




Single machine scheduling problems with truncated learning effects and exponential past-sequence-dependent delivery times

Xiao-Yuan Wang^{1,3} · Dan-Yang Lv¹ · Ping Ji² · Na Yin¹ · Ji-Bo Wang¹  · Jin Qian⁴

Received: 8 December 2023 / Revised: 15 February 2024 / Accepted: 27 March 2024 /

Published online: 22 April 2024

© The Author(s) under exclusive licence to Sociedade Brasileira de Matemática Aplicada e Computacional 2024

Abstract

This paper studies the single machine scheduling problems with truncated logarithm processing times and exponential past-sequence-dependent delivery times. We prove that the makespan and total completion time minimizations are polynomially solvable. For the total weighted completion time minimization, we illustrate that it remains polynomially solvable under a special case; under the general case, this paper proposes heuristic, tabu search and branch-and-bound algorithms. Computational experiments indicate that the heuristic algorithm is more effective than tabu search algorithm.

Keywords Scheduling · Learning effect · Single machine · Delivery time · Branch-and-bound algorithm

Mathematics Subject Classification 90B35 · 68M20

1 Introduction

In the actual production and processing environment, workers, machines and other processing conditions have a great impact on the time when the job is handed over to the customer (i.e., the delivery times, see Zhang et al. 2022; Maecker et al. 2023). As the number of processing increases, the processing conditions deteriorate gradually, and the total processing time of the job to the customer increases. This kind of situation related to the previously processed jobs is called past-sequence-dependent delivery time. Koulamas and Kyparisis (2010) considered the single machine scheduling problems with past-sequence-dependent delivery times (denoted

✉ Ji-Bo Wang
wangjibo75@163.com; wangjibo@sau.edu.cn

¹ School of Science, Shenyang Aerospace University, Shenyang 110136, People's Republic of China

² Department of Management Science and Engineering, Khalifa University of Science and Technology, P O Box 127788, Abu Dhabi, UAE

³ School of Management, Xi'an Jiaotong University, Xi'an, People's Republic of China

⁴ Department of Mathematics, Northeastern University, Shenyang 110819, People's Republic of China

by Q_{psddt}), they proved that the $1|Q_{psddt}|\varphi$ problem can be solved in polynomial time, where φ is the objective function, $\varphi \in \{C_{\max}$ (makespan), $TCT (= \sum C_j$, total completion time), L_{\max} (maximum lateness), T_{\max} (maximum tardiness), $\sum U_j$ (number of tardy jobs)}, and C_j is the completion time of job J_j . Liu et al. (2012) showed that the problem $1|Q_{psddt}|\varphi$ can be solved in polynomial time, where $\varphi \in \{TWCT (= \sum v_j C_j$, total weighted completion time, where v_j is the weight of job J_j), $\sum v_j (1 - e^{-\eta C_j})$ ($0 < \eta < 1$, total discounted weighted completion time), $TADC (= \sum_{i=1}^n \sum_{j=i}^n |C_i - C_j|$), $\sum (A_1 E_j + A_2 T_j + A_3 d)$, $E_j = \max\{0, d - C_j\}$ and $T_j = \max\{0, C_j - d\}$ are earliness and tardiness of job J_j , d is the common due date, $A_1 \geq 0$, $A_2 \geq 0$, $A_3 \geq 0$ are given constants (i.e., the unit cost of earliness, tardiness and common due date respectively). Wang et al. (2021) proved that both the problems $1|Q_{psddt}|\sum \varpi_j C_{[j]}$ and $1|Q_{psddt}|\sum \varpi_j (1 - e^{-\eta C_{[j]}})$ can be solved in $O(n \log n)$ time, where ϖ_j is the weight of j th position (position-dependent weight). Qian and Han (2022) integrated the single-machine due date assignment scheduling with Q_{psddt} and simple linear deterioration (SLD). For the earliness-tardiness cost (i.e., $\sum (A_1 E_j + A_2 T_j + A_3 d_j)$, where d_j is the due date of job J_j) minimization, they proved that the problem is polynomially solvable. Qian and Zhan (2022) and Mao et al. (2023) studied the single-machine due window assignment scheduling with Q_{psddt} and SLD. For the earliness-tardiness cost (i.e., $\sum (A_1 E_j + A_2 T_j + A_3 d_j^1 + A_4 D)$, where d_j^1 is the starting time of due window of job J_j , D is the size of common due window, and A_4 is the unit cost of D) minimization, Qian and Zhan (2022) proved that the problem is polynomially solvable. For the general total weighted earliness-tardiness cost minimization, Mao et al. (2023) proved that the problem is polynomially solvable, where the weights are positional-dependent weights.

In addition, a related research area focuses on the learning effects and scheduling (see Azzouz et al. 2018; Wang et al. 2022b; Heuser and Tauer 2023; Lei et al. 2023; Ma et al. 2023; Paredes-Astudillo et al. 2023; Pei et al. 2023; Wang and Wang 2023; Xin et al. 2023). Yang and Yang (2012) explained the single-machine scheduling problems with Q_{psddt} and learning effects. They proved that $1|p_{i,[j]} = p_i j^{a_i}, Q_{psddt}|\sum (A_1 E_j + A_2 T_j + A_3 d_1 + A_4 D)$ and $1|p_{i,[j]} = p_i j^{a_i}, Q_{psddt}|TADC$ can be solved in polynomial time, where $p_{i,[j]} = p_i j^{a_i}$ is the actual processing time of job J_i is scheduled at j th position, and p_i (resp. a_i) is the basic processing time (resp. learning rate) of job J_i . Shen and Wu (2013) considered single-machine scheduling with Q_{psddt} and general learning effects. They proved that some regular objective functions minimizations can be solved in polynomial time. Zhao and Tang (2014) examined the single-machine scheduling problems with Q_{psddt} and general position-dependent processing times. Recently, Toksari et al. (2022) considered the single-machine scheduling problems with general Q_{psddt} and learning effects. For the objective functions C_{\max} and TCT , they proved that the problem remains polynomially solvable. For the special cases of T_{\max} and $TWCT$, they proved that the problem remains polynomially solvable. In 2023, Ren et al. (2023) studied the same problem as Toksari et al. (2022), for the general cases of T_{\max} and $TWCT$, Ren et al. (2023) proposed a branch-and-bound algorithm and some heuristic algorithms. Wang et al. (2023) combined Q_{psddt} and truncated learning effect. They proved that three due date assignment problems can be solved in polynomial time.

However, subjected to the uncontrolled learning effects, the actual job processing times will plummet to zero dramatically due to the increasing number of jobs which are already processed or the emergence of jobs with long processing time. Hence, Wu et al. (2012, 2013), Wu and Wang (2016), Wang et al. (2022a) and Zhao (2022) studied the truncated learning effects. In this article, we extend the results of Toksari et al. (2022), Ren et al. (2023) and Wang et al. (2023), but consider the scheduling model with general Q_{psddt} and truncated learning effects. The objective functions are given as follows: C_{\max} , TCT and $TWCT$. We prove that the C_{\max} and TCT minimizations are polynomial solvable. Under a special

case of the *TWCT* minimization, we prove that it remains polynomially solvable. To solve the general case of the *TWCT* minimization, the heuristic, tabu search and branch-and-bound algorithms are proposed. We also deal with the computational performances of these algorithms for the general case of the *TWCT* minimization.

The structure of this paper is given as follows: Sect. 2 describes the problem; Sect. 3 puts forward several useful properties; Sect. 4 analyzes the single-machine C_{\max} and *TCT* problems; Sect. 5 solves the *TWCT* problem in general case. Conclusions are given in the last section.

2 Problem description

Considering that n jobs are to be processed on a single machine, jobs can be processed at time 0 and can only be processed one job at a time. Let p_i be the basic processing time (i.e., the processing time without any learning effects) of the job $J_i \in \{J_1, J_2, \dots, J_n\}$, Toksari et al. (2022) studied the following model: the actual processing time of job J_i scheduled at position j is

$$p_{i,[j]} = p_i \left(1 + \sum_{s=1}^{j-1} \ln p_{[s]} \right)^\vartheta, \quad (i = 1, \dots, n),$$

where $\vartheta \leq 0$ is the learning index. Wang et al. (2023) studied the following model: the actual processing time of job J_i scheduled at position j is

$$p_{i,[j]} = p_i \max \left\{ \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^\vartheta, \delta \right\}, \quad (i = 1, \dots, n).$$

This paper generalizes the model of Toksari et al. (2022) and Wang et al. (2023), i.e., the logarithm truncation time function with position-dependent weight is:

$$p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]}} + \beta, \delta \right\}, \tag{1}$$

where $0 < a \leq 1, \alpha \geq 0, \beta \geq 0, \alpha + \beta = 1, \delta$ is a truncation parameter ($0 < \delta \leq 1$), and $w_s \geq 1$ is the position-dependent weight ($s = 1, 2, \dots, n$). As in Toksari et al. (2022), the processing of job $p_{i,[j]}$ must be followed the general Q_{psddt} delivery time:

$$Q_{i,[j]} = \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b, \tag{2}$$

where $\gamma > 0$ is a constant and $b \geq 1$ is the index of delivery times. The objective is to minimize C_{\max}, TCT , and *TWCT*. As in Brucker (2007) and Pinedo (2015), the corresponding problem can be expressed as:

$$1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]}} + \beta, \delta \right\} | \varphi,$$

where $\varphi \in \{C_{\max}, TCT, TWCT\}$, and GQ_{psddt} denotes the general delivery time (2).

3 Basic properties

Lemma 1 $\lambda(\alpha a^x + Y^{-1}\beta) - (\alpha a^{w_j \ln \lambda + x} + Y^{-1}\beta) \leq 0$ if $0 < a \leq 1, \lambda \geq 1, Y \geq 0, w_j \geq 1$ and $x \geq 1$.

Proof Let $F(\lambda) = \lambda(\alpha a^x + Y^{-1}\beta) - (\alpha a^{w_j \ln \lambda + x} + Y^{-1}\beta)$, and taking the first and second partial derivative with respect to λ , it follows that

$$F'(\lambda) = (\alpha a^x + Y^{-1}\beta) - \frac{\alpha w_j \ln a \cdot a^{w_j \ln \lambda + x}}{\lambda},$$

$$F''(\lambda) = -\frac{\alpha w_j \cdot w_j \ln a \ln a \cdot a^{w_j \ln \lambda + x} + \alpha w_j \ln a \cdot a^{w_j \ln \lambda + x}}{\lambda^2}.$$

According to the given conditions $0 < a \leq 1, \lambda \geq 1, w_j \geq 1$ and $x \geq 1, F''(\lambda) \leq 0$, that is, $F'(\lambda)$ is a decreasing function with respect to $\lambda \geq 1$, and $F'(\lambda) \leq F'(1) \leq 0$ can be obtained. Hence, $F(\lambda)$ is decreasing function for $0 < a \leq 1, \lambda \geq 1, w_j \geq 1$ and $x \geq 1$, that is

$$F(\lambda) \leq F(1) = (\alpha a^x + Y^{-1}\beta) - (\alpha a^x + Y^{-1}\beta) = 0.$$

□

Lemma 2

$$\left(\begin{array}{l} (\lambda_1 - \lambda_2\lambda) (e^x)^{\frac{1}{w_j}} (\alpha + Y^{-1}\beta) + (\lambda_1 - \lambda_2)\gamma\varphi^b Y^{-1} + \lambda_2\gamma Y^{-1} \left(\varphi + (e^x)^{\frac{1}{w_j}} \right)^b \\ -\lambda_1\gamma Y^{-1} \left(\varphi + \lambda (e^x)^{\frac{1}{w_j}} \right)^b + \lambda_2\lambda (e^x)^{\frac{1}{w_j}} (\alpha a^x + Y^{-1}\beta) - \lambda_1 (e^x)^{\frac{1}{w_j}} (\alpha a^{w_j \ln \lambda + x} + Y^{-1}\beta) \end{array} \right) \leq 0$$

if $0 < a \leq 1, b \geq 1, \lambda \geq 1, 0 \leq \lambda_1 \leq 1, 0 \leq \lambda_2 \leq 1, Y \geq 0, w_j \geq 1$ and $x \geq 1$.

Proof The proof is similar to **Lemma 1**. Let

$$F = \left(\begin{array}{l} (\lambda_1 - \lambda_2\lambda) (e^x)^{\frac{1}{w_j}} (\alpha + Y^{-1}\beta) + (\lambda_1 - \lambda_2)\gamma\varphi^b Y^{-1} + \lambda_2\gamma Y^{-1} \left(\varphi + (e^x)^{\frac{1}{w_j}} \right)^b \\ -\lambda_1\gamma Y^{-1} \left(\varphi + \lambda (e^x)^{\frac{1}{w_j}} \right)^b + \lambda_2\lambda (e^x)^{\frac{1}{w_j}} (\alpha a^x + Y^{-1}\beta) - \lambda_1 (e^x)^{\frac{1}{w_j}} (\alpha a^{w_j \ln \lambda + x} + Y^{-1}\beta) \end{array} \right),$$

and take the first derivative with respect to λ :

$$F'(\lambda) = \left(\begin{array}{l} -\lambda_2 (e^x)^{\frac{1}{w_j}} (\alpha + Y^{-1}\beta) - \lambda_1 b \gamma Y^{-1} (e^x)^{\frac{1}{w_j}} (\varphi + \lambda (e^x)^{\frac{1}{w_j}})^{b-1} \\ +\lambda_2 (e^x)^{\frac{1}{w_j}} (\alpha a^x + Y^{-1}\beta) - \frac{\lambda_1 (e^x)^{\frac{1}{w_j}} \alpha w_j \ln a \cdot a^{w_j \ln \lambda + x}}{\lambda} \end{array} \right)$$

Continue to take the second derivative with respect to λ ,

$$F''(\lambda) = \left(\begin{array}{l} -\lambda_1\gamma Y^{-1} b(b-1) (e^x)^{\frac{2}{w_j}} (\varphi + \lambda (e^x)^{\frac{1}{w_j}})^{b-2} \\ -\frac{\lambda_1 \alpha w_j^2 \ln a \ln a \cdot a^{w_j \ln \lambda + x} - \lambda_1 (e^x)^{\frac{1}{w_j}} \alpha w_j \ln a \cdot a^{w_j \ln \lambda + x}}{\lambda^2} \end{array} \right).$$

Obviously, when $0 < a \leq 1, b \geq 1, \lambda \geq 1, 0 \leq \lambda_1, \lambda_2 \leq 1, w_j \geq 1$ and $x \geq 1$ are satisfied, $F''(\lambda) \leq 0$, that is, $F'(\lambda)$ is a decreasing function of $\lambda \geq 1$, and $F'(\lambda) \leq F'(1) \leq 0$.

Therefore, $F(\lambda)$ is decreasing for $0 < a \leq 1, b \geq 1, \lambda \geq 1, 0 \leq \lambda_1, \lambda_2 \leq 1, w_j \geq 1$ and $x \geq 1$, that is $F(\lambda) \leq F(1) \leq 0$. □

4 Main results

Theorem 4.1 For the problem $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} | C_{\max}$, the optimal sequence can be obtained according to the SPT (smallest processing time first) rule.

Proof Assume that the optimal sequence is $\sigma = [\sigma_1, J_h, J_k, \sigma_2]$, where J_h and J_k meet $p_h \leq p_k$, and J_h is at the j th position of the sequence, and J_k is immediately after J_h . The sequence $\sigma' = [\sigma_1, J_k, J_h, \sigma_2]$ can be obtained by exchanging J_h and J_k .

Let X be the completion time before the job J_h , we have

$$C_{h,[j]}(\sigma) = X + p_h \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b, \tag{3}$$

and

$$\begin{aligned} C_{k,[j+1]}(\sigma) &= X + p_h \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} \\ &+ p_k \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h + \beta}, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b. \end{aligned} \tag{4}$$

For sequence $\sigma' = [\sigma_1, J_k, J_h, \sigma_2]$, we have

$$C_{k,[j]}(\sigma') = X + p_k \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b, \tag{5}$$

and

$$\begin{aligned} C_{h,[j+1]}(\sigma') &= X + p_k \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} \\ &+ p_h \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta}, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b. \end{aligned} \tag{6}$$

According to the truncated learning effects, the problem can be proved in the following two cases:

Case 1. When $\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta} \geq \delta$, the difference between formulas (4) and (6) is

$$\begin{aligned} C_{k,[j+1]}(\sigma) - C_{h,[j+1]}(\sigma') &= (p_h - p_k) \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta} \right) \\ &+ p_k \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h + \beta} \right) \\ &- p_h \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta} \right) \\ &+ \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b - \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b. \end{aligned}$$

From $p_h \leq p_k$, and $(p_h - p_k) \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta} \right) \leq 0$, we have

$$\gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b - \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b \leq 0.$$

Let $\lambda = p_k/p_h$, $x = w_j \ln p_h$ and $Y = a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]}}$, then $\ln \lambda = \ln p_k - \ln p_h$ and $e^x = (p_h)^{w_j}$. According to **Lemma 1**, we have

$$\begin{aligned} & \left(p_k \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h} + \beta \right) - p_h \left(\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k} + \beta \right) \right) / p_h \cdot Y \\ &= \lambda (\alpha a^x + Y^{-1} \beta) - (\alpha a^{w_j \ln \lambda + x} + Y^{-1} \beta) \\ &\leq 0. \end{aligned}$$

To sum up, when $\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k} + \beta \geq \delta$, we have $C_{k,[j+1]}(\sigma) - C_{h,[j+1]}(\sigma') \leq 0$.

Case 2. When $\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k} + \beta < \delta$, let $T = \sum_{s=1}^{j-1} w_s \ln p_{[s]}$, from (4) and (6), we have

$$\begin{aligned} & (C_{k,[j+1]}(\sigma) - C_{h,[j+1]}(\sigma')) / p_h \\ &= \max\{\alpha a^T + \beta, \delta\} - \delta + \lambda \left[\max\{\alpha a^{T+x} + \beta, \delta\} - \max\{\alpha a^T + \beta, \delta\} \right]. \end{aligned} \tag{7}$$

Taking the first derivative of λ , the formula (7) is a decreasing function with respect to $\lambda \geq 1$. In this case, we also have $C_{k,[j+1]}(\sigma) - C_{h,[j+1]}(\sigma') \leq 0$.

From the above cases, the optimal sequence of $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} |C_{\max}$ can be obtained by the *SPT* rule. \square

Theorem 4.2 For the problem $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} |TCT$, the optimal sequence can be obtained by the *SPT* rule.

Proof From **Theorem 4.1**, if $p_h \leq p_k$, we have $C_{h,[j]}(\sigma) \leq C_{k,[j]}(\sigma')$ and $C_{k,[j+1]}(\sigma) \leq C_{h,[j+1]}(\sigma')$, then $C_{h,[j]}(\sigma) + C_{k,[j+1]}(\sigma) \leq C_{k,[j]}(\sigma') + C_{h,[j+1]}(\sigma')$. That is, the optimal sequence of the problem $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} |TCT$ can be obtained by the *SPT* rule. \square

Theorem 4.3 For the problem $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} |TWCT$, if the processing times and weights are anti-agreeable, i.e., $p_h \leq p_k$ implies $v_h \geq v_k$ for all the jobs J_h and J_k , the optimal sequence can be obtained by sequencing the jobs in non-decreasing order of p_h/v_h (*WSPT* rule).

Proof Similar to the above theorems, for sequence $\sigma = [\sigma_1, J_h, J_k, \sigma_2]$, we have

$$\begin{aligned} & v_h C_{h,[j]}(\sigma) + v_k C_{k,[j+1]}(\sigma) \\ &= v_h \left(X + p_h \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b \right) \\ &+ v_k \left(X + p_h \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta}, \delta \right\} \right. \\ &\quad \left. + p_k \max \left\{ \alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h} + \beta, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b \right). \end{aligned}$$

For the sequence $\sigma' = [\sigma_1, J_k, J_h, \sigma_2]$, we have

$$\begin{aligned}
 & v_k C_{k,[j]}(\sigma') + v_h C_{h,[j+1]}(\sigma') \\
 &= v_k \left(X + p_k \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b \right) \\
 & \quad + v_h \left(X + p_k \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta \right\} \right. \\
 & \quad \left. + p_h \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b \right).
 \end{aligned}$$

The proof is similar to **Theorem 4.1** and can be divided into the following two cases:

Case 1. When $\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta \geq \delta$, we have

$$\begin{aligned}
 & TWCT(\sigma) - TWCT(\sigma') \\
 &= (v_h p_h - v_k p_k) \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta \right) + (v_k p_h - v_h p_k) \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta \right) \\
 & \quad + (v_h - v_k) \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b + v_k \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b - v_h \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b \\
 & \quad + v_k p_k \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h + \beta \right) - v_h p_h \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta \right).
 \end{aligned}$$

Let $\lambda_1 = \frac{v_h}{v_h + v_k}$, $\lambda_2 = \frac{v_k}{v_h + v_k}$, $x = w_j \ln p_h$, $Y = \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]}$ and $\varphi = \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b$, then according to **Lemma 2**, we have

$$\begin{aligned}
 & \left(\frac{(v_h p_h - v_k p_k) \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta \right) + (v_h - v_k) \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b}{(v_h + v_k) Y} \right. \\
 & \quad \left. + v_k \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h \right)^b - v_h \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b \right. \\
 & \quad \left. + v_k p_k \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_h + \beta \right) - v_h p_h \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta \right) \right) \\
 &= v_k \left(X + p_k \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} \right)^b \right) \\
 & \quad + v_h \left(X + p_k \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta \right\} \right. \\
 & \quad \left. + p_h \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta, \delta \right\} + \gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k \right)^b \right) \\
 & \leq 0.
 \end{aligned}$$

According to the *WSPT* rule (i.e., $p_h/v_h \leq p_k/v_k$), we have

$$(v_k p_h - v_h p_k) \cdot \left(\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta \right) \leq 0,$$

that is, $v_h C_{h,[j]}(\sigma) + v_k C_{k,[j+1]}(\sigma) \leq v_k C_{k,[j]}(\sigma') + v_h C_{h,[j+1]}(\sigma')$.

Case 2. When $\alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + w_j \ln p_k + \beta < \delta$, similarly, let $T = \sum_{s=1}^{j-1} w_s \ln p_{[s]}$, we have

$$\begin{aligned}
 & TWCT(\sigma) - TWCT(\sigma') \\
 &= (v_h p_h - v_k p_k) \max\{\alpha a^T + \beta, \delta\} + (v_k p_h - v_h p_k) \max\{\alpha a^T + \beta, \delta\}
 \end{aligned}$$

$$\begin{aligned}
 &+(v_h - v_k)\gamma\varphi^b + (v_h - v_k)\gamma\varphi^b \\
 &+v_k\gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_h\right)^b - v_h\gamma \left(1 + \sum_{s=1}^{j-1} p_{[s]} + p_k\right)^b \\
 &+v_k p_k \max\{\alpha a^{T+x} + \beta, \delta\} - v_h p_h \delta - v_h\gamma \left(\varphi + \lambda \left(e^x\right)^{\frac{1}{w_j}}\right)^b.
 \end{aligned}$$

According to the above formula, we have

$$\begin{aligned}
 &\frac{\left((v_h p_h - v_k p_k) \max\{\alpha a^T + \beta, \delta\} + v_k p_k \max\{\alpha a^{T+x} + \beta, \delta\} \right.}{(v_h + v_k)p_h} \\
 &\quad \left. - v_h p_h \delta - v_h\gamma \left(\varphi + \lambda \left(e^x\right)^{\frac{1}{w_j}}\right)^b \right) \\
 &= \lambda_1(1 - \delta) - \lambda_1 \left(e^x\right)^{\frac{1}{w_j}} \gamma \left(\varphi + \lambda \left(e^x\right)^{\frac{1}{w_j}}\right)^b.
 \end{aligned}$$

Similarly, by taking the derivative of this formula with respect to λ , it can be obtained that this formula is a decreasing function with respect to $\lambda \geq 1$, and

$$\lambda_1(1 - \delta) - \lambda_1 \left(e^x\right)^{\frac{1}{w_j}} \gamma \left(\varphi + \lambda \left(e^x\right)^{\frac{1}{w_j}}\right)^b \leq 0$$

can be easily obtained. According to the *WSPT* rule, if $p_h/v_h \leq p_k/v_k$, we have $(v_k p_h - v_h p_k) \max\{\alpha a^T + \beta, \delta\} \leq 0$, that is, $v_h C_{h,[j]}(\sigma) + v_k C_{k,[j+1]}(\sigma) \leq v_k C_{k,[j]}(\sigma') + v_h C_{h,[j+1]}(\sigma')$.

In summary, the theorem holds. □

5 General case of TWCT

For the general case, the complexity of the $1|GQ_{psddt}, P_{i,[j]} = p_i \max\{\alpha a^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta}\}|TWCT$ problem is still open. We will propose the following heuristic, Tabu search and branch-and-bound algorithms to solve this problem.

5.1 Heuristic algorithm

According to Theorem 4.3 and Nawaz et al. (1983), the following heuristic algorithm can be proposed.

Algorithm 1 (H-A)

Part 1.

- Step 1.** Sequence the jobs by the *SPT* rule.
- Step 2.** Sequence the jobs in non-increasing order of v_h .
- Step 3.** Sequence the jobs by the *WSPT* rule.
- Step 4.** Choose the better solution from Steps 1–3.

Part 2.

- Step 1.** Let σ_0 be the sequence obtained in **Part 1**.

Step 2. Set $t = 2$, select the first two jobs in σ_0 , and select the partial sequence with the best objective function value among the two possible sequences.

Step 3. Iterate t in sequence, that is, $t = t + 1$, select the t th job, insert it in all possible t positions, and select a sequence from t sequences to minimize $TWCT$. Next, by exchanging the jobs at positions j and i , where $1 \leq j < t, j < i \leq t$. A better partial sequence is selected from $t(t - 1)/2$ sequences to minimize the value of $TWCT$.

Step 4. If $t = n$, stop the algorithm; otherwise, go back to Step 3.

Since the time of Part 1 is $O(n)$ and the time of Part 2 is $O(n^2)$ (see Nawaz et al. (1983)), then the time of Algorithm 1 is $O(n^2)$.

5.2 Tabu search algorithm

Problem 1| $GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha \alpha^{\sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta} \right\}$ | $TWCT$ can also be solved by tabu search algorithm. Let the initial sequence can be sorted by the order of increasing p_h/v_h , and let the maximum number of iterations be $100n$. The steps of tabu search algorithm are as follows:

Algorithm 2 (T-S)

Step 1. Let the tabu list empty and the iteration number be 0.

Step 2. The initial sequence can be obtained according to the WSPT rule, and calculate the value of $TWCT$, and let the current sequence is the optimal sequence σ^* .

Step 3. Search the relevant neighborhood of the current sequence, and determine whether there is a sequence σ with the smallest objective function in the relevant neighborhood, and the sequence is not in the tabu list.

Step 4. If $TWCT(\sigma) < TWCT(\sigma^*)$, then set $\sigma^* = \sigma$, and update the tabu list and the number of iteration.

Step 5. If there is no sequence in the associated neighborhood, but it is not in the tabu list, or the given maximum number of iteration is reached, then the final sequence is output. Otherwise, update the tabu list and turn Step 3.

5.3 Branch-and-bound algorithm

Let $\sigma = [\sigma_v, \sigma_u]$ be a job sequence, where σ_v is the scheduled part, and σ_u is the unscheduled part. Suppose that there are l jobs in σ_v , then the completion time for the $(l + 1)$ th job in the unscheduled part is:

$$C_{[l+1]} = C_{[l]} + p_{[l+1]} \max \left\{ \alpha \alpha^{\sum_{s=1}^l w_s \ln p_{[s]} + \beta, \delta} \right\} + \gamma \left(1 + \sum_{s=1}^l p_{[s]} \right)^b,$$

and for the $(l + 2)$ th job, the completion time is:

$$C_{[l+2]} = C_{[l]} + p_{[l+1]} \max \left\{ \alpha \alpha^{\sum_{s=1}^l w_s \ln p_{[s]} + \beta, \delta} \right\} + p_{[l+2]} \max \left\{ \alpha \alpha^{\sum_{s=1}^l w_s \ln p_{[s]} + w_{l+1} \ln p_{[l+1]} + \beta, \delta} \right\} + \gamma \left(1 + \sum_{s=1}^l p_{[s]} + p_{[l+1]} \right)^b.$$

By analogy, the formula for the completion time of the n th job can be obtained as follows:

$$C_{[n]} = C_{[l]} + p_{[l+1]} \max \left\{ \alpha \alpha^{\sum_{s=1}^l w_s \ln p_{[s]} + \beta, \delta} \right\} + \dots$$

$$+ p_{[n]} \max \left\{ \alpha a^{\sum_{s=1}^l w_s \ln p_{[s]} + \sum_{s=l+1}^{n-1} w_s \ln p_{[s]} + \beta, \delta} \right\} + \gamma \left(1 + \sum_{s=1}^l p_{[s]} + \sum_{s=l+1}^{n-1} p_{[s]} \right)^b,$$

then the objective function can be expanded as follows:

$$TWCT = \sum_{i=1}^l v_i C_i + \sum_{i=l+1}^n v_{[i]} C_{[i]} = \sum_{i=1}^l v_i C_i + \sum_{i=l+1}^n \left[(v_{[i]} + \dots + v_{[n]}) p_{[i]} \max \left\{ \alpha a^{\sum_{s=1}^l w_s \ln p_{[s]} + \sum_{s=l+1}^{i-1} w_s \ln p_{[s]} + \beta, \delta} \right\} + v_{[i]} \gamma \left(1 + \sum_{s=1}^l p_{[s]} + \sum_{s=l+1}^{i-1} p_{[s]} \right)^b \right], \tag{8}$$

where $\sum_{i=1}^l v_i C_i$ is known, and the second item is the function value of unsorted part. Let $v_{\min} = \min\{v_j | j \in \sigma_u\}$, from Theorems 1 and 2, a lower bound of formula (8) can be obtained by:

$$LB_1 = \sum_{i=1}^l v_i C_i + \sum_{i=l+1}^n \left((n - i + 1) v_{\min} p_{(i)} \max \left\{ \alpha a^{\sum_{s=1}^l w_s \ln p_s + \sum_{s=l+1}^{i-1} w_s \ln p_{(s)} + \beta, \delta} \right\} + v_{\min} \gamma \left(1 + \sum_{s=1}^l p_s + \sum_{s=l+1}^{i-1} p_{(s)} \right)^b \right), \tag{9}$$

where $p_{(l+1)} \leq p_{(l+2)} \leq \dots \leq p_{(n)}$.

Let $p_{\min} = \min\{p_j | j \in \sigma_u\}$, then the second lower bound can be obtained as follows:

$$LB_2 = \sum_{i=1}^l v_i C_i + \sum_{i=l+1}^n \left((v_{(i)} + \dots + v_{(n)}) p_{\min} \max \left\{ \alpha a^{\sum_{s=1}^l w_s \ln p_s + \sum_{s=l+1}^{i-1} w_s \ln p_{\min} + \beta, \delta} \right\} + v_{(i)} \gamma \left(1 + \sum_{s=1}^l p_s + (i - l - 1) p_{\min} \right)^b \right), \tag{10}$$

where $v_{(l+1)} \geq v_{(l+2)} \geq \dots \geq v_{(n)}$.

When $p_{<l+1>} \leq p_{<l+2>} \leq \dots \leq p_{<n>}$ and $v_{(l+1)} \geq v_{(l+2)} \geq \dots \geq v_{(n)}$, ($p_{<s>}$ and $v_{(s)}$ do not necessarily correspond to the same job), from Theorem 3, the following lower bound can be obtained:

$$LB_3 = \sum_{i=1}^l v_i C_i + \sum_{i=l+1}^n \left((v_{(i)} + \dots + v_{(n)}) p_{<i>} \max \left\{ \alpha a^{\sum_{s=1}^l w_s \ln p_s + \sum_{s=l+1}^{i-1} w_s \ln p_{<s>} + \beta, \delta} \right\} + (v_{(i)} + \dots + v_{(n)}) \gamma \left(1 + \sum_{s=1}^l p_s + \sum_{s=l+1}^{i-1} p_{<s>} \right)^b \right). \tag{11}$$

Now, in order to get a tighter lower bound, the maximum value of (9), (10) and (11) can be taken as the lower bound, namely:

$$LB = \max\{LB_1, LB_2, LB_3\}. \tag{12}$$

The following branch-and-bound algorithm is proposed to specifically solve the problem $1|GQ_{psddt}, p_{i,[j]} = p_i \max \left\{ \alpha a \sum_{s=1}^{j-1} w_s \ln p_{[s]} + \beta, \delta \right\} |TWCT$.

Algorithm 3 (B-B)

Step 1. The optimal sequence obtained according to the above **Algorithm 1** is taken as the initial sequence, and the objective function value is taken as the upper bound.

Step 2. Calculate the lower bound of each node according to (12). If the lower bound of the unfinished part is greater than or equal to the objective function value of the initial solution, the node and all nodes after the node are deleted. Calculate the objective function value of the completed sequence. If it is less than the initial solution, the sequence replaces the initial sequence, otherwise the sequence is deleted.

Step 3. Traverse all nodes.

5.4 An example for B-B algorithm

Now consider an example with $n = 5$, the processing time vector and weight vector are given as follows: $(p_1, p_2, p_3, p_4, p_5) = (12, 4, 7, 11, 6)$, $(v_1, v_2, v_3, v_4, v_5) = (4, 2, 5, 3, 6)$, and $(w_1, w_2, w_3, w_4, w_5) = (1, 1, 1, 1, 1)$. And $\alpha = \beta = 0.5$, $\gamma = \delta = 0.5$ and $a = 0.8$, and $b = 1$. The example is first solved according to **Algorithm 1 (H-A)**, in order to obtain the initial sequence required by the **B-B Algorithm**.

Part 1.

Step 1. The sequence $\sigma_1 = [J_2, J_5, J_3, J_4, J_1]$ is obtained according to *SPT* rule, and the objective function value is $TWCT = 462.0505$.

Step 2. According to the descending order of v_h , the sequence $\sigma_2 = [J_5, J_3, J_1, J_4, J_2]$ is obtained. And the corresponding value of objective function is $TWCT = 442.7471$.

Step 3. The sequence $\sigma_3 = [J_5, J_3, J_2, J_1, J_4]$ is obtained according to p_h/v_h increasing order, and the objective function value is $TWCT = 418.0759$.

Step 4. By comparison, the sequence corresponding to the value of the minimum objective function is $\sigma_3 = [J_5, J_3, J_2, J_1, J_4]$, that is, $\sigma_0 = [J_5, J_3, J_2, J_1, J_4]$.

Part 2.

Step 1. Let $\sigma_0 = [J_5, J_3, J_2, J_1, J_4]$ be the initial sequence.

Step 2. Set $t = 2$, the two possible sequences are $[J_5, J_3, J_2, J_1, J_4]$ and $[J_3, J_5, J_2, J_1, J_4]$, and the corresponding objective function values are $TWCT = 418.0759$ and $TWCT = 436.3744$, respectively. Then the better partial sequence is $[J_5, J_3, J_2, J_1, J_4]$.

Step 3. Set $t = 3$, and the three possible sequences are $[J_2, J_5, J_3, J_1, J_4]$, $[J_5, J_2, J_3, J_1, J_4]$, and $[J_5, J_3, J_2, J_1, J_4]$, separately. And the corresponding objective function values are $TWCT = 435.7654$, $TWCT = 422.8465$ and $TWCT = 418.0759$, respectively. Then the better one is $[J_5, J_3, J_2, J_1, J_4]$.

Now, swap the jobs at positions 1 and 2, 1 and 3; and 2 and 3 to obtained the sequences $[J_3, J_5, J_2, J_1, J_4]$, $[J_2, J_3, J_5, J_1, J_4]$ and $[J_5, J_2, J_3, J_1, J_4]$, and the corresponding objective function values are $TWCT = 436.3744$, $TWCT = 452.1878$ and $TWCT = 422.8465$. After comparison, the better sequence is $[J_5, J_3, J_2, J_1, J_4]$.

Step 4. Carry out the above step for the remaining jobs, and the optimal sequence can be obtained as: $[J_5, J_3, J_2, J_1, J_4]$, and the objective function value is $TWCT = 418.0759$.

Then the objective function value $TWCT = 418.0759$ corresponding to the sequence is used as the upper bound of the following **Algorithm 3 (B-B)**. For the given parameters of the problem, according to **Algorithm 3 (B-B)**, the following search tree can be obtained,

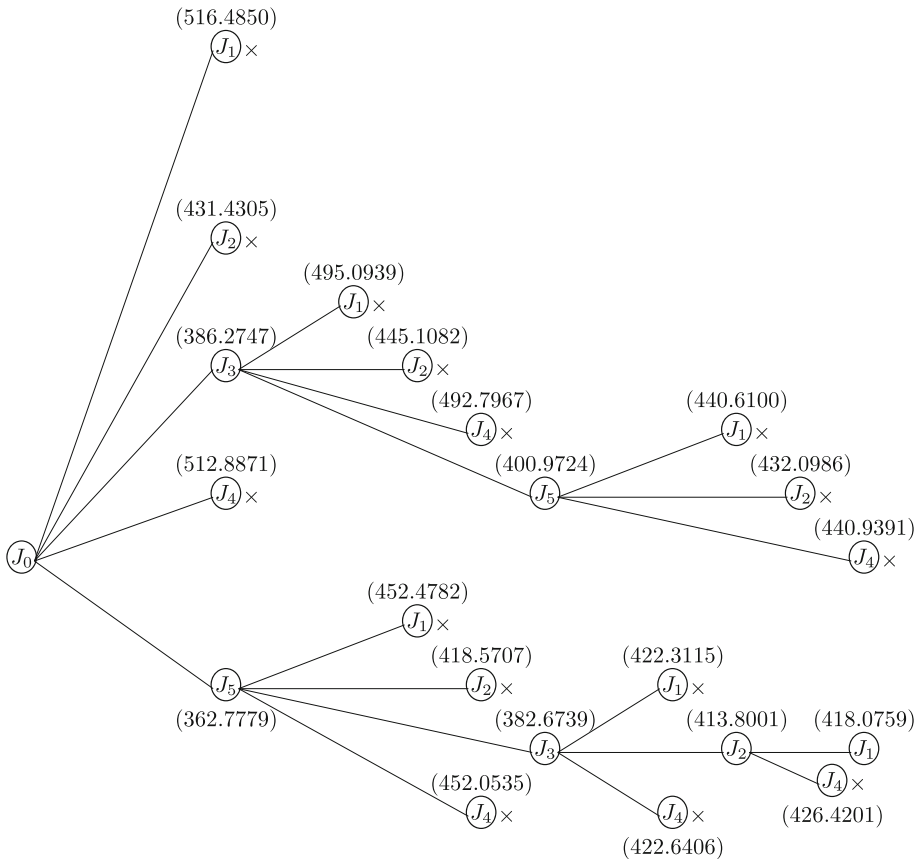


Fig. 1 Search tree of the Algorithm 3 (B-B) for the example (× denotes pruning)

which is defined as Fig. 1. The numbers in Fig. 1 represent the values of lower bound, and J_0 is defined as the level 0.

As level 1, i.e., $j = 1$, for the job J_1 , the lower bounds LB_1 , LB_2 and LB_3 can be obtained as follows according to formulas (9), (10) and (11), respectively:

$$\begin{aligned}
 LB_1 &= \sum_{i=1}^1 v_i C_i + \sum_{i=2}^5 \left((5 - i + 1)v_{\min} p_{(i)} \max \left\{ \alpha a^{\sum_{s=1}^1 \ln p_s + \sum_{s=2}^{i-1} \ln p_{[s]}} + \beta, \delta \right\} \right. \\
 &\quad \left. + v_{\min} \gamma \left(1 + \sum_{s=1}^1 p_s + \sum_{s=2}^{i-1} p_{(s)} \right)^b \right) \\
 &= 12.5 \times 4 + 21.6487 \times 2 + 27.9133 \times 2 + 35.4025 \times 2 + 45.4094 \times 2 = 310.7478. \\
 LB_2 &= \sum_{i=1}^1 v_i C_i + \sum_{i=2}^5 \left((v_{(i)} + \dots + v_{(5)}) p_{\min} \max \left\{ \alpha a^{\sum_{s=1}^1 \ln p_s + \sum_{s=2}^{i-1} \ln p_{\min}} + \beta, \delta \right\} \right. \\
 &\quad \left. + v_{(i)} \gamma \left(1 + \sum_{s=1}^1 p_s + (i - 1 - 1)p_{\min} \right)^b \right) \\
 &= 12.5 \times 4 + 21.6487 \times 6 + 26.4918 \times 5 + 31.1106 \times 3 + 35.5647 \times 2 = 476.8124. \\
 LB_3 &= \sum_{i=1}^1 v_i C_i + \sum_{i=2}^5 \left((v_{(i)} + \dots + v_{(5)}) p_{<i>} \max \left\{ \alpha a^{\sum_{s=1}^1 w_s \ln p_s + \sum_{s=2}^{i-1} w_s \ln p_{<s>}} + \beta, \delta \right\} \right. \\
 &\quad \left. + (v_{(i)} + \dots + v_{(5)}) \gamma \left(1 + \sum_{s=1}^1 p_s + \sum_{s=2}^{i-1} p_{<s>} \right)^b \right) \\
 &= 12.5 \times 4 + 21.6487 \times 6 + 27.9133 \times 5 + 35.4025 \times 3 + 45.4094 \times 2 = 516.4850.
 \end{aligned}$$

Then the lower bound $LB = 516.4850$ can be obtained according to $LB = \max\{LB_1, LB_2, LB_3\}$. And the calculation process of other nodes is similar to the calculation process of this node. Then from the Fig. 1, the optimal sequence is $[J_5, J_3, J_2, J_1, J_4]$, and the optimal value of the objective function is $TWCT = 418.0759$.

5.5 Data simulation

The H-A algorithm (i.e. Algorithm 1), T-S algorithm (i.e. Algorithm 2), and B-B algorithm (i.e. Algorithm 3) were programmed in CLion 2020.x64 and carried out on a CPU Intel(R) Core(TM) i5-10500 3.10GHz, 8GB RAM with Windows 10 operating system. The number of jobs $n = 15, 16, 17, 18, 19, 20$ were tested. The parameters setting can be obtained as follows: $w_s = 1 (s = 1, \dots, n)$, $\alpha = \beta = 0.5$, $\gamma = \delta = 0.5$, $a = 0.8$ and $b = 1$, and the range of processing time, delivery time and weight are $p_j \in [1, 100]$, $q_j \in [1, 100]$ and $v_j \in [1, 100]$, separately.

For Algorithm B-B, set the maximum CPU time at 3600 s for each instance of the number of jobs. And the percentage relative error of the solution produced by H-A algorithm and B-B algorithm is calculated as

$$\frac{TWCT(\text{H-A}) - TWCT(\text{B-B})}{TWCT(\text{B-B})} \times 100\%,$$

and the percentage relative error of the solution produced by T-S algorithm and B-B algorithm is

$$\frac{TWCT(\text{T-S}) - TWCT(\text{B-B})}{TWCT(\text{B-B})} \times 100\%,$$

where $TWCT(\text{H-A})$, $TWCT(\text{B-B})$ and $TWCT(\text{T-S})$ are the value of objective function $TWCT$ generated by Algorithm 1, Algorithm 3 and Algorithm 2, separately. The experimental results are shown in Table 1. It can be seen from table 1 that when $n \geq 20$, the average CPU time of B-B algorithm is 3600 s, and the CPU time of T-S algorithm is greater than that of H-A algorithm. From Table 1, it can be seen that H-A algorithm is better, and the maximum error is less than 0.9%.

6 Conclusion

In this paper, single machine scheduling problems with delivery times and the truncated logarithm processing time have been studied. The problems include makespan, total completion time, and total weighted completion time, and the results showed that the problems of minimizing the makespan and the total completion time can be solved in polynomial time. In addition, the general problem of minimizing the total weighted completion time is analyzed in details, and the heuristic algorithm (i.e., Algorithm 1), tabu search algorithm (i.e., Algorithm 2) and branch-and-bound algorithm (i.e., Algorithm 3) were proposed. Computational study demonstrated that H-A algorithm is more effective than T-S algorithm. Future research can consider (1) scheduling with group technology (Liu and Wang 2023); (2) job-rejection scheduling with setup times (Mor and Shapira 2020; Liu et al. 2024); (3) scheduling with deterioration effects (Sun et al. 2011; Wang et al. 2011; Sun et al. 2023; Lv and Wang 2024; Lv et al. 2024; Wang et al. 2024c; Zhang et al. 2024); (4) scheduling with position-dependent weights (Wang et al. 2024a); flow shop or parallel machines scheduling (Wang et al. 2011;

Table 1 Results of Algorithms

<i>n</i>	(H-A) CPU time (s)		(T-S) CPU time (s)		(B-B) CPU time (s)		Error percentage of H-A Algorithm(%)		Error percentage of T-S Algorithm(%)		Node number of B-B Algorithm	
	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max
15	0.0016	0.0031	0.2415	0.2511	8.3301	24.575	0.18762	0.33564	2.89764	7.34792	1,855,389	5,356,960
16	0.0018	0.0023	0.3064	0.3137	33.7175	149.981	0.35166	0.58964	1.59986	8.58971	5,774,627	26,440,393
17	0.003	0.0055	0.3839	0.4005	379.655	2768.59	0.05646	0.16874	4.47125	11.12427	48,365,076	448,976,138
18	0.0038	0.0061	0.5178	0.5252	1067.543	3600	0.55125	0.86338	3.23347	7.98788	76,892,770	515,072,168
19	0.0055	0.0087	0.6412	0.6653	2596.559	3600	0.00469	0.12354	5.98664	12.64582	324,389,920	667,609,772
20	0.004	0.0056	0.7713	0.8121	3600	3600	0.10918	0.42754	3.47866	9.54771	355,047,372	771,080,634

Wang and Wang 2012; Wang et al. 2019; Sun et al. 2021; Kovalev et al. 2023; Wang et al. 2024b) with GQ_{psddt} .

Funding This work was supported by the Science Research Foundation of Educational Department of Liaoning Province (LJKMZ20220532).

Data Availability The data used to support the findings of this paper are available from the corresponding author upon request.

Declarations

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

References

- Azzouz A, Ennigrou M, Said LB (2018) Scheduling problems under learning effects: classification and cartography. *Int J Prod Res* 56(4):1642–1661
- Brucker P (2007) Scheduling algorithms, 2nd edn. Springer, Heidelberg
- Heuser P, Tauer B (2023) Single-machine scheduling with product category-based learning and forgetting effects. *Omega* 115:102786
- Koulamas C, Kyparisis GJ (2010) Single-machine scheduling problems with past-sequence-dependent delivery times. *Int J Prod Econ* 126:264–266
- Kovalev S, Chalamon I, Bécuwe A (2023) Single machine scheduling with resource constraints: equivalence to two-machine flow-shop scheduling for regular objectives. *J Oper Res Soc.* <https://doi.org/10.1080/01605682.2023.2244529>
- Lei W, Sun L, Ren N, Jia X, Wang J-B (2023) Research on delivery times scheduling with sum of logarithm processing times-based learning effect. *Asia-Pac J Oper Res.* <https://doi.org/10.1142/S0217595923500148>
- Liu WG, Wang X (2023) Group technology scheduling with due-date assignment and controllable processing times. *Processes* 11:1271
- Liu M, Zheng F, Chu C, Xu Y (2012) New results on single-machine scheduling with past-sequence-dependent delivery times. *Theoret Comput Sci* 438:55–61
- Liu WG, Wang X, Li L, Dai W (2024) Due-window assignment scheduling with job-rejection, truncated learning effects and setup times. *J Ind Manag Optim* 20(1):313–324
- Lv D-Y, Wang J-B (2024) No-idle flow shop scheduling with deteriorating jobs and common due date under dominating machines. *Asia-Pac J Oper Res.* <https://doi.org/10.1142/S0217595924500039>
- Lv Z-G, Zhang L-H, Wang X-Y, Wang J-B (2024) Single machine scheduling proportionally deteriorating jobs with ready times subject to the total weighted completion time minimization. *Mathematics* 12:610
- Ma R, Guo SA, Zhang XY (2023) An optimal online algorithm for single-processor scheduling problem with learning effect. *Theoret Comput Sci* 928:1–12
- Maecker S, Shen LJ, Mönch L (2023) Unrelated parallel machine scheduling with eligibility constraints and delivery times to minimize total weighted tardiness. *Comput Oper Res* 149:105999
- Mao R-R, Wang Y-C, Lv D-Y, Wang J-B, Lu Y-Y (2023) Delivery times scheduling with deterioration effects in due window assignment environments. *Mathematics* 11:3983
- Mor B, Shapira D (2020) Scheduling with regular performance measures and optional job rejection on a single machine. *J Oper Res Soc* 71(8):1315–1325
- Nawaz M, Ensore JEE, Ham I (1983) A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* 11:91–95
- Pan L, Sun X, Wang J-B, Zhang L-H, Lv D-Y (2023) Due date assignment single-machine scheduling with delivery times, position-dependent weights and deteriorating jobs. *J Comb Optim* 45:100
- Paredes-Astudillo YA, Botta-Genoulaz V, Montoya-Torres JR (2023) Impact of learning effect modelling in flowshop scheduling with makespan minimisation based on the Nawaz-Ensore-Ham algorithm. *Int J Prod Res.* <https://doi.org/10.1142/10.1080/00207543.2023.2204967>
- Pei J, Zhou Y, Yan P, Pardalos PM (2023) A concise guide to scheduling with learning and deteriorating effects. *Int J Prod Res* 61:2010–2031
- Pinedo ML (2015) Scheduling theory, algorithms, and systems, 5th edn. Springer, Heidelberg
- Qian J, Han H (2022) The due date assignment scheduling problem with the deteriorating jobs and delivery time. *J Appl Math Comput* 68:2173–2186

- Qian J, Zhan Y (2022) The due window assignment problems with deteriorating job and delivery time. *Mathematics* 10:1672
- Ren N, Lv D-Y, Wang J-B, Wang X-Y (2023) Solution algorithms for single-machine scheduling with learning effects and exponential past-sequence-dependent delivery times. *J Ind Manag Optim* 19:8429–8450
- Shen L-X, Wu Y-B (2013) Single machine past-sequence-dependent delivery times scheduling with general position-dependent and time-dependent learning effects. *Appl Math Model* 37:5444–5451
- Sun L-H, Sun L-Y, Wang J-B (2011) Single-machine scheduling to minimize total absolute differences in waiting times under simple linear deterioration. *J Oper Res Soc* 62:768–775
- Sun X, Geng X-N, Liu F (2021) Flow shop scheduling with general position weighted learning effects to minimise total weighted completion time. *J Oper Res Soc* 72(12):2674–2689
- Sun X, Liu T, Geng X-N, Hu Y, Xu J-X (2023) Optimization of scheduling problems with deterioration effects and an optional maintenance activity. *J Sched* 26:251–266
- Toksari MD, Aydogan EK, Atalay B, Sari S (2022) Some scheduling problems with sun of logarithm processing times based learning effect and exponential past sequence dependent delivery times. *J Ind Manag Optim* 18:1795–1807
- Wang J-B, Wang M-Z (2012) Worst-case analysis for flow shop scheduling problems with an exponential learning effect. *J Oper Res Soc* 63:130–137
- Wang Y-C, Wang J-B (2023) Study on convex resource allocation scheduling with a time-dependent learning effect. *Mathematics* 11:3179
- Wang J-J, Wang J-B, Liu F (2011) Parallel machines scheduling with a deteriorating maintenance activity. *J Oper Res Soc* 62:1898–1902
- Wang J-B, Wang J-J, Ji P (2011) Scheduling jobs with chain precedence constraints and deteriorating jobs. *J Oper Res Soc* 62:1765–1770
- Wang J-B, Liu F, Wang J-J (2019) Research on m-machine flow shop scheduling with truncated learning effects. *Int Trans Oper Res* 26(3):1135–1151
- Wang J-B, Cui B, Ji P, Liu W-W (2021) Research on single-machine scheduling with position-dependent weights and past-sequence-dependent delivery times. *J Comb Optim* 41:290–303
- Wang X, Liu WG, Li L, Zhao P, Zhang R (2022) Resource dependent scheduling with truncated learning effects. *Math Biosci Eng* 19(6):5957–5967
- Wang J-B, Zhang L-H, Lv Z-G, Lv D-Y, Geng X-N, Sun X (2022) Heuristic and exact algorithms for single-machine scheduling problems with general truncated learning effects. *Comput Appl Math* 41(8):417
- Wang S-H, Lv D-Y, Wang J-B (2023) Research on position-dependent weights scheduling with delivery times and truncated sum-of-processing-times-based learning effect. *J Ind Manag Optim* 19:2824–2837
- Wang J-B, Bao H, Wang C (2024) Research on multiple slack due-date assignments scheduling with position-dependent weights. *Asia-Pac J Oper Res* 41:2350039. <https://doi.org/10.1142/10.1142/S0217595923500392>
- Wang D, Fang K, Luo W, Ouyang W (2024) Approximation algorithms for mixed batch scheduling on parallel machines. *J Oper Res Soc*. <https://doi.org/10.1080/01605682.2024.2314251>
- Wang J-B, Wang Y-C, Wan C, Lv D-Y, Zhang L (2024) Controllable processing time scheduling with total weighted completion time objective and deteriorating jobs. *Asia-Pac J Oper Res* 41:2350026. <https://doi.org/10.1142/S0217595923500264>
- Wu Y-B, Wang J-J (2016) Single-machine scheduling with truncated sum-of-processing-times based learning effect including proportional delivery times. *Neural Comput Appl* 27:937–943
- Wu C-C, Yin Y, Wu W-H, Cheng S-R (2012) Some polynomial solvable single-machine scheduling problems with a truncation sum-of-processing-times based learning effect. *Eur J Ind Eng* 6:441–453
- Wu C-C, Yin Y, Cheng S-R (2013) Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions. *J Oper Res Soc* 64:147–156
- Xin X, Jiang Q, Li C, Li S, Chen K (2023) Permutation flow shop energy-efficient scheduling with position-based learning effect. *Int J Prod Res* 61:382–409
- Yang S-J, Yang D-L (2012) Single-machine scheduling problems with past-sequence-dependent delivery times and position-dependent processing times. *J Oper Res Soc* 63:1508–1515
- Zhang C, Li Y, Cao J, Yang Z, Coelho LC (2022) Exact and matheuristic methods for the parallel machine scheduling and location problem with delivery time and due date. *Comput Oper Res* 147:105936
- Zhang L-H, Geng X-N, Xue J, Wang J-B (2024) Single machine slack due window assignment and deteriorating jobs. *J Ind Manag Optim* 20:1593–1614
- Zhao S (2022) Scheduling jobs with general truncated learning effects including proportional setup times. *Comput Appl Math* 41(4):146
- Zhao C-L, Tang H-Y (2014) Single machine scheduling problems with general position-dependent processing times and past-sequence-dependent delivery times. *J Appl Math Comput* 45:259–274

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.