RAIRO-Oper. Res. 55 (2021) S2747–S2758
https://doi.org/10.1051/ro/2020117

RAIRO Operations Research
www.rairo-ro.org

SCHEDULING WITH POSITION-DEPENDENT WEIGHTS, DUE-DATE ASSIGNMENT AND PAST-SEQUENCE-DEPENDENT SETUP TIMES

Li-Yan Wang[1], Xue Huang[1], Wei-Wei Liu[2,*], Yu-Bin Wu[1] and Ji-Bo Wang[1]

Abstract. In this study, we consider single-machine scheduling problems with past-sequence-dependent (denoted by psd) setup times in which the setup times of jobs are proportional to the length of already processed jobs. Under common (CON) and slack (SLK) due-date assignment methods, we prove that the weighted sum of earliness, tardiness and due-date minimization remains polynomially solvable. We also give some extensions for the scheduling problems with psd setup times.

Mathematics Subject Classification. 90B35.

Received March 5, 2020. Accepted October 10, 2020.

1. Introduction

In many realistic scheduling problems, the setup times (costs) of jobs (tasks) are either sequence independent or sequence dependent [1,2]. On the other hand, Koulamas and Kyparisis [16] introduced past-sequence-dependent (psd) setup times for single-machine scheduling, *i.e.*, the setup time of a job is dependent on all already scheduled jobs. Using the three-field notation (see [12]), they proved that the scheduling problem $1|\mathrm{psd}|Z$ remains polynomially solvable, where $Z \in \{C_{\max}, \sum_{i=1}^{n} C_i, \sum_{i=1}^{n}\sum_{j=i}^{n} |C_j - C_i|, \lambda \sum_{i=1}^{n} C_i + (1-\lambda)\sum_{i=1}^{n}\sum_{j=i}^{n} |C_j - C_i|\}$, $0 \le \lambda \le 1$, $C_{\max} = \max\{C_i | j = 1, 2, \ldots, n\}$ is the makespan (C_i is the completion time of job J_i), $\sum_{i=1}^{n} C_i$ is the total completion time, $\mathrm{TADC} = \sum_{i=1}^{n}\sum_{j=i}^{n} |C_j - C_i|$ is the total absolute differences in completion times. Biskup and Herrmann [4] considered single-machine scheduling problems with psd setup times and due dates. They showed that the problem $1|\mathrm{psd}|\sum_{i=1}^{n} L_i$ can be solved by the SPT (Smallest Processing Time first) rule, where $L_i = C_i - d_i$ is the lateness of job J_i, and d_i is the due-date of job J_i. If processing times and due dates are agreeable, they proved that the problem $1|\mathrm{psd}|Z$ ($Z \in \{\sum_{i=1}^{n} T_i, L_{\max}, T_{\max}\}$) can be solved in $O(n \log n)$ time, where $T_i = \max\{0, C_i - d_i\}$ is the tardiness of job J_i, $L_{\max} = \max\{L_i\}$ is maximum lateness, and $T_{\max} = \max\{T_i\}$ is maximum tardiness. If a non-restrictive common due-date d_{opt} is given, Biskup and Herrmann [4] also proved that the problem $1|\mathrm{psd}, d_i = d_{\mathrm{opt}}|Z$ ($Z \in \{\sum_{i=1}^{n}(E_i + T_i), \sum_{i=1}^{n}(\alpha E_i + \beta T_i + \eta d), \sum_{i=1}^{n}(\alpha E_i + \beta T_i + \zeta C_i)\}$) can be solved in polynomial time, where $E_i = \max\{0, d - C_i\}$ is the earliness of job J_i, and $\alpha, \beta, \eta, \zeta$ are given constants. Koulamas and Kyparisis [17] proved that the problem $1|\mathrm{psd}|Z$ ($Z \in \{L_{\max}, T_{\max}, \sum_{i=1}^{n} U_i\}$, where $U_i = 1$ if $C_i > d_i$, otherwise $U_i = 0$) can be solved in $O(n^2)$ time. They also proposed solution algorithms to solve the problem $1|\mathrm{psd}|Z$ ($Z \in \{\sum_{i=1}^{n} w_i T_i, \sum_{i=1}^{n} w_i U_i, \sum_{i=1}^{n} w_i (E_i + T_i)\}$), where w_i is the weight of job J_i.

Keywords. Scheduling, single-machine, past-sequence-dependent setup times, position-dependent weights.

[1] School of Science, Shenyang Aerospace University, Shenyang 110136, P.R. China.
[2] Department of Science, Shenyang Sport University, Shenyang 110102, P.R. China.
*Corresponding author: liuww_2010@163.com

Numerous researchers have considered psd setup times and additional factors such as learning and/or deterioration effects. Cheng *et al.* [7] investigated single-machine scheduling problems with deteriorating jobs and psd setup times. Kuo and Yang [18], Wang [28], Wang *et al.* [33], Wang and Wang [31], Kuo *et al.* [19], Hsu *et al.* [14], Mani *et al.* [25], Wang and Li [29], and Soroush [26] examined single-machine scheduling with psd setup times and job-independent (job-dependent) learning effects. Cheng *et al.* [6], Huang *et al.* [15], and Wang and Wang [32] studied scheduling problems with deteriorating jobs and learning effects.

For scheduling problems and models, due-date assignment methods have drawn increasing attention [8, 10, 11, 21, 38], *i.e.*, jobs are to be completed neither too early nor too late. Brucker [5], Liu *et al.* [22], and Wang *et al.* [35] researched single-machine scheduling with due-date assignment and position-dependent weights, *i.e.*, the weight is not related to the job but to the position in which the job is scheduled. Brucker [5] considered the common (CON) due-date assignment with position-dependent weights. Liu *et al.* [22] dealt with the slack (SLK) due-date assignment with position-dependent weights. Wang *et al.* [35] scrutinized CON and SLK due-date assignment methods with learning effects and resource allocation. Under position-dependent weights, they proved that several scheduling problems can be solved in polynomial time. *"The scheduling problem with* psd *setup times has many real-world applications. For example, consider the scheduling problem of a high-tech manufacturing environment in which a batch of jobs consisting of a group of electronic components needs to be mounted on an IC board" [16].* This paper extends the results of Brucker [5], Liu *et al.* [22], and Wang *et al.* [35], by revisiting psd setup times.

The remaining part of this study is organized as follows. Section 2 formulates the scheduling model. In Sections 3 and 4, we consider CON and SLK due-date assignment problems, respectively. In Section 5, we expound upon the work. Last section presents our conclusions.

2. Formulation

Considering a single-machine, on which there are n jobs $J = \{J_1, J_2, \ldots, J_n\}$ waiting for processing. It is assumed that all the jobs are available at time zero, and preemption (the machine and jobs) is not allowed. Let s_i be the psd setup time of job J_i and p_i be the processing time of job J_i. We assume that the psd setup time of job $J_{\rho(i)}$ is given as follows:

$$s_{\rho(1)} = 0 \quad \text{and} \quad s_{\rho(i)} = \gamma \sum_{h=1}^{i-1} p_{\rho(h)}, \tag{2.1}$$

where $\rho(i)$ is some job scheduled in the ith position in a sequence ρ, $\gamma \geq 0$ is a normalizing constant, and total processing requirement of job $J_{\rho(i)}$ is $\gamma \sum_{h=1}^{i-1} p_{\rho(h)} + p_{\rho(i)}$. For a given sequence ρ, let $C_i = C_{\rho(i)}$ be the completion time of job J_i, by a mathematical induction, we have

$$\begin{aligned} C_{\rho(i)} &= \sum_{j=1}^{i} (s_{\rho(j)} + p_{\rho(j)}) \\ &= \sum_{j=1}^{i} p_{\rho(j)} + \sum_{j=1}^{i} s_{\rho(j)} \\ &= \sum_{j=1}^{i} p_{\rho(j)} + \sum_{j=1}^{i} \left(\gamma \sum_{h=1}^{j-1} p_{\rho(h)} \right) \\ &= \sum_{j=1}^{i} p_{\rho(j)} + \gamma \sum_{j=1}^{i} \sum_{h=1}^{j-1} p_{\rho(h)} \end{aligned}$$

$$= \sum_{j=1}^{i} p_{\rho(j)} + \gamma \sum_{j=1}^{i} (i-j) p_{\rho(j)}$$

$$= \sum_{j=1}^{i} (1 + \gamma(i-j))\, p_{\rho(j)}. \tag{2.2}$$

3. Common due-date assignment

For the common (CON) due-date assignment, we have $d_i = d_{\text{opt}}$, $i = 1, 2, \ldots, n$, where d_{opt} is a decision variable. The problem is to determine d_{opt} and a sequence of jobs such that the following total cost is minimized:

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}} = \sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\text{opt}}| + \omega_0 d_{\text{opt}}, \tag{3.1}$$

where ω_i $(i = 0, 1, 2, \ldots, n)$ is the non-negative weight of ith position in a sequence (*i.e.*, the position-dependent weights), and $L_i = C_i - d_i$ is lateness of job J_i. Using the three-field notation (see [12]), the problem can be denoted as $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$, where 1 denotes a single-machine.

Obviously, for an optimal sequence of the problem $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$, there exists no-idle time between the processing of jobs and the first job starts at time zero (see Brucker [5], Lem. 7.1).

Now, we introduce a dummy job J_0, where its processing time is $p_0 = 0$ and weight is ω_0. Obviously, the job J_0 is always scheduled at time 0, yielding

$$\sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\text{opt}}| + \omega_0 d_{\text{opt}} = \sum_{i=0}^{n} \omega_i |C_{\rho(i)} - d_{\text{opt}}|,$$

and an optimal sequence is given by $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$, where $\rho_{(0)} = 0$.

Lemma 3.1. *For a given sequence* $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$ *of the problem* $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$, $d_{\text{opt}} = C_{\rho(k)} = \sum_{i=1}^{k} (1 + \gamma(k-i))\, p_{\rho(i)}$, *where* k *is a median for the sequence* $\omega_0, \omega_1, \ldots, \omega_n$,

$$\sum_{i=0}^{k-1} \omega_i \le \sum_{i=k}^{n} \omega_i \quad \text{and} \quad \sum_{i=0}^{k} \omega_i \ge \sum_{i=k+1}^{n} \omega_i. \tag{3.2}$$

Proof. Let $C_{\rho(k)} < d_{\text{opt}} < C_{\rho(k+1)}$, we have

$$Z = \sum_{i=1}^{k} \omega_i (d_{\text{opt}} - C_{\rho(i)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(i)} - d_{\text{opt}}) + \omega_0 d_{\text{opt}}.$$

When $d_{\text{opt}} = C_{\rho(k)}$ and $d_{\text{opt}} = C_{\rho(k+1)}$, we have

$$Z_1 = \sum_{i=1}^{k} \omega_i (C_{\rho(k)} - C_{\rho(i)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(i)} - C_{\rho(k)}) + \omega_0 C_{\rho(k)},$$

$$Z_2 = \sum_{i=1}^{k+1} \omega_i (C_{\rho(k+1)} - C_{\rho(i)}) + \sum_{i=k+2}^{n} \omega_i (C_{\rho(i)} - C_{\rho(k+1)}) + \omega_0 C_{\rho(k+1)},$$

$$Z - Z_1 = \sum_{i=1}^{k} \omega_i (d_{\text{opt}} - C_{\rho(i)} - C_{\rho(k)} + C_{\rho(i)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(i)} - d_{\text{opt}} - C_{\rho(i)} + C_{\rho(k)}) + \omega_0 (d_{\text{opt}} - C_{\rho(k)})$$

$$= \sum_{i=1}^{k} \omega_i (d_{\text{opt}} - C_{\rho(k)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(k)} - d_{\text{opt}}) + \omega_0 (d_{\text{opt}} - C_{\rho(k)})$$
$$= \left(\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \right) \left(d_{\text{opt}} - C_{\rho(k)} \right)$$

and

$$Z - Z_2 = \sum_{i=1}^{k} \omega_i (d_{\text{opt}} - C_{\rho(i)} - C_{\rho(k+1)} + C_{\rho(i)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(i)} - d_{\text{opt}} - C_{\rho(i)} + C_{\rho(k+1)}) + \omega_0 (d_{\text{opt}} - C_{\rho(k+1)})$$
$$= \sum_{i=1}^{k} \omega_i (d_{\text{opt}} - C_{\rho(k+1)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(k+1)} - d_{\text{opt}}) + \omega_0 (d_{\text{opt}} - C_{\rho(k+1)})$$
$$= \left(\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \right) \left(d_{\text{opt}} - C_{\rho(k+1)} \right).$$

When $\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \geq 0$ $\left(\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \leq 0 \right)$, we have $Z_1 \leq Z$ $(Z_2 \leq Z)$, then $d_{\text{opt}} = C_{\rho(k)}$ $(d_{\text{opt}} = C_{\rho(k+1)})$, *i.e.*, d_{opt} is equal to the completion time of some job.

From the above analysis, when $d_{\text{opt}} = C_{\rho(k)}$, it follows that $\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \geq 0$. When $d_{\text{opt}} = C_{\rho(k+1)}$, we have $\sum_{i=0}^{k} \omega_i - \sum_{i=k+1}^{n} \omega_i \leq 0$, *i.e.*, if $d_{\text{opt}} = C_{\rho(k)}$, we have $\sum_{i=0}^{k-1} \omega_i - \sum_{i=k}^{n} \omega_i \leq 0$.

In summary, when $d_{\text{opt}} = C_{\rho(k)}$, we have $\sum_{i=0}^{k-1} \omega_i \leq \sum_{i=k}^{n} \omega_i$ and $\sum_{i=0}^{k} \omega_i \geq \sum_{i=k+1}^{n} \omega_i$. □

Remark. The properties of Lemma 3.1 is the same as Brucker [5].

Lemma 3.2. *For a given sequence* $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$ *of the problem* $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$*, the optimal total cost can be written as:*

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}} = \sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\text{opt}}| + \omega_0 d_{\text{opt}} = \sum_{i=1}^{n} \theta_i p_{\rho(i)}, \tag{3.3}$$

where

$$\theta_i = \begin{cases} \sum_{v=0}^{i-1} (1 + \gamma (k - i))\, \omega_v + \sum_{v=i}^{k} \gamma (k - v)\, \omega_v + \sum_{v=k+1}^{n} \gamma (v - k)\, \omega_v, & \text{for} \quad i = 1, 2, \ldots, k; \\ \sum_{v=i}^{n} (1 + (v - k - 1)\, \gamma)\, \omega_v, & \text{for} \quad i = k+1, k+2, \ldots, n. \end{cases} \tag{3.4}$$

Proof. From Lemma 3.1 and equation (2.2), we have $d_{\text{opt}} = C_{\rho(k)}$ and $C_{\rho(i)} = \sum_{j=1}^{i} (1 + \gamma(i - j))\, p_{\rho(j)}$, hence

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$$
$$= \omega_0 C_{\rho(k)} + \sum_{i=1}^{k} \omega_i (C_{\rho(k)} - C_{\rho(i)}) + \sum_{i=k+1}^{n} \omega_i (C_{\rho(i)} - C_{\rho(k)})$$

$$
\begin{aligned}
&= \sum_{i=0}^{k} \omega_i \sum_{v=1}^{i} \gamma(k-i) p_{\rho(v)} + \sum_{i=0}^{k} \omega_i \sum_{v=i+1}^{k} (1+\gamma(k-v)) p_{\rho(v)} \\
&\quad + \sum_{i=k+1}^{n} \omega_i \sum_{v=1}^{k} \gamma\,(i-k)\, p_{\rho(v)} + \sum_{i=k+1}^{n} \omega_i \sum_{v=k+1}^{i} (1+\gamma(i-v)) p_{\rho(v)} \\
&= \sum_{v=1}^{k} p_{\rho(v)} \left(\sum_{i=0}^{v-1} (1+\gamma\,(k-v))\,\omega_i + \sum_{i=v}^{k} \gamma\,(k-i)\,\omega_i + \sum_{i=k+1}^{n} \gamma\,(i-k)\,\omega_i \right) \\
&\quad + \sum_{v=k+1}^{n} p_{\rho(v)} \left(\sum_{i=v}^{n} (1+(i-v)\,\gamma)\,\omega_i \right) \\
&= \sum_{i=1}^{n} \theta_i p_{\rho(i)},
\end{aligned}
$$

where

$$
\theta_i = \begin{cases} \sum\limits_{v=0}^{i-1} (1+\gamma\,(k-i))\,\omega_v + \sum\limits_{v=i}^{k} \gamma\,(k-v)\,\omega_v + \sum\limits_{v=k+1}^{n} \gamma\,(v-k)\,\omega_v, & \text{for} \quad i=1,2,\ldots,k; \\ \sum\limits_{v=i}^{n} (1+(v-k-1)\,\gamma)\,\omega_v, & \text{for} \quad i=k+1,k+2\ldots,n. \end{cases}
$$

□

Lemma 3.3 ([13])**.** *"The sum of products $\sum_{i=1}^{n} x_i y_i$ is minimized if sequence $x_1, x_2, \ldots, x_n$ is ordered nondecreasingly and sequence $y_1, y_2, \ldots, y_n$ is ordered nonincreasingly or vice versa, and it is maximized if the sequences are ordered in the same way."*

The term (3.3) can be minimized by Lemma 3.3, hence the $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$ problem can be solved by the following algorithm:

Algorithm 3.4. Step 1. By Lemma 3.1, calculate k.
Step 2. By using Lemma 3.3 (let $x_i = p_i$, $y_i = \theta_i$, see (3.4)) to determine the optimal job sequence.
Step 3. Set $d_{\text{opt}} = C_{\rho(k)} = \sum_{i=1}^{k} (1+\gamma(k-i))\, p_{\rho(i)}$.

Theorem 3.5. *Algorithm 3.4 solves the problem $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$ in $O(n \log n)$ time.*

Proof. The correctness of Algorithm 3.4 follows from Lemmas 3.1–3.3. Steps 1 and 3 can be performed in linear time $O(n)$, and Step 2 requires $O(n \log n)$ time. Thus, the overall computational complexity of Algorithm 3.4 is $O(n \log n)$. □

The following example is used to illustrate Algorithm 3.4 for the $1\,|\text{psd}, \text{CON}, d_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$ problem.

Example 3.6. Consider $n = 8$, $\gamma = 0.5$: the processing times are $p_1 = 7, p_2 = 5, p_3 = 6, p_4 = 9, p_5 = 10, p_6 = 3, p_7 = 8, p_8 = 11$; the position-dependent weights are $\omega_0 = 4, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5, \omega_4 = 1, \omega_5 = 8, \omega_6 = 7, \omega_7 = 6, \omega_8 = 9$.

By Algorithm 3.4, according to Lemma 3.1, $k = 5$. From Lemma 3.2, we have $\theta_1 = 49, \theta_2 = 48, \theta_3 = 46.5, \theta_4 = 44.5, \theta_5 = 38, \theta_6 = 34, \theta_7 = 27, \theta_8 = 18$. From Lemma 3.3, the optimal sequence is $\rho = [J_6, J_2, J_3, J_1, J_7, J_4, J_5, J_8]$, $d_{\text{opt}} = C_{\rho(5)} = \sum_{j=1}^{5} (1 + 0.5 * (5-j))\, p_{\rho(j)} = 52$, and $\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}} = 2055.5$.

4. Slack due-date assignment

For the slack (SLK) due date assignment, we have $d_i = s_i + p_i + q_{\text{opt}}$, where q_{opt} is a decision variable. The problem is to determine q_{opt} and a sequence of jobs such that the following cost is minimized:

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}} = \sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\rho(i)}| + \omega_0 q_{\text{opt}}. \tag{4.1}$$

Using the three-field notation, the problem can be denoted as $1\,|\text{psd}, \text{SLK}, q_{\text{opt}}|\, \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}}$.

Obviously, for an optimal sequence of the problem $1\,|\text{psd}, \text{SLK}, q_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}}$, there exists no-idle time between the processing of jobs, and the first job starts at time zero (see [22]).

Similar to Section 3, we introduce a dummy job J_0 (its processing time $p_0 = 0$ and weight ω_0) which is always scheduled at time 0, then

$$\sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\rho(i)}| + \omega_0 q_{\text{opt}} = \sum_{i=0}^{n} \omega_i |C_{\rho(i)} - d_{\rho(i)}|,$$

and an optimal sequence is given by $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$, where $\rho_{(0)} = 0$.

Lemma 4.1. *If $C_{\rho(i)} \geq d_{\rho(i)}$ then $C_{\rho(i+1)} \geq d_{\rho(i+1)}$, and if $C_{\rho(i)} \leq d_{\rho(i)}$ then $C_{\rho(i-1)} \leq d_{\rho(i-1)}$.*

Proof. If $C_{\rho(i)} \geq d_{\rho(i)} = s_{\rho(i)} + p_{\rho(i)} + q_{\text{opt}} \geq q_{\text{opt}}$, then $C_{\rho(i+1)} = C_{\rho(i)} + s_{\rho(i)} + p_{\rho(i)} \geq d_{\rho(i)} + s_{\rho(i)} + p_{\rho(i)} \geq q_{\text{opt}} + s_{\rho(i)} + p_{\rho(i)} = d_{\rho(i+1)}$.

If $C_{\rho(i)} \leq d_{\rho(i)}$, then $C_{\rho(i-1)} + s_{\rho(i)} + p_{\rho(i)} \leq s_{\rho(i)} + p_{\rho(i)} + q_{\text{opt}}$, $C_{\rho(i-1)} \leq q_{\text{opt}} \leq s_{\rho(i-1)} + p_{\rho(i-1)} + q_{\text{opt}} = d_{\rho(i-1)}$. □

Lemma 4.2. *For a given sequence $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$ of the problem $1\,|\text{psd}, \text{SLK}, q_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}}$, $q_{\text{opt}} = C_{\rho(l)} = \sum_{i=1}^{l} (1 + \gamma(l-i))\, p_{\rho(i)}$, where l is a median for the sequence $\omega_0, \omega_1, \ldots, \omega_n$,*

$$\sum_{i=0}^{l} \omega_i \leq \sum_{i=l+1}^{n} \omega_i \quad \text{and} \quad \sum_{i=0}^{l+1} \omega_i \geq \sum_{i=l+2}^{n} \omega_i. \tag{4.2}$$

Proof. Define $C_{\rho(l)} < q_{\text{opt}} < C_{\rho(l+1)}$, then $C_{\rho(l)} + s_{\rho(l+1)} + p_{\rho(l+1)} < q_{\text{opt}} + s_{\rho(l+1)} + p_{\rho(l+1)} < C_{\rho(l+1)} + s_{\rho(l+1)} + p_{\rho(l+1)}$, we have $C_{\rho(l+1)} < d_{\rho(l+1)}$. Since $C_{\rho(l)} + s_{\rho(l+2)} + p_{\rho(l+2)} < q_{\text{opt}} + s_{\rho(l+2)} + p_{\rho(l+2)} < C_{\rho(l+1)} + s_{\rho(l+2)} + p_{\rho(l+2)}$, it follows that $d_{\rho(l+2)} < C_{\rho(l+2)}$. From Lemma 4.1, we have

$$\begin{aligned} Z &= \sum_{i=1}^{l+1} \omega_i (d_{\rho(i)} - C_{\rho(i)}) + \sum_{i=l+2}^{n} \omega_i (C_{\rho(i)} - d_{\rho(i)}) + \omega_0 q_{\text{opt}} \\ &= \sum_{i=1}^{l+1} \omega_i (s_{\rho(i)} + p_{\rho(i)} + q_{\text{opt}} - C_{\rho(i)}) + \sum_{i=l+2}^{n} \omega_i (C_{\rho(i)} - s_{\rho(i)} - p_{\rho(i)} - q_{\text{opt}}) + \omega_0 q_{\text{opt}}. \end{aligned}$$

When $q_{\text{opt}} = C_{\rho(l)}$, then $d_{\rho(i)} = s_{\sigma(i)} + p_{\sigma(i)} + C_{\rho(l)}$,

$$Z_1 = \sum_{i=1}^{l+1} \omega_i (s_{\rho(i)} + p_{\rho(i)} + C_{\rho(l)} - C_{\rho(i)}) + \sum_{i=l+2}^{n} \omega_i (C_{\rho(i)} - s_{\rho(i)} - p_{\rho(i)} - C_{\rho(l)}) + \omega_0 C_{\rho(l)}.$$

When $q_{\text{opt}} = C_{\rho(l+1)}$, then $d_{\rho(i)} = s_{\rho(i)} + p_{\rho(i)} + C_{\rho(l+1)}$,

$$Z_2 = \sum_{i=1}^{l+1} \omega_i (s_{\rho(i)} + p_{\rho(i)} + C_{\rho(l+1)} - C_{\rho(i)}) + \sum_{i=l+2}^{n} \omega_i (C_{\rho(i)} - s_{\rho(i)} - p_{\rho(i)} - C_{\rho(l+1)}) + \omega_0 C_{\rho(l+1)}.$$

$$
\begin{aligned}
Z - Z_1 &= \sum_{i=1}^{l+1} \omega_i(q_{\text{opt}} - C_{\rho(l)}) + \sum_{i=l+2}^{n} \omega_i(C_{\rho(l)} - q_{\text{opt}}) + \omega_0(q_{\text{opt}} - C_{\rho(l)}) \\
&= \left(\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i\right) \left(q_{\text{opt}} - C_{\rho(l)}\right), \\
Z - Z_2 &= \sum_{i=1}^{l+1} \omega_i(q_{\text{opt}} - C_{\sigma(l+1)}) + \sum_{i=l+2}^{n} \omega_i(C_{\rho(l+1)} - q_{\text{opt}}) + \omega_0(q_{\text{opt}} - C_{\sigma(l+1)}) \\
&= \left(\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i\right) \left(q_{\text{opt}} - C_{\rho(l+1)}\right).
\end{aligned}
$$

When $\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i \geq 0$ $\left(\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i \leq 0\right)$, $Z_1 \leq Z$ $(Z_2 \leq Z)$, then $q_{\text{opt}} = C_{\rho(l)}$ $(q_{\text{opt}} = C_{\rho(l+1)})$, *i.e.*, q_{opt} is equal to the completion time of some job.

From the above analysis, when $q_{\text{opt}} = C_{\rho(l)}$, it follows that $\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i \geq 0$. When $q_{\text{opt}} = C_{\rho(l+1)}$, it follows that $\sum_{i=0}^{l+1} \omega_i - \sum_{i=l+2}^{n} \omega_i \leq 0$, so, when $q_{\text{opt}} = C_{\rho(l)}$, it follows that $\sum_{i=0}^{l} \omega_i - \sum_{i=l+1}^{n} \omega_i \leq 0$.

In summary, when $q_{\text{opt}} = C_{\rho(l)}$, we have $\sum_{i=0}^{l} \omega_i \leq \sum_{i=l+1}^{n} \omega_i$ and $\sum_{i=0}^{l+1} \omega_i \geq \sum_{i=l+2}^{n} \omega_i$. □

Remark. The properties of Lemmas 4.1 and 4.2 is the same as Liu *et al.* [22].

Lemma 4.3. *For the problem* $1\,|\text{psd}, \text{SLK}, q_{\text{opt}}|\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}}$, *the optimal total cost can be written as:*

$$
\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}} = \sum_{i=1}^{n} \omega_i |C_{\rho(i)} - d_{\rho(i)}| + \omega_0 q_{\text{opt}} = \sum_{i=1}^{n} \theta_i p_{\rho(i)}, \tag{4.3}
$$

where

$$
\theta_i = \begin{cases}
\sum_{v=0}^{i} \left(1 + \gamma\left(l - i\right)\right) \omega_v + \sum_{v=i+1}^{l+1} \gamma\left(l - v + 1\right) \omega_v & \\
\quad + \sum_{v=l+2}^{n} \gamma\left(v - l - 1\right) \omega_v, & \text{for} \quad i = 1, 2, \ldots, l; \\
\sum_{v=i+1}^{n} \left(1 + \left(v - l - 2\right) \gamma\right) \omega_v, & \text{for} \quad i = l+1, l+2, \ldots, n-1; \\
0, & \text{for} \quad i = n.
\end{cases} \tag{4.4}
$$

Proof. Let $\rho = [\rho_{(0)}, \rho_{(1)}, \ldots, \rho_{(n)}]$ and $q_{\text{opt}} = C_{\sigma(l)}$ be an optimal solution such that equation (4.2) can be satisfied, we have

$$
\begin{aligned}
&\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 q_{\text{opt}} \\
&\quad = \omega_0 C_{\rho(l)} + \sum_{i=1}^{l+1} \omega_i (C_{\rho(l)} - C_{\rho(i)}) + \sum_{i=l+2}^{n} \omega_i (C_{\rho(i)} - C_{\rho(l)}) \\
&\quad = \sum_{i=0}^{l+1} \omega_i \sum_{v=1}^{i-1} \gamma(l - i + 1) p_{\rho(v)} + \sum_{i=0}^{l+1} \omega_i \sum_{v=i}^{l} (1 + \gamma(l - v)) p_{\rho(v)}
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{i=l+2}^{n}\omega_i\sum_{v=1}^{l}\gamma(i-l-1)p_{\rho(v)}+\sum_{i=l+2}^{n}\omega_i\sum_{v=l+1}^{i-1}(1+\gamma(i-v-1))p_{\rho(v)}\\
&=\sum_{v=1}^{l}p_{\rho(v)}\left(\sum_{i=0}^{v}(1+\gamma\,(l-v))\,\omega_i+\sum_{i=v+1}^{l+1}\gamma\,(l-i+1)\,\omega_i+\sum_{i=l+2}^{n}\gamma\,(i-l-1)\,\omega_i\right)\\
&+\sum_{v=l+1}^{n-1}p_{\rho(v)}\left(\sum_{i=v+1}^{n}(1+(i-v-1)\,\gamma)\,\omega_i\right)\\
&=\sum_{i=1}^{n}\theta_i p_{\rho(i)},
\end{aligned}
$$

where

$$
\theta_i=\begin{cases}
\sum\limits_{v=0}^{i}(1+\gamma\,(l-i))\,\omega_v+\sum\limits_{v=i+1}^{l+1}\gamma\,(l-v+1)\,\omega_v & \\
\quad+\sum\limits_{v=l+2}^{n}\gamma\,(v-l-1)\,\omega_v, & \text{for} \quad i=1,2,\ldots,l;\\
\sum\limits_{v=i+1}^{n}(1+(v-l-2)\,\gamma)\,\omega_v, & \text{for} \quad i=l+1,l+2,\ldots,n-1;\\
0, & \text{for} \quad i=n.
\end{cases}
$$

□

The term (4.3) can be minimized by Lemma 3.3; hence the $1\,|\mathrm{psd},\mathrm{SLK},q_{\mathrm{opt}}|\sum_{i=1}^{n}\omega_i|L_{\rho(i)}|+\omega_0 q_{\mathrm{opt}}$ problem can be solved by the following algorithm:

Algorithm 4.4. Step 1. By Lemma 4.2, calculate l.
Step 2. By using Lemma 3.3 (let $x_i=p_i$, $y_i=\theta_i$ (see (4.4))) to determine the optimal job sequence.
Step 3. Set $q_{\mathrm{opt}}=C_{\rho(l)}=\sum_{i=1}^{l}(1+\gamma(l-i))\,p_{\rho(i)}$.

Theorem 4.5. *Algorithm 4.4 solves the problem* $1\,|\mathrm{psd},\mathrm{SLK},q_{\mathrm{opt}}|\sum_{i=1}^{n}\omega_i|L_{\rho(i)}|+\omega_0 q_{\mathrm{opt}}$ *in* $O(n\log n)$ *time.*

Proof. Similar to the proof of Theorem 3.5. □

The following example is used to illustrate Algorithm 4.4 for the problem $1\,|\mathrm{psd},\mathrm{SLK},q_{\mathrm{opt}}|\sum_{i=1}^{n}\omega_i|L_{\rho(i)}|+\omega_0 q_{\mathrm{opt}}$.

Example 4.6. The input data in this example is the same as in Example 3.6.

By Algorithm 4.4, according to Lemma 4.2, $l=4$. From Lemma 4.3, we have $\theta_1=48,\theta_2=46.5$, $\theta_3=44.5,\theta_4=38,\theta_5=34,\theta_6=27,\theta_7=18,\theta_8=0$. From Lemma 3.3, the optimal sequence is $\rho=[J_6,J_2,J_3,J_1,J_7,J_4,J_5,J_8]$, $q_{\mathrm{opt}}=C_{\rho(4)}=\sum_{j=1}^{4}(1+0.5*(4-j))\,p_{\rho(j)}=33.5$, and $\sum_{i=1}^{n}\omega_i|L_{\rho(i)}|+\omega_0 q_{\mathrm{opt}}=1604.5$.

5. Extensions

5.1. Truncated job-dependent learning effect

In this subsection, the proposed model is extended by the introduction of truncated job-dependent learning effect [3,23,24,27,34,36,37], *i.e.*, if job J_i is scheduled in the rth position in a sequence, its actual processing time is given by

$$p_i^A=p_i\max\{r^{a_i},b\},i,r=1,\ldots,n, \tag{5.1}$$

where $a_i \leq 0$ is the job-dependent learning effect, and b is a truncation parameter ($0 < b < 1$). The psd setup time of job $J_{\rho(i)}$ is $s_{\rho(i)} = \gamma \sum_{h=1}^{i-1} p_{\rho(h)}^A$. For the SLK due-date assignment, $d_i = s_i + p_i^A + q_{\text{opt}}$.

Obviously, Lemmas 3.1, 3.2, 4.1 and 4.2 still hold when truncated job-dependent learning effect is introduced. Similar to the above analysis, we have

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}} = \sum_{i=1}^{n} \theta_i p_{\rho(i)}^A = \sum_{i=1}^{n} \theta_i p_{\rho(i)} \max\{r^{a_{\rho(i)}}, b\}, \tag{5.2}$$

where, for the CON due-date assignment, θ_i $(i = 1, 2, \ldots, n)$ is given by (3.4); for the SLK due-date assignment, θ_i $(i = 1, 2, \ldots, n)$ is given by (4.4).

Let

$$x_{ir} = \begin{cases} 1, & \text{if job } J_i \text{ is assigned to the } r\text{th position,} \\ 0, & \text{otherwise.} \end{cases}$$

Then, we can formulate the sequence of the problem $1\left|\text{psd}, \text{CON/SLK}, d_{\text{opt}}/q_{\text{opt}}\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}}$ as the following assignment problem:

$$\text{Min} \sum_{i=1}^{n} \sum_{r=1}^{n} \theta_r p_i \max\{r^{a_i}, b\} x_{ir} \tag{5.3}$$

$$\text{s.t.} \sum_{i=1}^{n} x_{ir} = 1, r = 1, \ldots, n \tag{5.4}$$

$$\sum_{r=1}^{n} x_{ir} = 1, i = 1, \ldots, n \tag{5.5}$$

$$x_{ir} = \{0, 1\}. \tag{5.6}$$

Based on the above analysis, the problem $1\left|\text{psd}, \text{CON/SLK}, d_{\text{opt}}/q_{\text{opt}}, p_i^A = p_i \max\{r^{a_i}, b\}\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}}$ can be solved by the following algorithm:

Algorithm 5.1. Step 1. For the CON due-date assignment, by using Lemma 3.1, calculate k; For the SLK due-date assignment, by using Lemma 4.2, calculate l.

Step 2. Solve the assignment problem (5.3) to (5.6) to determine the optimal sequence.

Step 3. Calculate $d_{\text{opt}} = C_{\rho(k)} = \sum_{i=1}^{k} (1 + \gamma(k - i)) p_{\rho(i)}^A$, $q_{\text{opt}} = C_{\rho(l)} = \sum_{i=1}^{l} (1 + \gamma(l - i)) p_{\rho(i)}^A$.

Based on the above analysis, we have

Theorem 5.2. *The problem* $1\left|\text{psd}, \text{CON/SLK}, d_{\text{opt}}/q_{\text{opt}}, p_i^A = p_i \max\{r^{a_i}, b\}\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}}$ *can be solved by Algorithm 5.1 in* $O(n^3)$ *time.*

The following example is only used to illustrate Algorithm 5.1 for the problem $1\left|\text{psd}, \text{CON}, d_{\text{opt}}, p_i^A = p_i \max\{r^{a_i}, b\}\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$.

Example 5.3. Consider $n = 8$, $\gamma = 0.5$, $b = 0.6$: the processing times are $p_1 = 7, p_2 = 5, p_3 = 6, p_4 = 9, p_5 = 10, p_6 = 3, p_7 = 8, p_8 = 11$; the position-dependent weights are $\omega_0 = 4, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5, \omega_4 = 1, \omega_5 = 8, \omega_6 = 7, \omega_7 = 6, \omega_8 = 9$ and job-dependent learning effects are $a_1 = -0.27, a_2 = -0.25, a_3 = -0.3, a_4 = -0.29, a_5 = -0.32, a_6 = -0.33, a_7 = -0.28, a_8 = -0.31$.

By Algorithm 5.1 and Example 3.6, we have $k = 5$, and $\theta_1 = 49, \theta_2 = 48, \theta_3 = 46.5, \theta_4 = 44.5, \theta_5 = 38, \theta_6 = 34, \theta_7 = 27, \theta_8 = 18$. The values $\theta_r p_i \max\{r^{a_i}, b\}$ are given in Table 1. The costs of solution for the assignment problem (5.3–5.6) are given in bold in Table 1 and the optimal sequence is $\rho = [J_6, J_2, J_3, J_1, J_5, J_7, J_4, J_8]$, $d_{\text{opt}} = C_{\rho(5)} = \sum_{j=1}^{5} (1 + 0.5 * (5 - j)) p_{\rho(j)} = 41.08366$, and $\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}} = 1421.016$.

TABLE 1. Values $\theta_r p_i \max\{r^{a_i}, b\}$.

$J_i \setminus r$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$
J_1	343.0000	278.6514	241.9515	**214.2406**	172.2502	146.7155	113.4000	75.60000
J_2	245.0000	**201.8151**	176.6618	157.3313	127.0607	108.6203	82.99640	54.00000
J_3	294.0000	233.9287	**200.6632**	176.1543	140.6837	122.4000	97.20000	64.80000
J_4	441.0000	353.3337	304.3199	267.9200	214.4494	183.6000	**145.8000**	97.20000
J_5	490.0000	384.5135	327.1705	285.5623	**228.0000**	204.0000	162.0000	108.0000
J_6	**147.0000**	114.5573	97.07875	84.48925	68.40000	61.20000	48.60000	32.40000
J_7	392.0000	316.2590	273.4947	241.4756	193.7144	**164.6973**	129.6000	86.40000
J_8	539.0000	425.9068	363.8631	318.5034	253.8023	224.4000	178.2000	**118.8000**

For a special case: $a_i = a$, we have:

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}} = \sum_{i=1}^{n} \theta_i p_{\rho(i)}^A = \sum_{i=1}^{n} \theta_i p_{\rho(i)} \max\{r^a, b\}. \tag{5.7}$$

Obviously, the minimization of term (5.7) can be obtained by Lemma 3.3 (*i.e.*, $x_i = p_i$, $y_i = \theta_i \max\{i^a, b\}$), hence, we have the following result:

Theorem 5.4. *The problem* $1\left|\text{psd}, \text{CON/SLK}, d_{\text{opt}}/q_{\text{opt}}, p_i^A = p_i \max\{r^a, b\}\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}}$ *can be solved in* $O(n\text{log}n)$ *time.*

5.2. Deteriorating jobs

In this subsection, we introduce deteriorating jobs [9,30] to the scheduling, *i.e.*, the actual processing time of job J_i is given by

$$p_i^A = p_i + ct, i = 1, \ldots, n, \tag{5.8}$$

where $c \geq 0$ is the deterioration rate, and t is its starting time.

Clearly, Lemmas 3.1, 3.2, 4.1 and 4.2 still hold when deteriorating jobs are introduced. Similar to the above analysis, we have

$$\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}} = \sum_{i=1}^{n} \theta_i p_{\rho(i)}^A = \sum_{i=1}^{n} \Delta_i p_{\sigma(i)}, \tag{5.9}$$

where

$$\begin{aligned}
&\Delta_1 = \theta_1 + c\theta_2 + c(1+c)\theta_3 + \ldots + c(1+c)^{n-2}\theta_n \\
&\Delta_2 = \theta_2 + c\theta_3 + c(1+c)\theta_4 + \ldots + c(1+c)^{n-3}\theta_n \\
&\Delta_3 = \theta_3 + c\theta_4 + c(1+c)\theta_5 + \ldots + c(1+c)^{n-4}\theta_n \\
&\ldots \\
&\Delta_{n-1} = \theta_{n-1} + c\theta_n \\
&\Delta_n = \theta_n,
\end{aligned} \tag{5.10}$$

where, for the CON due-date assignment, θ_i $(i = 1, 2, \ldots, n)$ is given by (3.4); for the SLK due-date assignment, θ_i $(i = 1, 2, \ldots, n)$ is given by (4.4).

Obviously, the minimization of term (5.9) can be obtained by Lemma 3.3 (*i.e.*, $x_i = p_i$, $y_i = \Delta_i$), thus yielding the following result.

TABLE 2. Scheduling problems with the psd setup times.

Problem	Complexity	Ref.
$1\left\|\text{psd}, \text{CON}, d_{\text{opt}}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 d_{\text{opt}}$	$O(n \log n)$	Theorem 3.5
$1\left\|\text{psd}, \text{SLK}, q_{\text{opt}}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 q_{\text{opt}}$	$O(n \log n)$	Theorem 4.5
$1\left\|\text{psd}, \text{CON}, q_{\text{opt}}, p_i^A = p_i \max\{r^{a_i}, b\}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 d_{\text{opt}}$	$O(n^3)$	Theorem 5.2
$1\left\|\text{psd}, \text{SLK}, q_{\text{opt}}, p_i^A = p_i \max\{r^{a_i}, b\}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 q_{\text{opt}}$	$O(n^3)$	Theorem 5.2
$1\left\|\text{psd}, \text{CON}, q_{\text{opt}}, p_i^A = p_i \max\{r^{a}, b\}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 d_{\text{opt}}$	$O(n \log n)$	Theorem 5.4
$1\left\|\text{psd}, \text{SLK}, q_{\text{opt}}, p_i^A = p_i \max\{r^{a}, b\}\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 q_{\text{opt}}$	$O(n \log n)$	Theorem 5.4
$1\left\|\text{psd}, \text{CON}, d_{\text{opt}}, p_i + ct\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 d_{\text{opt}}$	$O(n \log n)$	Theorem 5.5
$1\left\|\text{psd}, \text{SLK}, q_{\text{opt}}, p_i + ct\right\| \sum_{i=1}^{n} \omega_i \|L_{\rho(i)}\| + \omega_0 q_{\text{opt}}$	$O(n \log n)$	Theorem 5.5

Theorem 5.5. *The problem* $1\left|\text{psd}, \text{CON/SLK}, d_{\text{opt}}/q_{\text{opt}}, p_i^A = p_i + ct\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}/q_{\text{opt}}$ *can be solved in* $O(n\text{log}n)$ *time.*

The following example is only used to illustrate Algorithm 3.4 for the problem $1\left|\text{psd}, \text{CON}, d_{\text{opt}}, p_i^A = p_i + ct\right| \sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}}$.

Example 5.6. The input data in this example is the same as in Example 3.6 except that $c = 0.1$.

By Algorithm 3.4 and Example 3.6, we have $k = 5$, and $\theta_1 = 49, \theta_2 = 48, \theta_3 = 46.5, \theta_4 = 44.5, \theta_5 = 38, \theta_6 = 34, \theta_7 = 27, \theta_8 = 18$. From (5.10), we have $\Delta_1 = 81.87243, \Delta_2 = 73.52039, \Delta_3 = 65.47308, \Delta_4 = 57.70280, \Delta_5 = 46.54800, \Delta_6 = 38.68000, \Delta_7 = 28.80000, \Delta_8 = 18$. From Lemma 3.3, the optimal sequence is $\rho = [J_6, J_2, J_3, J_1, J_7, J_4, J_5, J_8]$, $d_{\text{opt}} = C_{\rho(5)} = \sum_{j=1}^{5} (1 + 0.5 * (5 - j))\, p_{\rho(j)}^A = 59.04380$, and $\sum_{i=1}^{n} \omega_i |L_{\rho(i)}| + \omega_0 d_{\text{opt}} = 2616.481$.

6. Conclusions

In this paper, we considered single-machine scheduling problems with psd setup times and position-dependent weights. Under the CON and SLK due-date assignment methods, we proved that a non-regular objective function minimization can be solved in $O(n \log n)$ time (see Tab. 2). Further research may study other non-regular objective functions (such as the due-window assignment, Liman *et al.* [20], Zhang and Wang [39], and Zhang *et al.* [40]). In addition, multi-machine problems with the psd setup times are also interesting issues.

Acknowledgements. This work was supported by the Natural Science Foundation of Liaoning Province, China (2020-MS-233).

References

[1] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* **246** (2015) 345–378.

[2] A. Allahverdi, C.T. Ng, T.C.E. Cheng and M.Y. Kovalyov, A survey of scheduling problems with setup times or costs. *Eur. J. Oper. Res.* **187** (2008) 985–1032.

[3] D. Biskup, A state-of-the-art review on scheduling with learning effects. *Eur. J. Oper. Res.* **188** (2008) 315–329.

[4] D. Biskup and J. Herrmann, Single-machine scheduling against due dates with past-sequence-dependent setup times. *Eur. J. Oper. Res.* **191** (2008) 587–592.

[5] P. Brucker, Scheduling Algorithms, 3rd edition. Springer, Berlin-Heidelberg (2001).

[6] T.C.E. Cheng, W.-C. Lee and C.-C. Wu, Scheduling problems with deteriorating jobs and learning effects including proportional setup times. *Comput. Ind. Eng.* **58** (2010) 326–331.

[7] T.C.E. Cheng, W.-C. Lee and C.-C. Wu, Single-machine scheduling with deteriorating jobs and past-sequence-dependent setup times. *Appl. Math. Model.* **35** (2011) 1861–1867.

[8] P. Chrétienne, Minimizing the earliness and tardiness cost of a sequence of tasks on a single machine. *RAIRO:OR* **35** (2001) 165–187.
[9] S. Gawiejnowicz, Time-Dependent Scheduling. Springer, Berlin-Heidelberg (2008).
[10] V.S. Gordon, J.M. Proth and C.B. Chu, A survey of the state of-the-art of common due date assignment and scheduling research. *Eur. J. Oper. Res.* **139** (2002) 1–25.
[11] V.S. Gordon, J.M. Proth and C.B. Chu, Due date assignment and scheduling: SLK, TWK and other due date assignment models. *Prod. Plan. Control* **13** (2002) 117–132.
[12] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Disc. Math.* **5** (1979) 287–326.
[13] G.H. Hardy, J.E. Littlewood and G. Pólya, Inequalities, 2nd edition. Cambridge University Press, Cambridge, UK (1967).
[14] C.-J. Hsu, W.-H. Kuo and D.-L. Yang, Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects. *Appl. Math. Model.* **35** (2011) 1492–1496.
[15] X. Huang, G. Li, Y. Huo and P. Ji, Single machine scheduling with general time-dependent deterioration, position-dependent learning and past sequence-dependent setup times. *Optim. Lett.* **7** (2013) 1793–1804.
[16] C. Koulamas and G.J. Kyparisis, Single-machine scheduling problems with past-sequence-dependent setup times. *Eur. J. Oper. Res.* **187** (2008) 1045–1049.
[17] C. Koulamas and G.J. Kyparisis, New results for single-machine scheduling with past-sequence-dependent setup times and due date-related objectives. *Eur. J. Oper. Res.* **278** (2019) 149–159.
[18] W.-H. Kuo and D.-L. Yang, Single-machine scheduling with past-sequence-dependent setup times and learning effects. *Inf. Process. Lett.* **102** (2007) 22–26.
[19] W.-H. Kuo, C.-J. Hsu and D.-L. Yang, Some unrelated parallel machine scheduling problems with past-sequence-dependent setup time and learning effects. *Comput. Ind. Eng.* **61** (2011) 179–183.
[20] S.D. Liman, S.S. Panwalkar and S. Thongmee, Common due window size and location determination in a single machine scheduling problem. *J. Oper. Res. Soc.* **49** (1998) 1007–1010.
[21] H. Lin, C. He and Y. Lin, Bicriteria scheduling for due date assignment with total weighted tardiness. *RAIRO:OR* **52** (2018) 359–370.
[22] W. Liu, X. Hu and X.-Y. Wang, Single machine scheduling with slack due dates assignment. *Eng. Optim.* **49** (2017) 709–717.
[23] Y.-Y. Lu, G. Li, Y.-B. Wu and P. Ji, Optimal due-date assignment problem with learning effect and resource-dependent processing times. *Optim. Lett.* **8** (2014) 113–127.
[24] Y.-Y. Lu, F. Teng and Z.-X. Feng, Scheduling jobs with truncated exponential sum-of-logarithm-processing-times based and position-based learning effects. *Asia Pac. J. Oper. Res.* **32** (2015) 1550026.
[25] V. Mani, P.-C. Chang and S.-H. Chen, Single-machine scheduling with past-sequence-dependent setup times and learning effects: a parametric analysis. *Int. J. Syst. Sci.* **42** (2011) 2097–2102.
[26] H.M. Soroush, Scheduling in bicriteria single machine systems with past-sequence-dependent setup times and learning effects. *J. Oper. Res. Soc.* **65** (2014) 1017–1036.
[27] L. Tai, Optimizing batch-processing operations with batch-position-based learning effects. To appear in: *RAIRO:OR*. DOI: 10.1051/ro/2019108 (2021).
[28] J.-B. Wang, Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect. *Comput. Ind. Eng.* **55** (2008) 584–591.
[29] J.-B. Wang and J.-X. Li, Single machine past-sequence-dependent setup times scheduling with general position-dependent and time-dependent learning effects. *Appl. Math. Model.* **35** (2011) 1388–1395.
[30] J.-J. Wang and Y.-J. Liu, Single-machine bicriterion group scheduling with deteriorating setup times and job processing times. *Appl. Math. Comput.* **242** (2014) 309–314.
[31] X.-R. Wang and J.-B. Wang, Scheduling with past-sequence-dependent setup times and learning effects on a single machine. *Int. J. Adv. Manuf. Technol.* **48** (2010) 739–746.
[32] X.-Y. Wang and J.-J. Wang, Scheduling problems with past-sequence-dependent setup times and general effects of deterioration and learning. *Appl. Math. Model.* **37** (2013) 4905–4914.
[33] J.-B. Wang, D. Wang, L.-Y. Wang, L. Lin, N. Yin and W.-W. Wang, Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times. *Comput. Math. App.* **57** (2009) 9–16.
[34] X.-R. Wang, J. Jin, J.-B. Wang and P. Ji, Single machine scheduling with truncated job-dependent learning effect. *Optim. Lett.* **8** (2014) 669–677.
[35] J.-B. Wang, X.-N. Geng, L. Liu, J.-J. Wang and Y.-Y. Lu, Single machine CON/SLK due date assignment scheduling with controllable processing time and job-dependent learning effects. *Comput. J.* **61** (2018) 1329–1337.
[36] J.-B. Wang, F. Liu and J.-J. Wang, Research on m-machine flow shop scheduling with truncated learning effects. *Int. Trans. Oper. Res.* **26** (2019) 1135–1151.
[37] J.-B. Wang, M. Gao, J.-J. Wang, L. Liu and H. He, Scheduling with a position-weighted learning effect and job release dates. *Eng. Optim.* **52** (2020) 1475–1493.
[38] X. Xiong, D. Wang, T.C.E. Cheng, C. Wu and Y. Yin, Single-machine scheduling and common due date assignment with potential machine disruption. *Int. J. Prod. Res.* **56** (2018) 1345–1360.
[39] X. Zhang and Y. Wang, Single-machine scheduling CON/SLK due window assignment problems with sum-of-processed time based learning effect. *Appl. Math. Comput.* **250** (2015) 628–635.
[40] X. Zhang, W. Lin, W. Wu and C. Wu, Single-machine common/slack due window assignment problems with linear decreasing processing times. *Eng. Optim.* **49** (2017) 1388–1400.