{in_i}(\pi')=T+\vartheta_i+\delta_i\min\{T,\varpi\}+\tilde{p}_i$. Clearly, both $C_{kn_k}(\pi)$ and $C_{in_i}(\pi')$ are greater than *T*. Now, we consider the following cases:

Case 1: $\varpi\leq T$

$$C_{kn_k}(\pi')-C_{in_i}(\pi)=\delta_i\varpi-\delta_k\varpi+\delta_k\varpi-\delta_i\varpi=0. \tag{15}$$

Case 2: $T<\varpi$, $\varpi<C_{in_i}(\pi')$, $\varpi<C_{kn_k}(\pi)$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k\varpi-\delta_i\varpi \\ &= (\delta_i-\delta_k)(T-\varpi). \end{aligned} \tag{16}$$

Case 3: $T<\varpi$, $\varpi<C_{in_i}(\pi')$, $C_{kn_k}(\pi)\le\varpi$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k\varpi-\delta_i\left(C_{kn_k}(\pi)\right) \\ &> \delta_i T-\delta_k T+\delta_k C_{kn_k}(\pi)-\delta_i C_{kn_k}(\pi) \\ &= (\delta_i-\delta_k)\left(T-C_{kn_k}(\pi)\right). \end{aligned} \tag{17}$$

Case 4: $T<\varpi$, $C_{in_i}(\pi')\le\varpi$, $\varpi<C_{kn_k}(\pi)$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k C_{in_i}(\pi')-\delta_i\varpi \\ &> \delta_i T-\delta_k T+\delta_k C_{in_i}(\pi')-\delta_i C_{in_i}(\pi') \\ &= (\delta_i-\delta_k)\left(T-C_{in_i}(\pi')\right). \end{aligned} \tag{18}$$

Case 5: $T<\varpi$, $C_{in_i}(\pi')<\varpi$, $C_{kn_k}(\pi)<\varpi$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k\left(T+\vartheta_i+\delta_i T+\tilde{p}_i\right) \\ &\quad -\delta_i\left(T+\vartheta_k+\delta_k T+\tilde{p}_k\right) \\ &= \delta_k\left(\vartheta_i+\tilde{p}_i\right)-\delta_i\left(\vartheta_k+\tilde{p}_k\right). \end{aligned} \tag{19}$$

Case 6: $T<\varpi$, $C_{in_i}(\pi')=C_{kn_k}(\pi)<\varpi$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k C_{in_i}(\pi')-\delta_i C_{kn_k}(\pi) \\ &= (\delta_i-\delta_k)\left(T-C_{in_i}(\pi')\right). \end{aligned} \tag{20}$$

Case 7: $T<\varpi$, $C_{in_i}(\pi')=C_{kn_k}(\pi)>\varpi$

$$\begin{aligned} C_{kn_k}(\pi')-C_{in_i}(\pi) &= \delta_i T-\delta_k T+\delta_k\varpi-\delta_i\varpi \\ &= (\delta_i-\delta_k)(T-\varpi). \end{aligned} \tag{21}$$

In summary, when $\delta_i\le\delta_k$ and $\dfrac{\delta_i}{\vartheta_i+\tilde{p}_i}\le\dfrac{\delta_k}{\vartheta_k+\tilde{p}_k}$ hold in the above cases, we have $C_{kn_k}(\pi')-C_{in_i}(\pi)\ge 0$.

Lemma 4 **(**Hardy et al. [57] and Huang et al. [43]) For $\sum_{o=1}^{e} x_o y_o$, if sequence x_o is arranged in non-decreasing order and sequence y_o in non-increasing order, or vice versa, then $\sum_{o=1}^{e} x_o y_o$ is minimized; if both sequences are sorted in the same order, then $\sum_{o=1}^{e} x_o y_o$ is maximized.

4. Basic results

4.1. Lower bound

Potentially NP-hard is the problem considered in this paper. To obtain an exact solution, we establish upper and lower bounds, which lay the groundwork for a B&B algorithm. During computation, the algorithm employs a breadth-first search strategy, where the bounds are calculated to enable pruning and improve the search efficiency.

Let $\pi=\{\pi^{\tau},\pi^{\upsilon}\}$ be a scheduling sequence, where π^{τ} is the sequence of the first d determined groups, and π^{υ} is the sequence of the remaining $e-d$ undetermined scheduling sequences. We have the following:

$$\begin{aligned} F &= \sum_{o=1}^{e} \tilde{p}_o + \sum_{o=1}^{e} \vartheta_o + \sum_{o=2}^{e} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\} \\ &= \sum_{o=1}^{d} \tilde{p}_o + \sum_{o=1}^{d} \vartheta_o + \sum_{o=2}^{d} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\} + \\ &\quad \sum_{o=d+1}^{e} \tilde{p}_o + \sum_{o=d+1}^{e} \vartheta_o + \sum_{o=d+1}^{e} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\}, \end{aligned} \tag{22}$$

where $\sum_{o=d+1}^{e} \tilde{p}_o$ and $\sum_{o=d+1}^{e} \vartheta_o$ are constants, and $\sum_{o=d+1}^{e} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\}$ can be minimized by Lemmas 3 and 4. Let $\delta_{\langle o\rangle}$ denote $\delta_{d+1} \geq \delta_{d+2} \geq \cdots \geq \delta_e$, and $\hat{C}_{o-1,n_{o-1}}$ denote the unsorted groups sorted in non-increasing order of $\frac{\delta_o}{\vartheta_o + \tilde{p}_o}$, with $\delta_{\langle o\rangle}$ and $\hat{C}_{o-1,n_{o-1}}$ not necessarily referring to the same group. We give the lower bound as follows:

$$\begin{aligned} LB &= \sum_{o=1}^{d} \tilde{p}_o + \sum_{o=1}^{d} \vartheta_o + \sum_{o=2}^{d} \delta_o \min\{C_{o-1[n_{o-1}]}, \varpi\} \\ &\quad + \sum_{o=d+1}^{e} \tilde{p}_o + \sum_{o=d+1}^{e} \vartheta_o + \sum_{o=d+1}^{e} \delta_{\langle o\rangle} \min\{\hat{C}_{o-1,n_{o-1}}, \varpi\}. \end{aligned} \tag{23}$$

4.2. Upper bound

From the above analysis, we develop the following heuristic algorithm (Algorithm 1) to provide an upper bound for the branch-and-bound algorithm.

Algorithm 1 A constructive heuristic for

$1\left|S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\},p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\},GT\right|C_{\max}$.

Input: Set of groups $\pi=\{G_1,G_2,\cdots,G_e\}$,with parameters $n_o,p_{ok},\vartheta_o,\varphi_o,\varpi,\delta_o$ for each group, and functions f_o,g_o,h_o.

Output: An optimal group sequence π^*, and its makespan $C_{\max}$.

Step 1: for each group G_o, order the n_o jobs within the group in non-decreasing order of p_{ok} (SPT rule), and calculate the value of $\tilde{p}_o=\sum_{r=1}^{n_o}p_{o[r]}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\}$.

Step 2: generate six candidate group sequences according to the following rules (ties broken arbitrarily):

(a) descending $\delta_o,o=1,2,3,\cdots,e$;

(b) descending $\vartheta_o,o=1,2,3,\cdots,e$;

(c) ascending $n_o,o=1,2,3,\cdots,e$;

(d) ascending $\vartheta_o/n_o,o=1,2,3,\cdots,e$;

(e) descending $\vartheta_o/n_o,o=1,2,3,\cdots,e$;

(f) descending $\delta_o/\vartheta_o+\tilde{p}_o,o=1,2,3,\cdots,e$.

Step 3: for each candidate sequence π generated in Step 2, compute $C_{\max}(\pi)$ using Equation (3). Select the schedule that yields the smallest $C_{\max}$.

4.3. Branch-and-bound algorithm

Using the above lower and upper bounds, we present a B&B algorithm (i.e., Algorithm 2) to solve this problem.

Algorithm 2 A branch-and-bound algorithm for

$1\left|S_o^A=\vartheta_o+\delta_o\min\{t_o,\varpi\},p_{okr}^A=p_{ok}\max\left\{f_o\left(\sum_{l=1}^{r-1}g_o(p_{o[l]}),h_o(r)\right),\varphi_o\right\},GT\right|C_{\max}$.

Input: Set of groups $\pi=\{G_1,G_2,\cdots,G_e\}$,with parameters $n_o,p_{ok},\vartheta_o,\varphi_o,\varpi,\delta_o$ for each group, and functions f_o,g_o,h_o.

Output: An optimal group sequence π^*, and its makespan $C_{\max}$.

Step 1: (Initialization) The initial upper bound is derived from Algorithm 1 and put it into the priority queue as the root node.

Step 2: (Branching and bounding) Select a node from the priority queue and compute its lower bound (see Equation (23)). If the lower bound of this node is greater than or equal to the upper bound, then prune this node and all branches under it. If the lower bound of a complete sequence is less than the upper bound, then retain it as the new upper bound and solution, and go to Step 3).

Step 3: (Termination) Continue Step 2 until the priority queue is empty, then terminate the algorithm.

5. Mathematical model

5.1. MINLP

From the preceding analysis, we derive the optimal job sequence within each group. Then, we formulate a mixed-integer nonlinear programming (MINLP) model to solve $1\left|S_o^A = \vartheta_o + \delta_o \min\{t_o, \varpi\}, p_{okr}^A = p_{ok} \max\left\{f_o\left(\sum_{l=1}^{r-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_o\right\}, GT\right| C_{\max}$.

First, calculate the value of $\tilde{p}_o = \sum_{r=1}^{n_o} p_{o[r]} \max\left\{f_o\left(\sum_{l=1}^{r-1} g_o(p_{o[l]}), h_o(r)\right), \varphi_o\right\}$. Let $C_0 = 0$, and let x_{od} be a binary decision variable such that $x_{od} = 1$ if group o is assigned to position d, and $x_{od} = 0$ otherwise. The MINLP model is as follows:

$$\min F = C_{en_e} \tag{24}$$

s. t.

$$\sum_{o=1}^{e} x_{od} = 1, \quad d = 1, 2, 3, \ldots, e \tag{25}$$

$$\sum_{d=1}^{e} x_{od} = 1, \quad o = 1, 2, 3, \ldots, e \tag{26}$$

$$C_0 + \sum_{o=1}^{e} x_{o1}\left(\vartheta_o + \tilde{p}_o\right) \le C_{1n_1} \tag{27}$$

$$\begin{aligned} &C_{d-1n_{d-1}} + \sum_{o=1}^{e} x_{od}\left(\vartheta_o + \delta_o \min\left\{C_{d-1n_{d-1}}, \varpi\right\} + \tilde{p}_o\right) \le C_{dn_d} \\ &\forall o \in \{1, 2, 3, \cdots, e\}, \forall d \in \{2, 3, \cdots, e\} \end{aligned} \tag{28}$$

$$C_{d-1n_{d-1}} \ge 0 \qquad \forall d \in \{1, 2, 3, \cdots, e\} \tag{29}$$

$$x_{od} \in \{0, 1\} \qquad \forall o, d \in \{1, 2, 3, \cdots, e\} \tag{30}$$

Objective function (24) serves to minimize the makespan. Constraints (25) and (26) enforce a one-to-one correspondence between groups and positions. Constraints (27) and (28) capture the relationship between the completion times of adjacent groups. Constraints (29) and (30) are sign constraints and binary variables.

5.2. MILP

Through a further analysis of the above nonlinear model, the piecewise functions can be linearized. Let $b_d \in \{0,1\}$ be an indicator variable, $w_d = \min\{C_{d-1n_{d-1}}, \varpi\}$, $w_d \geq 0$, $z_{od} \geq 0$, and $z_{od} = x_{od} w_d$ be auxiliary variables, and $M_d = 2\sum_{o=1}^{e}(\vartheta_o + \delta_o \varpi + \tilde{p}_o)$, leading to the following MILP model:

$$\min F = C_{e-1n_{e-1}} + \sum_{o=1}^{e} x_{oe}(\vartheta_o + \tilde{p}_o) + \sum_{o=1}^{e} \delta_o z_{oe} \tag{31}$$

s. t.

$$\sum_{o=1}^{e} x_{od} = 1, \quad d = 1,2,3,\ldots,e \tag{32}$$

$$\sum_{d=1}^{e} x_{od} = 1, \quad o = 1,2,3,\ldots,e \tag{33}$$

$$C_0 + \sum_{o=1}^{e} x_{o1}(\vartheta_o + \tilde{p}_o) \leq C_{1n_1} \tag{34}$$

$$\begin{aligned} &C_{d-1n_{d-1}} + \sum_{o=1}^{e} x_{od}(\vartheta_o + \tilde{p}_o) + \sum_{o=1}^{e} \delta_o x_{od} w_d \leq C_{dn_d} \\ &\forall o \in \{1,2,3,\cdots,e\}, \forall d \in \{2,3,\cdots,e\} \end{aligned} \tag{35}$$

$$\begin{aligned} w_d &\leq C_{d-1n_{d-1}}, & w_d &\leq \varpi \\ w_d &\geq C_{d-1n_{d-1}} - M_d b_d, & w_d &\geq \varpi - M_d(1-b_d) \end{aligned} \tag{36}$$

$$\begin{aligned} z_{od} - w_d &\leq 0, & z_{od} - M_d x_{od} &\leq 0 \\ z_{od} - w_d + M_d(1-x_{od}) &\geq 0, & z_{od} &\geq 0 \end{aligned} \tag{37}$$

$$C_{d-1n_{d-1}} \geq 0 \qquad \forall d \in \{1,2,3,\cdots,e\} \tag{38}$$

$$b_d \in \{0,1\} \qquad \forall d \in \{1,2,3,\cdots,e\} \tag{39}$$

$$x_{od} \in \{0,1\} \qquad \forall o,d \in \{1,2,3,\cdots,e\} \tag{40}$$

Objective function (31) serves to minimize the makespan. Constraints (32) and (33) enforce a one-to-one correspondence between groups and positions. Constraints (34) and (35) capture the relationship

between the completion times of adjacent groups. Constraints (36) and (37) linearize w_d and z_{od}, respectively. Constraints (38), (39), and (40) are the sign restrictions and the binary variables.

6. Computational evaluation

We implemented and tested the heuristic algorithm and the B&B algorithm in C++ using Visual Studio 2022, the MILP model was tested in IBM ILOG Cplex Optimization Studio 22.1.1, and the MINLP model was tested in Gurobi 13.0. We conducted all tests on a desktop workstation equipped with an AMD R9-7945H CPU (2.50 GHz, 16 cores, 32 threads), 1 TB of physical memory, and a maximum memory capacity of 32 GB. The "CPU time" is defined as the running time of the HA, B&B, MILP, and MINLP, and the time unit is milli-seconds (ms). In MILP, MINLP, and B&B, we set the maximum solving time to 3600 (s).

All test instances in this paper are randomly generated based on uniform distributions. To mitigate random fluctuations, 30 independent replicate instances are created for each parameter combination. The experiment adopts a stratified controlled-variable design. The total number of jobs is fixed at $n=100$, and the number of groups e is set to six gradient values: 9, 10, 11, 12, 13, and 14. The number of jobs n_o in each group is randomly assigned subject to $n_o \geq 1$ and the total mass constraint $\sum_{o=1}^{e} n_o = n$. The normal processing time p_{ok} of each job and the nominal setup time ϑ_o of each group are integer values uniformly distributed over intervals $[10,100]$ and $[5,50]$, respectively. The learning coefficients α_o and β_o follow a continuous uniform distribution on $[-0.5,-0.1]$ complying with the monotonicity constraint $\alpha_o, \beta_o \leq 0$. Four fixed stratified values of the learning truncation parameter $\varphi_o(0.50, 0.55, 0.60, 0.65)$ are selected to analyze the influence of different truncation thresholds. The actual job processing time follows the truncated exponential learning formula proposed in this study:

$$p_{okr}^{A} = p_{ok} \max\left\{\left(1+\sum_{l=1}^{r-1} \ln p_{o[l]}\right)^{\alpha_o} r^{\beta_o}, \varphi_o\right\}. \tag{41}$$

The deterioration rate δ_o for group setup times obeys a continuous uniform distribution over $[0.05,0.2]$ such that $\delta_o \geq 0$. The saturation truncation threshold ϖ for deterioration is not randomly sampled; instead, it is deterministically calculated from the aggregate normal processing time of all jobs via $\varpi = 0.6\sum_{o=1}^{e}\sum_{k=1}^{n_o} p_{ok}$. For each fixed (e,φ_o) pair, all other parameters are randomly regenerated, and the heuristic algorithm, B&B algorithm, MILP, and MINLP models are run separately. We consistently record the average and maximum CPU runtime, the number of nodes explored by the B&B, and the error defined as follows:

$$rer = \frac{C_{\max}(\zeta) - C_{\max}(\zeta^*)}{C_{\max}(\zeta^*)} * 100, \tag{42}$$

where ζ is the sequence obtained from HA, B&B, MILP, and MINLP, and ζ^* is the sequence with the smallest value of $C_{\max}$ among HA, B&B, MILP, and MINLP.

From Table 3, we can see that the CPU runtimes of MINLP, MILP, and B&B significantly increase as the number of groups increases, but the growth rate of the B&B algorithm is slower than that of MINLP and MILP. Further observation reveals that the average runtime of MINLP is smaller than that of MILP, but in some instances, the maximum runtime of MINLP is larger than that of MILP.

For a more intuitive comparison of the solution efficiency of each algorithm under different experimental scales, Figures 1 through 4 present the average CPU times of the B&B algorithm, the MINLP model, and the MILP model, along with the average number of search nodes explored by the B&B algorithm for varying group counts. As shown in Figure 1, as the number of groups e increases from 9 to 14, the average CPU time of the B&B algorithm noticeably increases, yet the growth trend remains relatively gradual, indicating that the algorithm maintains satisfactory efficiency as the problem scale expands. In contrast, Figures 2 and 3 show that the average CPU times of the MINLP and MILP models increase far more dramatically, especially when the group count reaches 12 or more, where the solution time sharply surges. Overall, the B&B algorithm significantly outperforms the two mathematical programming models in terms of the solution efficiency, which corroborates the CPU time results recorded in Table 3.

From Table 4, we can see that the heuristic algorithm has a small error, and at this scale, both mathematical models can find exact solutions within the specified time limit. The ratio of the maximum number of nodes searched to the total number of nodes decreases as the number of groups increases (i.e., $\frac{Num_{SN}}{Num_{TN}}$). For example, the ratio $\frac{Num_{SN}}{Num_{TN}} = \frac{6246}{1.300711934\times10^{9}} \approx 4.802\times10^{-6}$ when $e = 12$, but the ratio $\frac{Num_{SN}}{Num_{TN}} = \frac{18728}{2.36975164805\times10^{11}} \approx 7.903\times10^{-8}$ when $e = 14$. Figure 4 further illustrates the trend in the average number of search nodes explored by the B&B algorithm. Although the node count increases with the number of groups, its growth rate is far slower than that of the entire solution space. This confirms that the lower bound proposed in this paper is sufficiently tight, thereby enabling effective pruning and substantially shrinking the search space, thus ensuring the computational efficiency of the B&B algorithm.

In summary, at this scale, the heuristic algorithm can obtain approximate solutions very quickly with a small margin of error. The B&B algorithm demonstrates satisfactory efficiency. Overall, MINLP has a better average solution efficiency than MILP, although in some instances, the maximum CPU runtime of MILP is smaller than that of MINLP.

Table 3. CPU time (ms) of algorithms.

		HA		B&B		MILP		MINLP	
$n*e$	φ_o	mean	max	mean	max	mean	max	mean	max
100*9	0.50	0	0	0.50	15	1675	2625	591	1421
	0.55	0	0	1.03	16	1559	2312	558	1094
	0.60	0	0	1.06	16	1514	2141	643	1344
	0.65	0	0	2.60	16	1481	1828	579	1328
100*10	0.50	0	0	0.53	16	3170	7156	1820	3937
	0.55	0	0	2.57	16	2475	5141	1592	5875
	0.60	0	0	3.63	31	2308	4235	1423	3109
	0.65	0	0	2.10	16	2090	2985	1106	1797
100*11	0.50	0	0	5.60	16	9042	29907	3877	11984
	0.55	0	0	2.67	16	7182	19078	2970	7109
	0.60	0	0	4.63	31	4620	8453	2331	5407
	0.65	0	0	3.63	31	3991	9078	1985	4735
100*12	0.50	0	0	5.67	16	29440	108235	12019	45954
	0.55	0	0	8.30	31	21588	101703	10493	32359
	0.60	0	0	10.33	78	15461	43656	7576	23938
	0.65	0	0	8.27	47	20875	20875	4565	13734
100*13	0.50	0	0	15.70	63	87523	485125	56207	384406
	0.55	0	0	12.60	63	74106	641282	35982	119312
	0.60	0	0	35.43	219	48502	240875	35276	178437
	0.65	0.53	16	28.07	172	25457	44688	15996	82875
100*14	0.50	0	0	30.23	141	315656	1432125	205410	523688
	0.55	0	0	31.30	219	230171	1244109	155978	1269140
	0.60	0	0	32.87	297	71928	211937	54462	176047
	0.65	0	0	43.27	172	59133	328906	45936	254735

Table 4. Number of nodes searched by B&B and the *rer* of algorithms.

		number of nodes searched by B&B		HA		MILP		MINLP	
$n*e$	φ_o	mean	max	rer_{mean}	$rer_{\max}$	rer_{mean}	$rer_{\max}$	rer_{mean}	$rer_{\max}$
100*9	0.50	87.27	469	1.00024	1.00314	1	1	1	1
	0.55	79.50	220	1.00008	1.00008	1	1	1	1
	0.60	67.93	194	1.00037	1.00548	1	1	1	1
	0.65	71.00	151	1.00027	1.00258	1	1	1	1
100*10	0.50	134.93	500	1.00035	1.00291	1	1	1	1
	0.55	126.40	383	1.00030	1.00264	1	1	1	1
	0.60	234.90	1916	1.00051	1.00646	1	1	1	1
	0.65	114.53	249	1.00012	1.00182	1	1	1	1
100*11	0.50	284.20	935	1.00047	1.00205	1	1	1	1
	0.55	206.37	653	1.00018	1.00256	1	1	1	1
	0.60	270.77	2014	1.00055	1.00452	1	1	1	1
	0.65	219.33	1473	1.00028	1.00187	1	1	1	1
100*12	0.50	321.10	1242	1.00030	1.00288	1	1	1	1
	0.55	496.63	2436	1.00023	1.00188	1	1	1	1
	0.60	759.17	6246	1.00041	1.00455	1	1	1	1
	0.65	449.87	2448	1.00036	1.00248	1	1	1	1
100*13	0.50	961.10	4730	1.00051	1.00246	1	1	1	1
	0.55	772.73	2878	1.00062	1.00336	1	1	1	1
	0.60	2023.67	11464	1.00053	1.00350	1	1	1	1
	0.65	1448.93	9404	1.00079	1.00329	1	1	1	1
100*14	0.50	1696.50	8302	1.00051	1.00387	1	1	1	1
	0.55	2035.20	14327	1.00067	1.00476	1	1	1	1
	0.60	2033.23	18728	1.00050	1.00357	1	1	1	1
	0.65	2696.63	10213	1.00038	1.00505	1	1	1	1

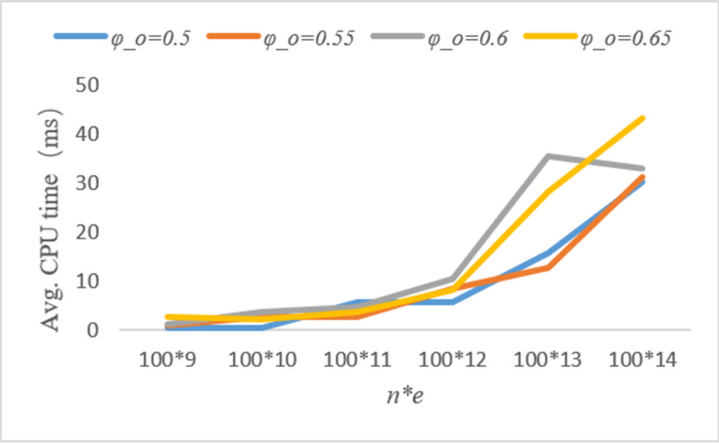


Figure 1. The average CPU time of B&B.

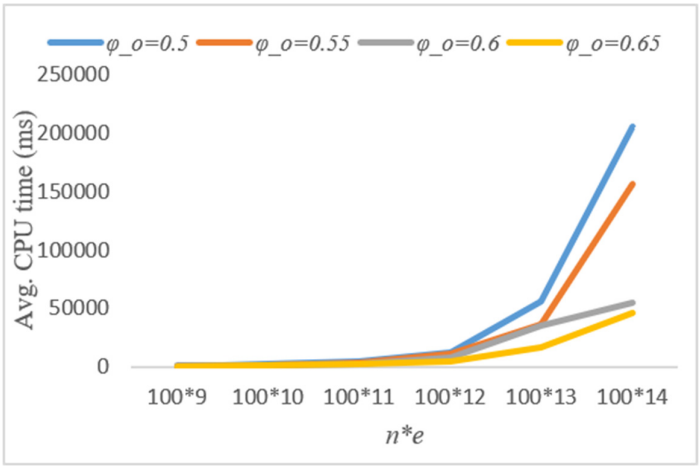


Figure 2. The average CPU time of MINLP.

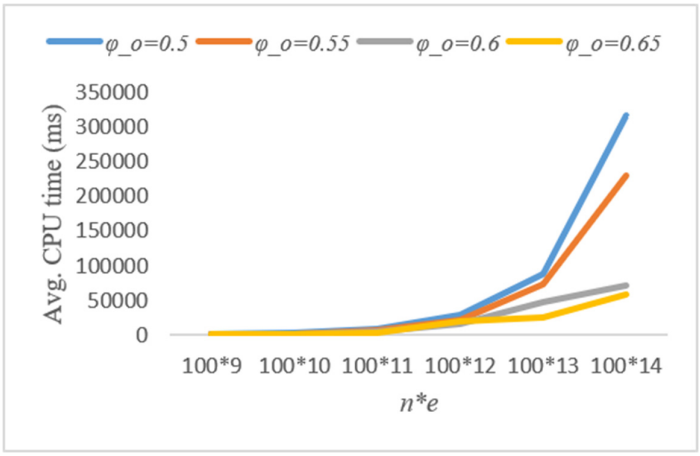


Figure 3. The average CPU time of MILP.

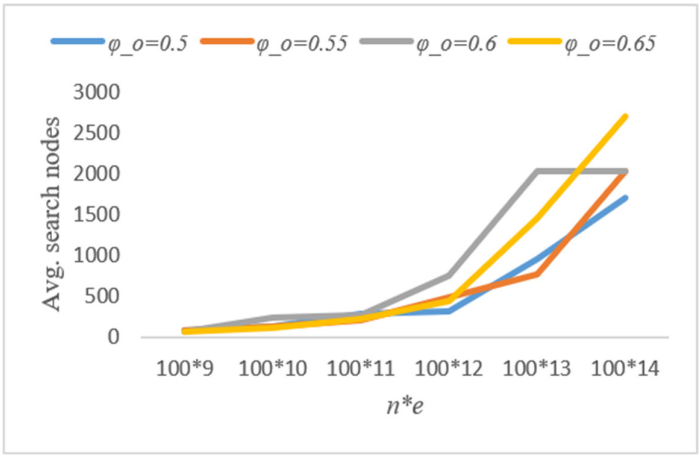


Figure 4. Average search nodes of B&B.

7. Summary and future research

We considered a single-machine group scheduling problem subject to both general deteriorating setup times and learning effects, where truncated exponential functions were incorporated into the learning and deteriorating effects. We proved that the SPT rule determines the optimal job sequence within each group, and we also analyzed a special case to obtain the optimal group sequence. In addition, a heuristic algorithm, a B&B algorithm, and two mathematical models were proposed to solve the problem. Experimental results showed that the B&B algorithm outperformed the two mathematical models in terms of the solution efficiency for the problem under consideration.

Future research can be pursued along the following directions. First, this paper only considers the single-machine environment, whereas multi-machine settings such as flow shops, job shops, and parallel machines are more common in actual manufacturing systems. Subsequent work can extend the proposed model to such multi-machine environments (e.g., flow shops; see Sun et al. [58], Lv and Wang [59], Jin et al. [60], Wang et al. [61], and Pan et al. [62]). Meanwhile, the efficiency of the B&B algorithm heavily relies on the tightness of the lower bound; therefore, designing tighter bounds to enhance pruning performance is a key avenue for future work. In addition, more efficient mathematical programming solution strategies merit further investigation. Second, at the application level, the learning effect model can be extended to more general forms, and the position-dependent truncated deterioration model can also be generalized. Moreover, practical constraints such as machine maintenance, due-window assignment, or multi-objective optimization can be incorporated to strengthen the model's practical relevance and applicability. Overall, this study provides an effective theoretical framework and solution methods for group scheduling problems under generalized effects, and the above directions are expected to drive the field toward scenarios that are more closely aligned with real-world production environments.

Author contributions

Heng Yin constructed the model, performed the theoretical derivations, and designed the algorithms; Heng Yin and Jia-Ming Gu conducted the numerical experiments; Heng Yin wrote the original draft; Ji-Bo Wang reviewed and edited the manuscript.

Use of Generative-AI Tools Declaration

The authors declare they have used Artificial Intelligence (AI) tools in the creation of this article. AI tools used: Deepseek. How were the AI tools used: The AI tool assisted in linguistic polishing and tone refinement of the authors' original content. It was not used to generate research data or scientific conclusions, and all modifications were finalized by the authors. Where in the article is the information located: Throughout the manuscript, primarily for linguistic polishing and formatting consistency.

Acknowledgments

This work was supported by the Science Research Foundation of Educational Department of Liaoning Province [20240218; LJ222510143003].

Conflict of interest

All authors declare no conflicts of interest in this paper.

Data availability

The experimental data and source code are available from the corresponding author upon reasonable request.

References

1. Y. C. Luo, Stochastic single-machine scheduling problems with both time-dependent deterioration and position-dependent learning effect, *J. Comb. Optim*., **50** (2025), 31. https://doi.org/10.1007/s10878-025-01355-7
2. A. Azzouz, M. Ennigrou, L. B. Said, Scheduling problems under learning effects: classification and cartography, *Int. J. Prod. Res.*, **56** (2018), 1642–1661. https://doi.org/10.1080/00207543.2017.1355576
3. S. Gawiejnowicz, A note on scheduling on a single processor with speed dependent on a number of executed jobs, *Inf. Process. Lett.*, **57** (1996), 297–300. https://doi.org/10.1016/0020-0190(96)00021-X
4. G. Mosheiov, Parallel machine scheduling with a learning effect, *J. Oper. Res. Soc.*, **52** (2001), 1165–1169. https://doi.org/10.1057/palgrave.jors.2601215
5. S. Gawiejnowicz, *Models and algorithms of time-dependent scheduling*, 2 Eds., Berlin: Springer, 2020. https://doi.org/10.1007/978-3-662-59362-2
6. D. Y. Bai, H. Y. Xue, L. Wang, C. C. Wu, H. D. Abdulkadir, Effective algorithms for single-machine learning-effect scheduling to minimize completion-time-based criteria with release dates, *Expert Syst. Appl.*, **156** (2020), 113445. https://doi.org/10.1016/j.eswa.2020. 113445
7. X. G. Zhang, Single machine and flowshop scheduling problems with sum-of-processing time based learning phenomenon, *J. Ind. Manag. Optim.*, **16** (2020), 231–244. https://doi.org/10.3934/jimo.2018148
8. S. Muştu, T. Eren, The single machine scheduling problem with setup times under an extension of the general learning and forgetting effects, *Optim. Lett.*, **15** (2021), 1327–1343. https://doi.org/10.1007/s11590-020-01641-9
9. M. Jemmali, L. Hidri, Bounding schemes for the parallel machine scheduling problem with DeJong's learning effect, *J. Parallel Distrib. Comput.*, **156** (2021), 101–118. https://doi.org/10.1016/j.jpdc.2021.05.003
10. Z. Liu, J. B. Wang, Single-machine scheduling with simultaneous learning effects and delivery times, *Mathematics*, **12** (2024), 2522. https://doi.org/ 10.3390/math12162522
11. Y. Sun, H. Y. He, Y. Z. Zhao, J. B. Wang, Minimizing makespan scheduling on a single machine with general positional deterioration effects, *Axioms*, **14** (2025), 290. https://doi.org/10.3390/axioms14040290
12. X. Y. Qiu, J. B. Wang, Single-machine scheduling with mixed due-windows and deterioration effects, *J. Appl. Math. Comput.*, **71** (2025), 2527–2542. https://doi.org/10.1007/s12190-024-02306-x

13. J. B. Wang, J. X. Xu, F. Guo, M. Q. Liu, Single-machine scheduling problems with job rejection, deterioration effects and past-sequence-dependent setup times, *Eng. Optim.*, **54** (2022), 471–486. https://doi.org/10.1080/0305215x.2021.1876041
14. L. M. Meng, R. Ma, Y. Z. Zhang, Single-machine online scheduling with non-delayed processing constraint and deterioration effect in the steel rolling process, *J. Oper. Res. Soc. China*, **14** (2026), 653–670. https://doi.org/10.1007/s40305-024-00552-1
15. X. Y. Sun, X. N. Geng, Single-machine scheduling with deteriorating effects and machine maintenance, *Int. J. Prod. Res.*, **57** (2019), 3186–3199. https://doi.org/10.1080/00207543.2019.1566675
16. D. Y. Lv, J. B. Wang, Research on two-machine flow shop scheduling problem with release dates and truncated learning effects, *Eng. Optim.*, **57** (2025), 1828–1848. https://doi.org/10.1080/0305215X.2024.2372633
17. L. Y. Wang, Due-window assignment scheduling problems with position-dependent weights, truncated learning effects and past-sequence-dependent setup-times, *Symmetry*, **18** (2026), 396. https://doi.org/10.3390/sym18030396
18. X. F. Wu, L. Lin, J. B. Wang, Total weighted completion time scheduling under resource allocation and time-dependent learning effects, *J. Ind. Manag. Optim.*, **22** (2026), 3219–3251. https://doi.org/10.3934/jimo.2026118
19. Q. F. Kong, C. M. Wei, J. B. Wang, Minmax delivery completion time scheduling with delivery times and deteriorating jobs, *Comput. Appl. Math.*, **45** (2026), 276. https://doi.org/10.1007/s40314-026-03684-7
20. Z. W. Sun, D. Y. Lv, P. Ji, J. B. Wang, Minimizing total completion time scheduling problem with ready times and linear deterioration functions of processing times, *J. Appl. Math. Comput.*, **72** (2026), 42. https://doi.org/10.1007/s12190-025-02694-8
21. T. Liu, X. Y. Sun, L. H. Zhang, Y. H. Fan, Coordinating single-machine scheduling with the general linear deterioration function and maintenance activity: minimizing the makespan, *Eng. Optim.*, (2026). https://doi.org/10.1080/0305215X.2026.2645254
22. R. Rudek, A generic optimization framework for scheduling problems under machine deterioration and maintenance activities, *Comput. Ind. Eng.*, **174** (2022), 108800. https://doi.org/10.1016/j.cie.2022.108800
23. J. Gao, J. Zou, Y. Z. Zhang, Single-machine scheduling with a deteriorating maintenance activity and DeJong's learning effect, *Int. J. Found. Comput. Sci.*, **36** (2025), 1–14. https://doi.org/10.1142/S0129054122460030
24. N. Li, R. Ma, Y. Z. Zhang, Single-machine scheduling with the learning effect of processing time and the deterioration effect of delivery time for prefabricated components, *J. Comb. Optim.*, **49** (2025), 37. https://doi.org/10.1007/s10878-025-01271-w
25. M. Parichehreh, H. Gholizadeh, A. M. Fathollahi-Fard, K. Y. Wong, An energy-efficient unrelated parallel machine scheduling problem with learning effect of operators and deterioration of jobs, *Int. J. Environ. Sci. Technol.*, **21** (2024), 9651–9676. https://doi.org/10.1007/s13762-024-05595-8
26. H. R. Song, J. K. Yang, J. B. Wang, Study on multiple due-windows assignment scheduling with learning and deteriorating effects, *Asia-Pac. J. Oper. Res.*, (2026). https://doi.org/10.1142/S0217595926500016

27. Y. Y. Lu, M. H. Li, A unified analysis for single-machine scheduling with resource allocations, learning and deteriorating effects simultaneously, *Comput. Appl. Math.*, **45** (2026), 388. https://doi.org/10.1007/s40314-026-03784-4
28. J. B. Wang, L. Liu, J. J. Wang, L. Li, Makespan minimization scheduling with ready times, group technology and shortening job processing times, *Comput. J.*, **61** (2018), 1422–1428. https://doi.org/10.1093/comjnl/bxy007
29. J. Qian, Y. Zhan, Single-machine group scheduling model with position-dependent and job-dependent DeJong's learning effect, *Mathematics*, **10** (2022), 2454. https://doi.org/10.3390/math10142454
30. F. Liu, J. Yang, Y. Y. Lu, Solution algorithms for single-machine group scheduling with ready times and deteriorating jobs, *Eng. Optim.*, **51** (2019), 862–874. https://doi.org/10.1080/0305215X.2018.1500562
31. Y. Chen, Y. X. Cheng, Group scheduling and due date assignment without restriction on a single machine, *IFIP Adv. Inf. Commun. Tech.*, **632** (2021), 250–257. https://doi.org/10.1007/978-3-030-85906-0_28
32. J. Qian, Note on single machine common flow allowance group scheduling with learning effect and resource allocation, *Comput. Ind. Eng.*, **186** (2023), 109750. https://doi.org/10.1016/j.cie.2023.109750
33. T. Keshavarz, M. Savelsbergh, N. Salmasi, A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties, *Appl. Math. Model.*, **39** (2015), 6410–6424. https://doi.org/10.1016/j.apm.2015.01.069
34. X. G. Zhang, Y. Wang, S. K. Bai, Single-machine group scheduling problems with deteriorating and learning effect, *Int. J. Syst. Sci.*, **47** (2016), 2402–2410. https://doi.org/10.1080/00207721.2014.998739
35. Y. Chen, Y. Xu, G. Q. Zhang, Y. X. Cheng, A single machine group scheduling problem with due date assignment and position-dependent costs, *Asia-Pac. J. Oper. Res.*, **40** (2023), 2340007. https://doi.org/10.1142/S0217595923400079
36. X. Y. Wang, W. G. Liu, Optimal different due-date assignment scheduling with group technology and resource allocation, *Mathematics*, **12** (2024), 436. https://doi.org/10.3390/math12030436
37. L. H. Zhang, N. Yin, Study on single-machine group scheduling with convex resource allocations and different due-date assignments, *Bull. Malays. Math. Sci. Soc.*, **49** (2026), 33. https://doi.org/10.1007/s40840-025-02031-z
38. X. Y. Wang, W. G. Liu, Single machine group scheduling jobs with resource allocations subject to unrestricted due date assignments, *J. Appl. Math. Comput.*, **70** (2024), 6283–6308. https://doi.org/10.1007/s12190-024-02216-y
39. M. H. Li, D. Y. Lv, Y. Y. Lu, J. B. Wang, Scheduling with group technology, resource allocation, and learning effect simultaneously, *Mathematics*, **12** (2024), 1029. https://doi.org/10.3390/math12071029
40. D. Y. Lv, J. B. Wang, Single-machine group technology scheduling with resource allocation and slack due window assignment including minmax criterion, *J. Oper. Res. Soc.*, **76** (2025), 1696–1712. https://doi.org/10.1080/01605682.2024.2430351
41. L. H. Zhang, J. B. Wang, Resource allocation and minmax scheduling under group technology and different due-window assignments, *Axioms*, **14** (2025), 827. https://doi.org/10.3390/axioms14110827

42. L. H. Zhang, M. H. Li, L. Lin, Study on controllable processing time and minmax group scheduling with common due-window assignment, *Symmetry*, **18** (2026), 358. https://doi.org/10.3390/sym18020358
43. X. Huang, H. Y. He, H. B. Bei, Y. Z. Zhao, N. Wang, Y. Chang, Group-scheduling with simultaneous learning effects and convex resource allocations, *Oper. Res. Perspect.*, **15** (2025), 100370. https://doi.org/10.1016/j.orp.2025.100370
44. N. Yin, H. Y. He, Y. Z. Zhao, Y. Chang, N. Wang, Integrating group setup time deterioration effects and job processing time learning effects with group technology in single-machine green scheduling, *Axioms*, **14** (2025), 480. https://doi.org/10.3390/axioms14070480
45. Z. Li, Y. T. Wang, Y. Y. Han, L. L. Meng, K. Z. Gao, Q. D. Chen, Co-evolutionary multi-objective optimization enhanced by reinforcemexpertg decision support in distributed group scheduling, *Expert Syst. Appl.*, **299** (2026), 129953. https://doi.org/10.1016/j.eswa.2025. 129953
46. C. M. Hu, Y. H. Zhai, S. J. Lu, X. B. Liu, Parallel machine group scheduling with time-dependent setup time and combined effects of deterioration and learning, *Memet. Comput.*, **18** (2026). https://doi.org/10.1007/s12293-025-00487-x
47. N. Yin, M. Gao, Single-machine group scheduling with general linear deterioration and truncated learning effects, *Comput. Appl. Math.*, **43** (2024), 386. https://doi.org/ 10.1007/s40314-024-02881-6
48. W. H. Kuo, D. L. Yang, Single-machine group scheduling with a time-dependent learning effect, *Comput. Oper. Res.*, **33** (2006), 2099–2112. https://doi.org/10.1016/j.cor. 2004.11.024
49. C. C. Wu, W. C. Lee, Single-machine group-scheduling problems with deteriorating setup times and job-processing times, *Int. J. Prod. Econ.*, **115** (2008), 128–133. https://doi.org/10.1016/j.ijpe.2008.05.004
50. S. J. Yang, D. L. Yang, Single-machine group scheduling problems under the effects of deterioration and learning, *Comput. Ind. Eng.*, **58** (2010), 754–758. https://doi.org/10.1016/j.cie.2010.02.006
51. W. H. Kuo, Single-machine group scheduling with time-dependent learning effect and position-based setup time learning effect, *Ann. Oper. Res.*, **196** (2012), 349–359. https://doi.org/10.1007/s10479-012-1111-8
52. Y. He, L. Sun, One-machine scheduling problems with deteriorating jobs and position-dependent learning effects under group technology considerations, *Int. J. Syst. Sci.*, **46** (2015), 1319–1326. https://doi.org/10.1080/00207721.2013.822126
53. J. B. Wang, X. Jia, J. X. Yan, S. H. Wang, J. Qian, Single-machine group scheduling problem with makespan objective and a proportional linear shortening, *RAIRO Oper. Res.*, **56** (2022), 1523–1532. https://doi.org/10.1051/ro/2022078
54. L. Ning, L. Sun, Single-machine group scheduling problems with general deterioration and linear learning effects, *Math. Probl. Eng.*, **2023** (2023), 1–9. https://doi.org/10. 1155/2023/1455274
55. R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Ann. Discret. Math.*, **5** (1979), 287–326. https://doi.org/10.1016/S0167-5060(08)70356-X
56. J. B. Wang, Z. W. Sun, M. Gao, Research on single-machine scheduling with due-window assignment and resource allocation under total resource consumption cost is bounded, *J. Appl. Math. Comput.*, **71** (2025), 7905–7927. https://doi.org/10.1007/s12190-025-02599-6.
57. G. H. Hardy, J. E. Littlewood, G. Polya, Inequalities, UK: Cambridge University Press, 1934.

58. X. Y. Sun, X. N. Geng, F. Liu, Flow shop scheduling with general position weighted learning effects to minimise total weighted completion time, *J. Oper. Res. Soc.*, **72** (2021), 2674–2689. https://doi.org/10.1080/01605682.2020.1806746
59. D. Y. Lv, J. B. Wang, No-idle flow shop scheduling with deteriorating jobs and common due date under dominating machines, *Asia-Pac. J. Oper. Res.*, **41** (2024), 2450003. https://doi.org/10.1142/S0217595924500039
60. R. Y. Jin, W. T. Tong, Y. Xu, J. M. Dong, Two-stage flexible flow shop scheduling in GPU systems, *J. Oper. Res. Soc.*, **77** (2026), 793–804. https://doi.org/10.1080/01605682.2025.2504435
61. J. B. Wang, D. Y. Lv, C. Wan, Proportionate flow shop scheduling with job-dependent due windows and position-dependent weights, *Asia-Pac. J. Oper. Res.*, **42** (2025), 2450011. https://doi.org/10.1142/S0217595924500118
62. L. Pan, X. Y. Sun, X. N. Geng, Dynamic programming for proportionate flow-shop scheduling with due-date assignment, *Asia-Pac. J. Oper. Res.*, (2026). https://doi.org/10.1142/S0217595926500028



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)