mathematics

MDPI

Article

The Due Date Assignment Scheduling Problem with Delivery Times and Truncated Sum-of-Processing-Times-Based Learning Effect

Jin Qian and Yu Zhan *

College of Science, Northeastern University, Shenyang 110819, China; qianjin@mail.neu.edu.cn
* Correspondence: zhanyu@mail.neu.edu.cn

Abstract: This paper considers a single-machine scheduling problem with past-sequence-dependent delivery times and the truncated sum-of-processing-times-based learning effect. The goal is to minimize the total costs that comprise the number of early jobs, the number of tardy jobs and due date. The due date is a decision variable. There will be corresponding penalties for jobs that are not completed on time. Under the common due date, slack due date and different due date, we prove that these problems are polynomial time solvable. Three polynomial time algorithms are proposed to obtain the optimal sequence.

Keywords: scheduling; delivery times; learning effect; common due date; slack due date; different due date

Citation: Qian, J.; Zhan, Y. The Due Date Assignment Scheduling Problem with Delivery Times and Truncated Sum-of-Processing-Times-Based Learning Effect. *Mathematics* **2021**, *9*, 3085. https://doi.org/10.3390/math9233085

Academic Editor: Javier Alcaraz

Received: 28 October 2021
Accepted: 26 November 2021
Published: 30 November 2021

1. Introduction

Scheduling problems are widely used in manufacturing, logistics, and other practical applications. For a real-word example of our scheduling problems, consider a processing enterprise that has no inventory capacity. As the processing time increases, the processing technology improves. The processing time of the product becomes shorter. The pick-up time of each product is determined by the customer. If the product is produced before the pick-up time or after the pick-up time, an additional delivery fee will be incurred. The delivery price of each early (tardy) job is a fixed charge.

The following three forms of pick-up time are often considered:

(1) All products have a uniform delivery time;
(2) The pick-up time of each product is related to its own processing time and a constant;
(3) Each product has its own independent pick-up time.

The scheduling problem is a very classic discrete combinatorial optimization problem. The methods to solve the scheduling problem mainly include two types: the exact algorithm and approximate algorithm. Exact algorithms mainly include mathematical programming methods, dynamic programming, and branch and bound algorithms. Approximate algorithms mainly include heuristic algorithms and intelligent algorithms. For large-scale non-polynomial time-solvable scheduling problems, intelligent algorithms and machine learning algorithms can be used to solve them. In this paper, a single-machine scheduling problem is considered that contains due dates, the delivery time and learning effect. The actual processing time of a job is a learning function of the previous processing time. The objective function is to minimize the number of early jobs, the number of tardy jobs and due date. Under the common due date, slack due date and different due date, three polynomial time algorithms are proposed to obtain the optimal sequence.

2. Literature Review

In traditional scheduling problems, it is considered that the processing time of jobs is constant. However, in reality, the processing time is often reduced with the increase in

workers' skills and abilities. That means the processing time is no longer a constant. In 2011, Cheng et al. developed the branch-and-bound algorithm and simulated an annealing algorithm in order to study the single-machine scheduling problem with a learning effect and truncation processing time [1]. In 2013, Li et al. analyzed the polynomial time algorithm of the single-machine scheduling problem with a truncation processing time [2]. In 2013, Cheng et al. used a genetic algorithm and branch-and-bound algorithm to solve the two-machine flow-shop scheduling problem with a truncated learning function [3]. In 2016, Wu and Wang studied a single-machine scheduling problem with a learning effect and delivery times [4]. In 2017, Wang et al. solved the single-machine scheduling problem with resource allocation and deterioration effects by using the polynomial time algorithm [5]. In 2018, Wu et al. studied a two-stage scheduling problem with a position-based learning effect [6]. In 2018, Yin studied a single-scheduling problem with resource allocation and a learning effect [7]. In 2020, Zhang studied the scheduling problem with the sum-of-processing-times-based learning effect [8]. In 2020, Qian et al. designed a heuristic algorithm to study the single-scheduling problem with release times and a learning factor [9]. In 2020, Zou et al. studied a multi-machine scheduling problem with the sum-of-processing-times-based learning effect [10]. In 2021, Wu et al. studied a flow-shop scheduling problem with a truncated learning function [11].

In the field of scheduling, the delivery time has attracted extensive attention. The extra time required for a completed job to be delivered to the customer is called the *p ast-sequence-dependent (psd)* delivery time. In 2011, Yang et al. studied a single-machine scheduling problem with delivery times and a learning effect [12]. In 2012, Yang et al. studied a single-machine scheduling problem with delivery times and position-dependent processing times [13]. In 2013, Liu studied a scheduling problem with delivery times and deteriorating jobs [14]. In 2014, Zhao et al. studied a single-machine scheduling problem with delivery times and general position-dependent processing times [15]. In 2021, Qian et al. studied a single-machine scheduling problem with delivery times and deteriorating jobs [16].

In actual production scheduling, the jobs often have due dates. If a job is completed ahead of the due date, it will have an earliness cost; if a job is completed behind the due date, it will have a tardiness cost. In 2013, Yin et al. studied a single-machine scheduling problem with a due date, delivery times and learning effect [17]. In 2014, Lu et al. studied a single-machine scheduling problem with a due date, learning effect and resource allocation [18]. In 2015, Li et al. studied a single-machine scheduling problem with a slack due window, learning effect and resource allocation [19]. In 2016, Sun et al. studied a single-machine scheduling problem with a due date and convex resource allocation [20]. In 2019, Geng et al. studied a flow-shop scheduling problem with a common due date, resource allocation and learning effect [21]. In 2020, Liu et al. studied a single-machine scheduling problem with a due date, learning effect and resource allocation [22]. In 2021, Tian studied a single-machine scheduling problem with resource allocation and generalized earliness–tardiness penalties [23]. In 2021, Wang studied a single-machine scheduling problem with proportional setup times and earliness–tardiness penalties [24]. In 1996, Lann et al. studied a single-machine scheduling problem whose goal was to minimize the number of early and tardy jobs [25]. In 2017, Yuan studied a single-machine scheduling problem to minimize the number of tardy jobs [26]. In 2021, Hermelin studied a single-machine scheduling problem to minimize the weighted number of tardy jobs [27].

The problem is described in the Section 3. The research methods are discussed in the Section 4. Discussion of results is in the Section 5. The conclusion is given in the Section 6.

3. Notation and Problem Statement

Some notations used in this paper are introduced in Table 1.

Table 1. Symbol definition.

Symbol	Meaning
n	the number of jobs
p_j	the normal processing time of J_j
$p_{[j]}$	the normal processing time for the jth position
$p^A_{[j]}$	the actual processing time of $J_{[j]}$
$w_{[j]}$	the waiting time of $J_{[j]}$
$C_{[j]}$	the completion time of $J_{[j]}$
$C_{\max}$	the makespan
$q_{[j]}$	the delivery time of $J_{[j]}$
d	the due date
a	the learning index, $a < 0$
r	the delivery rate, $r > 0$
β	the truncation parameter, $0 < \beta < 1$
α, δ, η	the weights
$[j]$	the job arranged at the jth position
CON	the common due date
SLK	the slack due date
DIF	the different due date

Suppose there were n independent jobs $J = \{J_1, \cdots J_n\}$ continuously processed on a single machine. The machine can handle one job at a time. The actual processing time of J_j at the kth position was:

$$p^A_{j[k]} = p_j \max\{(1 + \sum_{i=1}^{k-1} p_{[i]})^a, \beta\}. \quad (1)$$

The delivery time $q_{[j]}$ of $J_{[j]}$ was:

$$q_{[j]} = rw_{[j]} = r \sum_{i=1}^{j-1} p^A_{[i]}, \quad (2)$$

where $w_{[j]} = \sum_{i=1}^{j-1} p^A_{[i]}$. The completion time of $J_{[j]}$:

$$C_{[j]} = w_{[j]} + p^A_{[j]} + q_{[j]}. \quad (3)$$

The common due date, slack due date and different due date were considered in this paper. For the CON model, the due date of each job was the same. For the SLK model, the due date was the sum of the processing time and certain parameter q. For the DIF model, each job had its own due date. The due date was a decision variable. If J_j was an early job, $U_j = 1$, $V_j = 0$. If J_j was a tardy job, $U_j = 0$, $V_j = 1$. By the three-field notation [28], the models could be defined as:

$$1|p^A_{j[k]} = p_j \max\{(1 + \sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, CON| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta d), \quad (4)$$

$$1|p^A_{j[k]} = p_j \max\{(1 + \sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, SLK| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta q), \quad (5)$$

$$1|p^A_{j[k]} = p_j \max\{(1 + \sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, DIF| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta d_j), \quad (6)$$

where q_{psd} represents the past-sequence-dependent delivery times. The following Figure 1 shows the just-in-time common due date scheduling model.

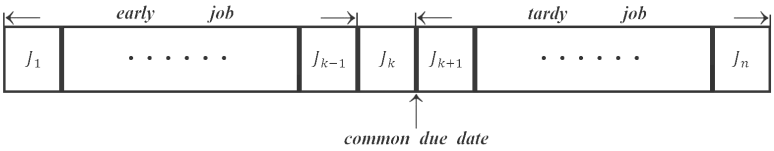

Figure 1. The just-in-time CON scheduling model.

4. Research Method

Lemma 1. *For the* $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}|C_{\max}$ *problem, an optimal schedule could be obtained by the SPT rule* [4].

Lemma 2. *For the* $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}|C_{\max}$ *problem, an optimal schedule could be obtained by the SPT rule* [4].

4.1. The Problem $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, CON|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d)$

Lemma 3. *For any job sequence, the due date d of the optimal scheduling was the completion time of some job.*

Proof. Suppose that the due date d of the optimal scheduling was not equal to the completion time of some job, i.e., $C_{[h]} < d < C_{[h+1]}$, $0 \le h < n$, $C_{[0]} = 0$. The objective function was:

$$Z = h\alpha + (n-h)\delta + n\eta d. \tag{7}$$

When d was equal to $C_{[h]}$, the objective function was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}, \tag{8}$$

$$Z - Z_1 = \alpha + n\eta(d - C_{[h]}) > 0. \tag{9}$$

Therefore, d was the completion time of some job. □

Lemma 4. *When* $\alpha \ge \delta$, *the due date d was equal to 0.*

Proof. When the due date d was equal to $C_{[h]}$, the objective function was:

$$Z = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}. \tag{10}$$

(1) When d was equal to $C_{[h-1]}$, the objective function was:

$$Z_1 = (h-2)\alpha + (n-h+1)\delta + n\eta C_{[h-1]}. \tag{11}$$

(2) When d was equal to $C_{[h+1]}$, the objective function was:

$$Z_2 = h\alpha + (n-h-1)\delta + n\eta C_{[h+1]}.$$

when $\alpha \ge \delta$,

$$Z - Z_1 = \alpha - \delta + n\eta(C_{[h]} - C_{[h-1]}) = \alpha - \delta + n\eta(p_{[h]} + rp_{[h-1]}) > 0, \tag{12}$$

$$Z - Z_2 = -\alpha + \delta + n\eta(C_{[h]} - C_{[h+1]}) = -\alpha + \delta - n\eta(p_{[h+1]} + rp_{[h]}) < 0, \tag{13}$$

$Z_2 > Z > Z_1$. Therefore, the due date d was equal to the start time of the first job. □

For the convenience of proof, we defined two sets: $G_1 = \{J_j | 1 \leq j \leq h\}$, $G_2 = \{J_j | h+1 \leq j \leq n\}$, $d = C_{[h]}$.

Lemma 5. *In the optimal scheduling, the jobs of set G_1 were arranged in an ascending order of normal processing time.*

Proof. There were two adjacent jobs J_u and J_v in the G_1, and J_u was in front of J_v which was at the $(k+1)$th position, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$. Suppose that the starting time of J_1 was 0, $d = C_{[h]}$, $1 \leq k < h \leq n$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_1). \tag{14}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_2). \tag{15}$$

$$Z_1 - Z_2 = n\eta(C_{[h]}(S_1) - C_{[h]}(S_2)). \tag{16}$$

From $p_u \leq p_v$ and Lemma 2, $Z_1 \leq Z_2$, i.e., the jobs of set G_1 were arranged in an ascending order of normal processing time. □

Lemma 6. *In the optimal scheduling, the jobs of set G_2 were arranged in any order of normal processing time.*

Proof. There were two adjacent jobs J_u and J_v in the G_2, and J_u was in front of J_v which was at the $(k+1)$th position, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$, $h < k < n$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_1). \tag{17}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_2). \tag{18}$$

$$Z_1 = Z_2. \tag{19}$$

Therefore, the jobs of set G_2 were arranged in any order of normal processing time. □

Lemma 7. *In the optimal scheduling, the processing time of any job in the G_1 was less than the processing time of any job in the G_2.*

Proof. There were two adjacent jobs J_u and J_v, J_u was at the hth position in the G_1, and J_v was at the $(h+1)$th position in the G_2, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_1). \tag{20}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta C_{[h]}(S_2). \tag{21}$$

$$Z_1 - Z_2 = n\eta(p_u - p_v)\max\{(1 + \sum_{k=1}^{h-1} p_{[k]})^a, \beta\}. \tag{22}$$

If $p_u \leq p_v$, $Z_1 \leq Z_2$, i.e., the processing time of any job in the G_1 was less than the processing time of any job in the G_2. □

The Algorithm 1 was summarized as follows:

Algorithm 1 $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, CON| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta d)$

Require: α, β, δ, η, a, r, p_j, n
Ensure: The optimal sequence, d

1: **First step:** All jobs were sorted by increasing processing time, i.e., $p_1 \leq \cdots \leq p_n$.
2: **Second step:** When h was from 0 to n, the objective function values were calculated, respectively.
3: **Last step:** The optimal position of h was determined by the smallest value of the objective function, and the optimal sequence was arrangedn an ascending order of normal processing time.

Theorem 1. *For the problem* $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, CON| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta d)$*, the complexity of the algorithm was* $O(nlogn)$.

Proof. The first step required $O(nlogn)$ time. The second step required $O(n)$ time. The third step was completed in a constant time. Therefore, the complexity of the algorithm was $O(nlogn)$. □

4.2. The Problem $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, SLK| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta q)$

Lemma 8. *For the optimal scheduling, q was equal to* $(1+r)$ *times the sum of actual processing time for some jobs.*

Proof. Suppose that q was not equal to $(1+r)$ times the sum of the actual processing time for some jobs, i.e., $(1+r)\sum_{j=1}^{h-1} p^A_{[j]} < q < (1+r)\sum_{j=1}^{h} p^A_{[j]}$, $1 \leq h \leq n$, $p_{[0]} = 0$. The objective function was:

$$Z = h\alpha + (n-h)\delta + n\eta q. \tag{23}$$

when $q = (1+r)\sum_{j=1}^{h-1} p^A_{[j]}$, the objective function was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p^A_{[j]}, \tag{24}$$

$$Z - Z_1 = \alpha + n\eta[q - (1+r)\sum_{j=1}^{h-1} p^A_{[j]}] > 0. \tag{25}$$

Therefore, q was equal to $(1+r)$ times the sum of the actual processing time for some jobs. □

Lemma 9. *When* $\alpha \geq \delta$*, q was equal to 0.*

Proof. When q was equal to $(1+r)\sum_{j=1}^{h-1} p^A_{[j]}$ for the optimal scheduling, the objective function was:

$$Z = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p^A_{[j]}. \tag{26}$$

(1) When $q = (1+r)\sum_{j=1}^{h-2} p_{[j]}^{A}$, the objective function was

$$Z_1 = (h-2)\alpha + (n-h+1)\delta + n\eta(1+r)\sum_{j=1}^{h-2} p_{[j]}^{A}. \tag{27}$$

(2) When $q = (1+r)\sum_{j=1}^{h} p_{[j]}^{A}$, the objective function was

$$Z_2 = h\alpha + (n-h-1)\delta + n\eta(1+r)\sum_{j=1}^{h} p_{[j]}^{A}. \tag{28}$$

$$Z - Z_1 = \alpha - \delta + n\eta(1+r)p_{[h-1]}^{A}, \tag{29}$$

$$Z - Z_2 = -\alpha + \delta - n\eta(1+r)p_{[h]}^{A}. \tag{30}$$

when $\alpha \geq \delta$, $Z_2 > Z > Z_1$. Therefore, q was equal to 0. □

For the convenience of proof, we defined two sets: $G_3 = \{J_j | 1 \leq j \leq h-1\}$, $G_4 = \{J_j | h \leq j \leq n\}$, $q = (1+r)\sum_{j=1}^{h-1} p_{[j]}^{A}$.

Lemma 10. *In the optimal scheduling, the jobs of set G_3 were arranged in an ascending order of normal processing time.*

Proof. There were two adjacent jobs J_u and J_v in the G_3, and J_u was in front of J_v which was at the $(k+1)$th position, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$. Suppose that the starting time of J_1 was 0, $q = (1+r)\sum_{j=1}^{h-1} p_{[j]}^{A}$, $1 \leq k \leq h-2$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^{A}(S_1). \tag{31}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^{A}(S_2). \tag{32}$$

$$Z_1 - Z_2 = n\eta(1+r)\left(\sum_{j=1}^{h-1} p_{[j]}^{A}(S_1) - \sum_{j=1}^{h-1} p_{[j]}^{A}(S_2)\right). \tag{33}$$

From $p_u \leq p_v$ and Lemma 1, $Z_1 \leq Z_2$, i.e., the jobs of set G_3 were arranged in an ascending order of normal processing time. □

Lemma 11. *In the optimal scheduling, the jobs of set G_4 were arranged in an ascending order of normal processing time.*

Proof. There were two adjacent jobs J_u and J_v in the G_4, and J_u was in front of J_v which was at the $(k+1)$th position, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$, $h \leq k < n$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^{A}(S_1). \tag{34}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^A(S_2). \tag{35}$$

$$Z_1 = Z_2. \tag{36}$$

Therefore, the jobs of set G_4 were arranged in any order of normal processing time. □

Lemma 12. *In the optimal scheduling, the processing time of any job in the G_3 was less than the processing time of any job in the G_4.*

Proof. There were two adjacent jobs J_u and J_v, and J_u was at the $(h-1)$th position in the G_3, and J_v was at the hth position in the G_4, $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$. The objective function of S_1 was:

$$Z_1 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^A(S_1). \tag{37}$$

when J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. The objective function of S_2 was:

$$Z_2 = (h-1)\alpha + (n-h)\delta + n\eta(1+r)\sum_{j=1}^{h-1} p_{[j]}^A(S_2). \tag{38}$$

$$Z_1 - Z_2 = n\eta(1+r)(p_u - p_v)\max\{(1+\sum_{k=1}^{h-2} p_{[k]})^a, \beta\}. \tag{39}$$

If $p_u \le p_v$, $Z_1 \le Z_2$, i.e., the processing time of any job in the G_3 was less than the processing time of any job in the G_4. □

The Algorithm 2 was summarized as follows:

Algorithm 2 $1|p_{j[k]}^A = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, SLK|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta q)$

Require: α, β, δ, η, a, r, p_j, n
Ensure: The optimal sequence, q

1: **First step:** All jobs were sorted by the increasing processing time, i.e., $p_1 \le \cdots \le p_n$.
2: **Second step:** When h was from 0 to n, the objective function values were calculated, respectively.
3: **Last step:** The optimal position of h was determines by the smallest value of the objective function, and the optimal sequence was arranged in an ascending order of the normal processing time.

Theorem 2. *For the problem $1|p_{j[k]}^A = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, SLK|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta q)$, the complexity of the algorithm was $O(nlogn)$.*

Proof. The first step required $O(nlogn)$ time. The second step required $O(n)$ time. The third step was completed in a constant time. Therefore, the complexity of the algorithm was $O(nlogn)$. □

4.3. The Problem $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, DIF| \sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d_j)$

Lemma 13. *In the optimal scheduling, if* $\eta C_j \geq \delta$*, the due date* d_j *of* J_j *was equal to 0; otherwise,* d_j *was equal to the completion time of* J_j*.*

Proof. The objective function was:

$$Z = \sum_{j=1}^{n} Z_j = \sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d_j), \tag{40}$$

$$Z_j = \alpha U_j + \delta V_j + \eta d_j. \tag{41}$$

(1) When $C_j > d_j$,

$$Z_j = \delta + \eta d_j. \tag{42}$$

(2) When $C_j = d_j$,

$$Z_j = \eta C_j. \tag{43}$$

(3) When $C_j < d_j$,

$$Z_j = \alpha + \eta d_j > \eta C_j. \tag{44}$$

$$Z_j = \min\{\delta, \eta C_j\}. \tag{45}$$

when $\eta C_j \geq \delta$, d_j was equal to 0; otherwise, d_j was equal to C_j. □

Lemma 14. *In the optimal scheduling, the jobs were sequenced in an increasing order of normal processing time.*

Proof. We considered the job sequence $S_1 = \{J_1, \cdots, J_u, J_v, \cdots, J_n\}$. There were two adjacent jobs J_u and J_v. J_u was at the kth position in the S_1 and J_v was at the $(k+1)$th position in the S_1, $1 \leq k < n$. Z_1 was the objective function of S_1. When J_u and J_v were swapped, the sequence of jobs was $S_2 = \{J_1, \cdots, J_v, J_u, \cdots, J_n\}$. Z_2 was the objective function of S_2.

$$Z_1 - Z_2 = \min\{\delta, \eta C_{[k]}(S_1)\} + \min\{\delta, \eta C_{[k+1]}(S_1)\} - \min\{\delta, \eta C_{[k]}(S_2)\} - \min\{\delta, \eta C_{[k+1]}(S_2)\}. \tag{46}$$

$$C_{[k]}(S_1) = (1+r)\sum_{j=1}^{k-1} p^A_{[j]} + p_u \max\{(1+\sum_{i=1}^{k-1} p_{[j]})^a, \beta\}, \tag{47}$$

$$C_{[k]}(S_2) = (1+r)\sum_{j=1}^{k-1} p^A_{[j]} + p_v \max\{(1+\sum_{i=1}^{k-1} p_{[j]})^a, \beta\}, \tag{48}$$

$$C_{[k+1]}(S_1) = (1+r)\sum_{j=1}^{k-1} p^A_{[j]} + (1+r)p_u \max\{(1+\sum_{i=1}^{k-1} p_{[j]})^a, \beta\} + p_v \max\{(1+\sum_{i=1}^{k-1} p_{[j]} + p_u)^a, \beta\}, \tag{49}$$

$$C_{[k+1]}(S_2) = (1+r)\sum_{j=1}^{k-1} p^A_{[j]} + (1+r)p_v \max\{(1+\sum_{i=1}^{k-1} p_{[j]})^a, \beta\} + p_u \max\{(1+\sum_{i=1}^{k-1} p_{[j]} + p_v)^a, \beta\}. \tag{50}$$

From $p_u \leq p_v$ and Lemma 1, $C_{[k]}(S_1) \leq C_{[k]}(S_2)$, $C_{[k+1]}(S_1) \leq C_{[k+1]}(S_2)$, $Z_1 \leq Z_2$. Therefore, the jobs were sequenced in an increasing order of normal processing time in the optimal scheduling. □

The Algorithm 3 was summarized as follows:

Algorithm 3 $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, DIF|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d_j)$

Require: α, β, δ, η, a, r, p_j, n
Ensure: The optimal sequence, d_j
1: **First step:** The optimal sequence was sequenced by an increasing order of normal processing time, i.e., $p_1 \leq \cdots \leq p_n$.
2: **Last step:** The due date d_j was determined by the relationship between ηC_j and δ.

Theorem 3. *For the problem* $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, DIF|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d_j)$*, the complexity of the algorithm was* $O(nlogn)$.

Proof. The first step required $O(nlogn)$ time. The second step required $O(n)$ time. Therefore, the complexity of the algorithm was $O(nlogn)$. □

5. Discussion of Results

5.1. Numerical Discussion

In this section, we used an example to show the calculation process for three different due dates.

Example 1. *There were five jobs processed sequentially on the same machine. The processing time of each job is shown in the Tables 2–20 below:*
$\alpha = 1$, $\delta = 2$, $\eta = 0.2$, $a = -1$, $\beta = 0.5$, $r = 0.1$.

Table 2. Normal processing time.

Job	J_1	J_2	J_3	J_4	J_5
p_j	4	3	5	2	1

Solution 1: $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, CON|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d)$

First step: $p_5 < p_4 < p_2 < p_1 < p_3$. The processing sequence of jobs: $J_5 \rightarrow J_4 \rightarrow J_2 \rightarrow J_1 \rightarrow J_3$.

Second step:

(1) When $h = 0$, $d = 0$, $Z = 10$;
(2) When $h = 1$, $d = 1$, $Z = 9$;
(3) When $h = 2$, $d = 2.1$, $Z = 9.1$;
(4) When $h = 3$, $d = 3.7$, $Z = 9.7$;
(5) When $h = 4$, $d = 5.85$, $Z = 10.85$;
(6) When $h = 5$, $d = 8.55$, $Z = 12.55$.

Third step: The optimal due date was 1.

Table 3. Actual processing time.

$p^A_{[j]}$	$p^A_{[1]}$	$p^A_{[2]}$	$p^A_{[3]}$	$p^A_{[4]}$	$p^A_{[5]}$
Value	1	1	1.5	2	2.5

Table 4. Waiting time.

$w_{[j]}$	$w_{[1]}$	$w_{[2]}$	$w_{[3]}$	$w_{[4]}$	$w_{[5]}$
Value	0	1	2	3.5	5.5

Table 5. Delivery time.

$q_{[j]}$	$q_{[1]}$	$q_{[2]}$	$q_{[3]}$	$q_{[4]}$	$q_{[5]}$
Value	0	0.1	0.2	0.35	0.55

Table 6. Completion time.

$C_{[j]}$	$C_{[1]}$	$C_{[2]}$	$C_{[3]}$	$C_{[4]}$	$C_{[5]}$
Value	1	2.1	3.7	5.85	8.55

Solution 2: $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, SLK| \sum_{j=1}^{n} (\alpha U_j + \delta V_j + \eta q)$

First step: $p_5 < p_4 < p_2 < p_1 < p_3$. The processing sequence of jobs: $J_5 \to J_4 \to J_2 \to J_1 \to J_3$.

Second step:

(1) When $h = 0$, $q = 0$, $Z = 8$;
(2) When $h = 1$, $q = 1.1$, $Z = 8.1$;
(3) When $h = 2$, $q = 2.2$, $Z = 8.2$;
(4) When $h = 3$, $q = 3.85$, $Z = 8.85$;
(5) When $h = 4$, $q = 6.05$, $Z = 10.05$,

Third step: The optimal q was 0.

Table 7. Actual processing time.

$p^A_{[j]}$	$p^A_{[1]}$	$p^A_{[2]}$	$p^A_{[3]}$	$p^A_{[4]}$	$p^A_{[5]}$
Value	1	1	1.5	2	2.5

Table 8. Waiting time.

$w_{[j]}$	$w_{[1]}$	$w_{[2]}$	$w_{[3]}$	$w_{[4]}$	$w_{[5]}$
Value	0	1	2	3.5	5.5

Table 9. Delivery time.

$q_{[j]}$	$q_{[1]}$	$q_{[2]}$	$q_{[3]}$	$q_{[4]}$	$q_{[5]}$
Value	0	0.1	0.2	0.35	0.55

Table 10. Completion time.

$C_{[j]}$	$C_{[1]}$	$C_{[2]}$	$C_{[3]}$	$C_{[4]}$	$C_{[5]}$
Value	1	2.1	3.7	5.85	8.55

Table 11. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	1	1	1.5	2	2.5

Table 12. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	2.1	2.1	2.6	3.1	3.6

Table 13. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	3.2	3.2	3.7	4.2	4.7

Table 14. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	4.85	4.85	5.35	5.85	6.35

Table 15. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	7.05	7.05	7.55	8.05	8.55

Solution 3: $1|p^A_{j[k]} = p_j \max\{(1+\sum_{i=1}^{k-1} p_{[i]})^a, \beta\}, q_{psd}, DIF|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d_j)$

First step: $p_5 < p_4 < p_2 < p_1 < p_3$. The processing sequence of jobs: $J_5 \to J_4 \to J_2 \to J_1 \to J_3$.

Second step: $Z = 2.12$.

Table 16. Due date.

$d_{[j]}$	$d_{[1]}$	$d_{[2]}$	$d_{[3]}$	$d_{[4]}$	$d_{[5]}$
Value	1	2.1	3.7	5.85	8.55

5.2. Extension

In the learning effect scheduling model, the learning index a was less than 0. If $a > 0$, it became the forgetting effect scheduling model.

$$1|p^A_{j[k]} = p_j(1+\sum_{i=1}^{k-1} p_{[i]})^a, q_{psd}, CON(SLK, DIF)|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d), \quad (51)$$

where $a > 0$. The same method could prove the following conclusions. When $0 < a \leq 1$, the optimal sequence was obtained by the longest processing time order. When $a > 1$, the optimal sequence was obtained by the shortest processing time order. Take Example 1 above as an example to show the algorithmic process of the forgetting effect scheduling model (CON).

Solution 4: $1|p^A_{j[k]} = p_j(1+\sum_{i=1}^{k-1} p_{[i]})^a, q_{psd}, CON|\sum_{j=1}^{n}(\alpha U_j + \delta V_j + \eta d)$, where $a = 0.5$.

First step: $p_3 > p_1 > p_2 > p_4 > p_5$. The processing sequence of jobs: $J_3 \to J_1 \to J_2 \to J_4 \to J_5$.

Second step:

(1) When $h = 0$, $d = 0$, $Z = 10$;
(2) When $h = 1$, $d = 5$, $Z = 13$;
(3) When $h = 2$, $d = 15.3$, $Z = 22.3$;
(4) When $h = 3$, $d = 25.76$, $Z = 31.76$;
(5) When $h = 4$, $d = 33.929$, $Z = 38.929$;
(6) When $h = 5$, $d = 38.52$, $Z = 42.52$.

Third step: The optimal due date was 0.

Table 17. Actual processing time.

$p^A_{[j]}$	$p^A_{[1]}$	$p^A_{[2]}$	$p^A_{[3]}$	$p^A_{[4]}$	$p^A_{[5]}$
Value	5	9.8	9.49	7.21	3.87

Table 18. Waiting time.

$w_{[j]}$	$w_{[1]}$	$w_{[2]}$	$w_{[3]}$	$w_{[4]}$	$w_{[5]}$
Value	0	5	14.8	24.29	31.5

Table 19. Delivery time.

$q_{[j]}$	$q_{[1]}$	$q_{[2]}$	$q_{[3]}$	$q_{[4]}$	$q_{[5]}$
Value	0	0.5	1.48	2.429	3.15

Table 20. Completion time.

$C_{[j]}$	$C_{[1]}$	$C_{[2]}$	$C_{[3]}$	$C_{[4]}$	$C_{[5]}$
Value	5	15.3	25.76	33.929	38.52

6. Conclusions

Under the common due date, slack due date and different due date, a single-machine scheduling problem with delivery times and the truncated sum-of-processing-times-based learning effect was studied in this paper. The goal was to minimize the total costs that comprised the number of early jobs, the number of tardy jobs and the due date. Under different due dates, three polynomial time algorithms were proposed to obtain the optimal sequence and due dates, whose complexity was $O(nlogn)$. The optimal sequence was arranged in an ascending order of processing time. We gave three examples to show the calculation process of the algorithms. In the future, the multi-machine environment could be considered to expand the research, i.e., a flow-shop scheduling problem with a delivery time, truncated sum-of-processing-times-based learning effect and due dates could be considered whether there were polynomial time algorithms. The truncated sum-of-processing-times-based forgetting effect was also studied in the single-machine scheduling environment.

Author Contributions: The work presented here was performed in collaboration among all authors. J.Q. designed, analyzed and wrote the paper. Y.Z. analyzed and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Natural Science Foundation of Liaoning Province Project (grant no. 2021-MS-102) and the Fundamental Research Funds for the Central Universities (grant no. N2105021 and N2105020).

Data Availability Statement: Not applicable.

Acknowledgments: We thank the anonymous reviewers for their comments and insights that significantly improved our paper.

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

References

1. Cheng, T.C.E.; Cheng, S.R.; Wu, W.H.; Hsu, P.H.; Wu, C.C. A two-agent single-machine scheduling problem with truncated sum-of-processing-times-based learning considerations. *Comput. Ind. Eng.* **2011**, *60*, 534–541. [CrossRef]
2. Li, L.; Yang, S.W.; Wu, Y.B.; Huo, Y.Z.; Ji, P. Single machine scheduling jobs with a truncated sum-of-processing-times-based learning effect. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 261–267. [CrossRef]

3. Cheng, T.C.E.; Wu, C.C.; Chen, J.C.; Wu, W.H.; Cheng, S.R. Two-machine flowshop scheduling with a truncated learning function to minimize the makespan. *Int. J. Prod. Econ.* **2013**, *141*, 79–86. [CrossRef]
4. Wu, Y.B.; Wang, J.J. Single-machine scheduling with truncated sum-of-processing-times-based learning effect including proportional delivery times. *Neural Comput. Appl.* **2016**, *27*, 937–943. [CrossRef]
5. Wang, J.B.; Liu, M.; Yin, N. Scheduling jobs with controllable processing time, truncated job-dependent learning and deterioration effects. *J. Ind. Manag. Optim.* **2017**, *13*, 1025. [CrossRef]
6. Wu, C.C.; Wang, D.J.; Cheng, S.R.; Chung, I.H.; Lin, W.C. A two-stage three-machine assembly scheduling problem with a position-based learning effect. *Int. J. Prod. Res.* **2018**, *56*, 3064–3079. [CrossRef]
7. Yin, N. Single machine due window assignment resource allocation scheduling with job-dependent learning effect. *J. Appl. Math. Comput.* **2018**, *56*, 715–725. [CrossRef]
8. Zhang, X. Single machine and flowshop scheduling problems with sum-of-processing time based learning phenomenon. *J. Ind. Manag. Optim.* **2020**, *16*, 231. [CrossRef]
9. Qian, J.; Lin, H.; Kong, Y.; Wang, Y. Tri-criteria single machine scheduling model with release times and learning factor. *Appl. Math. Comput.* **2020**, *387*, 124543. [CrossRef]
10. Zou, Y.; Wang, D.; Lin, W.C.; Chen, J.Y.; Yu, P.W.; Wu, W.H.; Chao, Y.P.; Wu, C.C. Two-stage three-machine assembly scheduling problem with sum-of-processing-times-based learning effect. *Soft Comput.* **2020**, *24*, 5445–5462. [CrossRef]
11. Wu, C.C.; Zhang, X.; Azzouz, A.; Shen, W.L.; Cheng, S.R.; Hsu, P.H.; Lin, W.C. Metaheuristics for two-stage flow-shop assembly problem with a truncation learning function. *Eng. Optim.* **2021**, *53*, 843–866. [CrossRef]
12. Yang, S.J.; Hsu, C.J.; Chang, T.R.; Yang, D.L. Single-machine scheduling with past-sequence-dependent delivery times and learning effect. *J. Chin. Inst. Ind. Eng.* **2011**, *28*, 247–255. [CrossRef]
13. Yang, S.J.; Yang, D.L. Single-machine scheduling problems with past-sequence-dependent delivery times and position-dependent processing times. *J. Oper. Res. Soc.* **2012**, *63*, 1508–1515. [CrossRef]
14. Liu, M.; Wang, S.; Chu, C. Scheduling deteriorating jobs with past-sequence-dependent delivery times. *Int. J. Prod. Econ.* **2013**, *144*, 418–421. [CrossRef]
15. Zhao, C.; Tang, H. Single machine scheduling problems with general position-dependent processing times and past-sequence-dependent delivery times. *J. Appl. Math. Comput.* **2014**, *45*, 259–274. [CrossRef]
16. Qian, J.; Han, H. The due date assignment scheduling problem with the deteriorating jobs and delivery time. *J. Appl. Math. Comput.* **2021**, 1–14. in press. [CrossRef]
17. Yin, Y.; Liu, M.; Cheng, T.C.E.; Wu, C.C.; Cheng, S.R. Four single-machine scheduling problems involving due date determination decisions. *Inf. Sci.* **2013**, *251*, 164–181. [CrossRef]
18. Lu, Y.Y.; Li, G.; Wang, Y.B.; Ji, P. Optimal due-date assignment problem with learning effect and resource-dependent processing times. *Optim. Lett.* **2014**, *8*, 113–127. [CrossRef]
19. Li, G.; Luo, M.L.; Zhang, W.J.; Wang, X.Y. Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. *Int. J. Prod. Res.* **2015**, *53*, 1228–1241. [CrossRef]
20. Sun, L.H.; Cui, K.; Chen, J.H.; Wang, J. Due date assignment and convex resource allocation scheduling with variable job processing times. *Int. J. Prod. Res.* **2016**, *54*, 3551–3560. [CrossRef]
21. Geng, X.N.; Wang, J.B.; Bai, D. Common due date assignment scheduling for a no-wait flowshop with convex resource allocation and learning effect. *Eng. Optim.* **2019**, *51*, 1301–1323. [CrossRef]
22. Liu, W.; Jiang, C. Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation. *Eng. Optim.* **2020**, *52*, 74–89. [CrossRef]
23. Tian, Y. Single–Machine Due-Window Assignment Scheduling with Resource Allocation and Generalized Earliness/Tardiness Penalties. *Asia-Pac. J. Oper. Res.* **2021**, in press. [CrossRef]
24. Wang, W. Single-machine due-date assignment scheduling with generalized earliness-tardiness penalties including proportional setup times. *J. Appl. Math. Comput.* **2021**, in press. [CrossRef]
25. Lann, A.; Mosheiov, G. Single machine scheduling to minimize the number of early and tardy jobs. *Comput. Oper. Res.* **1996**, *23*, 769–781. [CrossRef]
26. Yuan, J. Unary NP-hardness of minimizing the number of tardy jobs with deadlines. *J. Sched.* **2017**, *20*, 211–218. [CrossRef]
27. Hermelin, D.; Karhi, S.; Pinedo, M.; Shabtay, D. New algorithms for minimizing the weighted number of tardy jobs on a single machine. *Ann. Oper. Res.* **2021**, *298*, 271–287. [CrossRef]
28. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Kan, A.H.G.R. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* **1979**, *5*, 287–326.