RAIRO-Oper. Res. 55 (2021) 561–569
https://doi.org/10.1051/ro/2021030

RAIRO Operations Research
www.rairo-ro.org

A NOTE ON PARALLEL-MACHINE SCHEDULING WITH CONTROLLABLE PROCESSING TIMES AND JOB-DEPENDENT LEARNING EFFECTS

Shan-Shan Lin*

Abstract. This note studies a unrelated parallel-machine scheduling problem with controllable processing times and job-dependent learning effects, where the objective function is to minimize the weighted sum of total completion time, total load, and total compression cost. We show that the problem can be solved in $O(n^{m+2})$ time, where m and n are the numbers of machines and jobs. We also show how to apply the technique to several single-machine scheduling problems with total criteria.

Mathematics Subject Classification. 90B35, 68M20.

Received July 31, 2020. Accepted February 23, 2021.

1. Introduction

Unlike classical scheduling, in scheduling problems with controllable processing times (resource allocation), the processing time of a job may be controlled by allocating extra resources (*e.g.*, fuel, gas, or catalyzers, see [25, 26, 40]). Scheduling problems with controllable processing times have attracted many scheduling researchers in recent years; see *e.g.*, [13, 21, 29, 38].

On the other hand, the idea of scheduling with a learning effect, understood as dependency of a job processing time on the job position number in schedule was introduced by Gawiejnowicz [6]. Examples of learning effects often appear in logistics, manufacturing and services settings [22]. For more studies on scheduling with learning effects, we refer the reader to Azzouz *et al.* [3] and Tai [31]. A more recent review of scheduling with learning effect can be found in books by Agnetis *et al.* [2], Strusevich and Rustogi [30], and Gawiejnowicz [7].

Recently, there has been increasing attention to scheduling problems involving both controllable processing times and learning effects (see [15, 19, 20, 34, 35]). Yin and Wang [39] considered single machine scheduling problem with controllable processing times and learning effects, *i.e.*, the actual processing time P_j of job J_j in position r is $P_j(x_j, r) = (p_j - x_j)r^a$, where p_j is normal processing time of J_j, x_j is compression of the processing time of job J_j, $0 \le x_j \le m_j$, m_j is maximum reduction in processing time of job J_j, and $a \le 0$ is a learning effect. For a cost function containing total completion time (waiting), total absolute differences in completion (waiting) times and total compression cost, they proved that the problem can be solved in polynomial time. Li *et al.* [16] addressed single machine scheduling problems with controllable processing times and job-dependent learning effects, *i.e.*, the actual processing time of job J_j in position r is $P_j(x_j, r) = (p_j - x_j)r^{a_j}$, where $a_j \le 0$ is the job-dependent learning index of job J_j [24]. In this note, we consider a model of parallel-machine scheduling,

Keywords. Scheduling, unrelated parallel-machine, controllable processing times, learning effect.

School of Business and Management, Fujian Jiangxia University, Fuzhou 350108, P.R. China.
*Corresponding author: 1257290990@qq.com

the goal of which is to determine the optimal job compressions and a job sequence such that the weighted sum of total completion time, total load, and total compression cost is minimized. Applying the technique of positional weights we show that the problem remains polynomially solvable if the number of machines is fixed. We also show how to apply the technique to several single-machine scheduling problems.

This note is organized as follows: Section 2 gives the problem formulation. In Sections 3–6, some preliminary results, main result and other results are given, respectively. Finally, Section 7 gives the conclusions.

2. Problem formulation

The problem under investigation can be described as follows: There are n independent jobs represented as $\{J_1, J_2, \ldots, J_n\}$, which have to be processed on m unrelated parallel machines, denoted as $\{M_1, M_2, \ldots, M_m\}$. All the jobs are available at time zero and preemptive steps are not allowed, as the machine cannot process two or more jobs simultaneously. Let n_i be the number of jobs assigned to machine M_i and $\sum_{i=1}^m n_i = n$. Let p_{ij}, a_{ij} and b_{ij} be the normal processing time, job-dependent learning effect, and compression rate of job J_j if it is scheduled on machine M_i, respectively. In this note, we consider a general model, *i.e.*, if job J_j is scheduled in the rth position on machine M_i, its actual processing time is defined by

$$P_{ij}(x_{ij}, r) = (p_{ij} - b_{ij}x_{ij})r^{a_{ij}}, \tag{2.1}$$

where $0 \le x_{ij} \le m_{ij} \le \frac{p_{ij}}{b_{ij}}$, m_{ij} is the maximum reduction in processing time of job J_j on machine M_i.

Let C_{ij} and v_{ij} be the completion time of job J_j on machine M_i and the unit cost to compress job J_j on machine M_i, respectively. The objective is to determine the optimal job compressions and the optimal job sequence to minimize the following total cost function: $F = \theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^m C_{\max}^i + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{ij}x_{ij}$, where $\theta_1 \ge 0, \theta_2 \ge 0, \theta_3 \ge 0$ are given constants, $C_{\max}^i$ is the makespan of machine M_i, and $\sum_{i=1}^m C_{\max}^i$ is the total load. As in Agnetis *et al.* [2], Gawiejnowicz [7], and Strusevich and Rustogi [30], we denote our scheduling problem as $Rm\,|P_{ij}(x_{ij}, r) = (p_{ij} - b_{ij}x_{ij})r^{a_{ij}}|\ \theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^m C_{\max}^i + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{ij}x_{ij}$, where Rm indicates the unrelated parallel-machine setting. For the single-machine scheduling (*i.e.*, $m = 1$), the subscript i can be removed, *i.e.*, the actual processing time of J_j is $P_j(x_j, r) = (p_j - b_jx_j)r^{a_j}$, and other symbols can be defined similarly.

In Table 1, we classify in the tabular form the single-machine and parallel-machine scheduling problems with controllable processing times and/or learning effects.

3. Preliminary results

For a given vector $(n_1, n_2, \ldots, n_m)$ and a sequence of jobs on each machine, substituting $C_{i[j]} = \sum_{l=1}^j (p_{i[l]} - b_{i[l]}x_{i[l]})l^{a_{i[l]}}$, $C_{\max}^i = \sum_{j=1}^{n_i} p_{i[j]}$ and $x_{i[j]} = \frac{1}{b_{i[j]}}\left(p_{i[j]} - P_{i[j]}^A j^{-a_{i[j]}}\right)$ into $\theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^m C_{\max}^i + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{ij}x_{ij}$, we have

$$\begin{aligned}
&\theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^m C_{\max}^i + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{ij}x_{ij}\\
&\qquad = \theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} C_{i[j]} + \theta_2 \sum_{i=1}^m C_{\max}^i + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{i[j]}x_{i[j]}\\
&\qquad = \theta_1 \sum_{i=1}^m \sum_{j=1}^{n_i} (n_i - j + 1)P_{i[j]}^A + \theta_2 \sum_{i=1}^m \sum_{j=1}^{n_i} P_{i[j]}^A + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{i[j]} \left(\frac{1}{b_{i[j]}}\left(p_{i[j]} - P_{i[j]}^A j^{-a_{i[j]}}\right)\right)\\
&\qquad = \sum_{i=1}^m \sum_{j=1}^{n_i} \lambda_{ij} P_{i[j]}^A + \theta_3 \sum_{i=1}^m \sum_{j=1}^{n_i} v_{i[j]} \left(\frac{1}{b_{i[j]}}\left(p_{i[j]} - P_{i[j]}^A j^{-a_{i[j]}}\right)\right)
\end{aligned}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left(\lambda_{ij} - \frac{\theta_3 v_{i[j]} j^{-a_{i[j]}}}{b_{i[j]}} \right) P_{i[j]}^{A} + \theta_3 \sum_{i=1}^{m} \sum_{j=1}^{n_i} \frac{v_{i[j]} p_{i[j]}}{b_{i[j]}}, \tag{3.1}$$

where $\lambda_{ij} = \theta_1(n_i - j + 1) + \theta_2$.

Let

$$\Omega_{ij} = \lambda_{ij} - \frac{\theta_3 v_{i[j]} j^{-a_{i[j]}}}{b_{i[j]}}, 1 \le i \le m; 1 \le j \le n_i, \tag{3.2}$$

where $\Omega_{ij}(1 \le i \le m; 1 \le j \le n_i)$ represents the position weight of position j in the sequence π on machine M_i. Since $\theta_3 \sum_{j=1}^{n_i} \sum_{j=1}^{n} \frac{v_{i[j]} p_{i[j]}}{b_{i[j]}}$ is a constant, obviously, for any given sequence, the optimal processing times can be written as follows:

$$P_{i[j]}^{A^*} = \begin{cases} p_{i[j]} j^{a_{i[j]}}, & \text{if } \Omega_{ij} < 0, \\ (p_{i[j]} - b_{i[j]} t_{i[j]}) j^{a_{i[j]}}, & \text{if } \Omega_{ij} = 0, \\ (p_{i[j]} - b_{i[j]} m_{i[j]}) j^{a_{i[j]}}, & \text{if } \Omega_{ij} > 0, \end{cases} \tag{3.3}$$

where $0 \le t_{i[j]} \le m_{i[j]}$ and $P_{i[j]}^{A^*}(1 \le i \le m; 1 \le j \le n_i)$ represent the optimal processing time of the job in position j on machine M_i. Therefore, the optimal compressions can be obtained by

$$x_{i[j]}^{*} = \frac{1}{b_{i[j]}} \left(p_{i[j]} - P_{i[j]}^{A^*} j^{-a_{i[j]}} \right), 1 \le i \le m; 1 \le j \le n_i. \tag{3.4}$$

In order to obtain the optimal sequence of the problem $Rm\,|P_{ij}(x_{ij}, r) = (p_{ij} - b_{ij}x_{ij})r^{a_{ij}}|\,\theta_1 \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^{m} C_{\max}^{i} + \theta_3 \sum_{i=1}^{m} \sum_{j=1}^{n_i} v_{ij} x_{ij}$, for a given $(n_1, n_2, \ldots, n_m)$ vector, we formulate it as an assignment problem.

From (3.2) and (3.3), let

$$\Omega_{ijr} = \lambda_{ir} - \frac{\theta_3 v_{ij} r^{-a_{ij}}}{b_{ij}}, \; i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n_i, \tag{3.5}$$

and

$$P_{ijr} = \begin{cases} p_{ij} r^{a_{ij}}, & \text{if } \Omega_{ijr} < 0, \\ (p_{ij} - b_{ij} t_{ij}) r^{a_{ij}}, & \text{if } \Omega_{ijr} = 0, \\ (p_{ij} - b_{ij} m_{ij}) r^{a_{ij}}, & \text{if } \Omega_{ijr} > 0, \end{cases} \tag{3.6}$$

where $0 \le t_{ij} \le m_{ij}$. Furthermore, let X_{ijr} be a 0/1 variable such that $X_{ijr} = 1$ if job J_j is scheduled in position r on machine M_i, and $X_{jr} = 0$, otherwise. As in Lin [19], the optimal matching of jobs to positions can be obtained by the following assignment problem:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{r=1}^{n_i} \Omega_{ijr} P_{ijr} X_{ijr} \tag{3.7}$$

subject to

$$\sum_{j=1}^{n} X_{ijr} = 1, \qquad i = 1, 2, \ldots, m; r = 1, 2, \ldots, n_i, \tag{3.8}$$

$$\sum_{i=1}^{m} \sum_{r=1}^{n_i} X_{ijr} = 1, \qquad j = 1, 2, \ldots, n, \tag{3.9}$$

$$X_{ijr} = 0 \text{ or } 1, \qquad i = 1, 2, \ldots, m; r = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n. \tag{3.10}$$

TABLE 1. Main scheduling problems with controllable jobs and learning effects.

Problem	Complexity	Ref.
$1 \mid P_j(x_j) = p_j - x_j \mid \sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} v_j x_j$	NP-hard	Wan *et al.* [32] and Hoogeveen and Woeginger [11]
$1 \mid \mathrm{CON}, P_j(x_j) = p_j - x_j \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j) + \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Panwalkar and Rajagopalan [27]
$1 \mid \mathrm{CON/SLK}, P_j(x_j) = p_j - x_j \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j) + \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Cheng *et al.* [5]
$1 \mid \mathrm{CONW}, P_j(x_j) = p_j - x_j \mid \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d'_j + \gamma D'_j) + \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Liman *et al.* [17, 18]
$1 \mid P_j(x_j) = p_j - x_j \mid \theta_1 C_{\max} + \theta_2 TC + \theta_3 \mathrm{TADC} + \theta_4 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Wang and Xia [36]
$1 \mid P_j(x_j) = p_j - x_j \mid \theta_1 C_{\max} + \theta_2 \mathrm{TW} + \theta_3 \mathrm{TADW} + \theta_4 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Wang and Xia [36]
$1 \left\| P_j(x_j) = p_j - x_j, p_j - p'_j = m \right\| \theta_1 C_{\max} + \theta_2 TC + \theta_3 \mathrm{TADC} + \theta_4 \sum_{j=1}^{n} v x_j$	$O(n \log n)$	Wang and Xia [36]
$1 \left\| P_j(x_j) = p_j - x_j, p_j - p'_j = m \right\| \theta_1 C_{\max} + \theta_2 \mathrm{TW} + \theta_3 \mathrm{TADW} + \theta_4 \sum_{j=1}^{n} v x_j$	$O(n \log n)$	Wang and Xia [36]
$1 \mid P_j(x_j, r) = (p_j - x_j) r^{a} \mid \theta_1 C_{\max} + \theta_2 TC + \theta_3 \mathrm{TADC} + \theta_4 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Yin and Wang [39]
$1 \mid P_j(x_j, r) = (p_j - x_j) r^{a} \mid \theta_1 C_{\max} + \theta_2 \mathrm{TW} + \theta_3 \mathrm{TADW} + \theta_4 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Yin and Wang [39]
$1 \mid P_j(x_j, r) = (p_j - x_j) r^{a_j} \mid \theta_1 f + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Li *et al.* [16]
$1 \mid \mathrm{CON/SLK/DIF}, P_j(x_j, r) = (p_j - x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Li *et al.* [16]
$1 \mid \mathrm{CONW/SLKW/DIFW}, P_j(x_j, r) = (p_j - x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d'_j + \gamma D'_j) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Li *et al.* [16]
$1 \mid \mathrm{CONW}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d' + \gamma D') + \theta_2 C_{\max} + \theta_3 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid \mathrm{SLKW}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta q' + \gamma D') + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid \mathrm{DIFW}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d'_j + \gamma D'_j) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid \mathrm{CON}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid \mathrm{SLK}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta q) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid \mathrm{DIF}, P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j) + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$1 \mid P_j(x_j, r) = (p_j - b_j x_j) r^{a_j} \mid \theta_1 g + \theta_2 \sum_{j=1}^{n} v_j x_j$	$O(n^3)$	Corollary 5.1
$Rm \mid P_{ij}(x_{ij}, r) = (p_{ij} - b_{ij} x_{ij}) r^{a_{ij}} \mid \theta_1 \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^{m} C_{\max}^{i} + \theta_3 \sum_{i=1}^{m} \sum_{j=1}^{n_i} v_{ij} x_{ij}$	$O(n^{m+2})$	Theorem 4.2

Notes. w_j is the weight of job J_j, $f \in \{C_{\max}, \mathrm{TC}, \mathrm{TW}, \mathrm{TADC}, \mathrm{TADW}\}$, $g \in \{C_{\max}, \mathrm{TC}, \mathrm{TW}, \mathrm{TADC}, \mathrm{TADW}\}$.

4. Main results

The optimal solution algorithm for the problem $Rm\,|P_{ij}(x_{ij},r)=(p_{ij}-b_{ij}x_{ij})r^{a_{ij}}|\ \theta_1\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij}+\theta_2\sum_{i=1}^{m}C_{\max}^{i}+\theta_3\sum_{i=1}^{m}\sum_{j=1}^{n_i}v_{ij}x_{ij}$ can be found by the following algorithm:

Algorithm 4.1.

Input: sequence $(p_{ij},a_{ij},b_{ij},v_{ij},m_{ij})$ for $1\le i\le m, 1\le j\le n$, numbers $\theta_1,\theta_2,\theta_3$.
Output: an optimal sequence π.

Step 1: Construct set $T:=\{(n_1,n_2,\ldots,n_m)\in\mathbb{Z}:0\le n_i\le n:\sum_{i=1}^{m}n_i=n\}$;
Step 2: **For** all $(n_1,n_2,\ldots,n_m)\in T$ **do** solve the assignment problem (3.7)–(3.10);
Step 3: Calculate $\min\{F(n_1,n_2,\ldots,n_m):(n_1,n_2,\ldots,n_m)\in T\}$;
Step 4: $\pi^*\leftarrow$ the sequence corresponding to the minimum total cost computed in Step 3;
return π^*.

The remaining question is how many $(n_1,n_2,\ldots,n_m)$ vectors exist? Note that n_i may be $0,1,2,\ldots,n$ $(i=1,2,\ldots,m)$. Therefore, if we know that n_i $(i=1,2,\ldots,m-1)$, then n_m can be determined (since $\sum_{i=1}^{m}n_i=n$), an upper bound on the number of vectors $(n_1,n_2,\ldots,n_m)$ is $(n+1)^{m-1}$. If $(n_1,n_2,\ldots,n_m)$ is given, then the optimal job sequence can be obtained by assignment problem (3.7)–(3.10) in $O(n^3)$ time by using the Hungarian method [14]. Consequently, we have the following result:

Theorem 4.2. *The problem* $Rm\,|P_{ij}(x_{ij},r)=(p_{ij}-b_{ij}x_{ij})r^{a_{ij}}|\,\theta_1\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij}+\theta_2\sum_{i=1}^{m}C_{\max}^{i}+\theta_3\sum_{i=1}^{m}\sum_{j=1}^{n_i}v_{ij}x_{ij}$ *can be solved in* $O(n^{m+2})$ *time.*

Remark: For the assignment problem, finding the optimal assignment is equivalent to finding a perfect matching of maximum weight in a bipartite graph $G=(V,E)$ with weights, where V is the set of vertices and E is the set of edges. To solve it, we can apply Kuhn's sequential algorithm [14] or the parallel algorithms by Goldberg and Tarjan [8], Goldberg *et al.* [9], and Goldberg *et al.* [10] for bipartite graphs with weights, whose complexity is less.

5. Other results

In this section, some common optimality criteria can be expressed as a weighted sum, where positional weights appear. That is, for some single machine scheduling problems their optimality criteria are in the form of $\sum_{j=1}^{n}\lambda_jP_{[j]}$.

Makespan $C_{\max}=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where $\lambda_j=1$;

Total completion time $\mathrm{TC}=\sum_{j=1}^{n}C_j=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where $\lambda_j=n-j+1$;

Total waiting time $\mathrm{TW}=\sum_{j=1}^{n}W_j=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where $\lambda_j=n-j$ and W_j is the waiting time of job J_j;

Total absolute differences in completion times [12] $\mathrm{TADC}=\sum_{j=1}^{n}\sum_{i=1}^{n}|C_j-C_i|=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where $\lambda_j=(j-1)(n-j+1)$;

Total absolute differences in waiting times [4] $\mathrm{TADW}=\sum_{j=1}^{n}\sum_{i=1}^{n}|W_j-W_i|=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where $\lambda_j=(j-1)(n-j+1)$;

Common due date (CON) assignment problem [27]: $\sum_{j=1}^{n}(\alpha E_j+\beta T_j+\delta d)=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where

$$\lambda_j=\begin{cases}n\delta+(j-1)\alpha, & \text{for } j=1,2,\ldots,h;\\ \beta(n-j+1), & \text{for } j=h,h+1\ldots,n,\end{cases}\tag{5.1}$$

E_j (T_j) is the earliness (tardiness) of job J_j, and d is the common due date of all jobs $(d=C_{[h]})$.

Slack due date (SLK) assignment problem [1]: $\sum_{j=1}^{n}(\alpha E_j+\beta T_j+\delta q)=\sum_{j=1}^{n}\lambda_jP_{[j]}$, where

$$\lambda_j=\begin{cases}n\delta+j\alpha, & \text{for } j=1,2,\ldots,h-1;\\ \beta(n-j), & \text{for } j=h,h+1\ldots,n,\end{cases}\tag{5.2}$$

d_j is the due date of job J_j and $d_j = P_j^A + q$ $(q = C_{[h-1]})$.

Unrestricted due date (DIF) assignment problem [28]: $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j) = \sum_{j=1}^{n} \lambda_j P_{[j]}$, where

$$\lambda_j = \min\{\beta, \delta\}(n - j + 1), j = 1, 2, \ldots, n. \tag{5.3}$$

Common due window (CONW) assignment problem [17, 18]: $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d' + \gamma D') = \sum_{j=1}^{n} \lambda_j P_{[j]}$, where

$$\lambda_j = \begin{cases} \theta_1[n\delta + (j-1)\alpha] + \theta_2, & \text{for } j = 1, 2, \ldots, h; \\ \theta_1 n\gamma + \theta_2, & \text{for } j = h+1, h+2, \ldots, l; \\ \theta_1 \beta(n - j + 1) + \theta_2, & \text{for } j = l+1, l+2 \ldots, n, \end{cases} \tag{5.4}$$

the common due window of job J_j is $[d', d'']$, $D' = d'' - d'$ is common due window size for all jobs, $d' = C_{[h]}$ and $d'' = C_{[l]}$.

Slack due window (SLKW) assignment problem [23, 33, 37]: $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta q' + \gamma D') = \sum_{j=1}^{n} \lambda_j P_{[j]}$, where

$$\lambda_j = \begin{cases} n\delta + j\alpha, & \text{for } j = 1, 2, \ldots, h-1; \\ n\gamma, & \text{for } j = h, h+1, \ldots, l-1; \\ \beta(n - j), & \text{for } j = l, l+1 \ldots, n, \end{cases} \tag{5.5}$$

the due window of job J_j is $[d'_j, d''_j]$, $d'_j = P_j^A + q'$, $d''_j = P_j^A + q''$, $D' = d''_j - d'_j = q'' - q'$, $q' = C_{[h-1]}$ and $q'' = C_{[l-1]}$.

Unrestricted due window (DIFW) assignment problem [33]: $\sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d'_j + \gamma D'_j) = \sum_{j=1}^{n} \lambda_j P_{[j]}$, where

$$\lambda_j = \min\{\beta, \delta\}(n - j + 1), j = 1, 2, \ldots, n \tag{5.6}$$

the due window of job J_j is $[d'_j, d''_j]$, and $D'_j = d''_j - d'_j$ is due window size of job J_j.

Corollary 5.1. *If $m = 1$, then problem $1\,|P_j(x_j, r) = (p_j - b_j x_j) r^{a_j}|\, \theta_1 X + \theta_2 \sum_{j=1}^{n} v_j x_j$ can be solved in $O(n^3)$ time, where $X \in \{C_{\max}, \mathrm{TC}, \mathrm{TW}, \mathrm{TADC}, \mathrm{TADW}, \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d), \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta q), \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d_j), \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d' + \gamma D), \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta q' + \gamma D'), \sum_{j=1}^{n}(\alpha E_j + \beta T_j + \delta d'_j + \gamma D'_j)\}$.*

6. Computational experiments

In order to verify the effectiveness of Algorithm 4.1 for the problem $Rm\,|P_{ij}(x_{ij}, r) = (p_{ij} - b_{ij} x_{ij}) r^{a_{ij}}|\, \theta_1 \sum_{i=1}^{m} \sum_{j=1}^{n_i} C_{ij} + \theta_2 \sum_{i=1}^{m} C_{\max}^i + \theta_3 \sum_{i=1}^{m} \sum_{j=1}^{n_i} v_{ij} x_{ij}$, using Microsoft Visual C++ 2008, we implemented Algorithm 4.1. For each problem size, 20 instances were generated randomly and solved on a PC with a 3.10 GHz CPU, Intel Core i5-10500, and 8.00 GB RAM. The characteristics of the instances are given as follows:

(1) $n = 50, 100, 150, 200, 300, 400$, $m = 1, 3, 5, 7, 9$, and $\theta_1 = \theta_2 = \theta_3 = 1$;
(2) p_{ij} $(j = 1, 2, \ldots, n; i = 1, 2, \ldots, m)$ is uniformly distributed over [1, 100];
(3) a_{ij} $(j = 1, 2, \ldots, n; i = 1, 2)$ is uniformly distributed over [−0.50, 0];
(4) b_{ij} $(j = 1, 2, \ldots, n; i = 1, 2)$ is uniformly distributed over [1, 10] and $b_{ij} \le p_{ij}$;
(5) v_{ij} $(j = 1, 2, \ldots, n; i = 1, 2)$ is uniformly distributed over [1, 10];
(6) m_{ij} $(j = 1, 2, \ldots, n; i = 1, 2)$ is uniformly distributed over [1, 10] and $m_{ij} \le \frac{p_{ij}}{b_{ij}}$.

The computational experiments of Algorithm 4.1 are summarized as follows. The average and max CPU time (second (s)) required to find the optimal solutions is shown in Table 2. From Table 2, we can observe that the computation time of Algorithm 4.1 increases moderately as n increases from 200 to 400. Table 2 also shows that the running time grows exponentially with the number of machines (*i.e.*, m).

Table 2. Computation time of Algorithm 4.1 in s.

Jobs (n)	Machines (m)	Mean	Max
50	1	0.266	0.277
	3	0.734	0.866
	5	2.081	2.132
	7	2.928	3.019
	9	3.797	3.817
100	1	2.111	2.239
	3	8.761	9.013
	5	11.703	13.857
	7	16.247	18.315
	9	21.897	25.112
150	1	2.177	2.322
	3	6.254	7.105
	5	17.385	30.229
	7	35.648	40.138
	9	73.748	80.256
200	1	3.766	3.991
	3	11.029	12.109
	5	68.321	75.236
	7	110.833	121.701
	9	255.927	263.529
300	1	8.252	8.934
	3	21.589	23.731
	5	198.364	209.335
	7	400.523	413.965
	9	974.135	989.231
400	1	13.953	14.652
	3	36.506	38.125
	5	195.603	653.401
	7	636.357	2010.352
	9	3379.671	3435.047

7. Conclusions

In this note, the unrelated parallel-machine scheduling problem with controllable processing times and job-dependent learning effects has been considered. If the number of machines is fixed, we proved that this problem is polynomially solvable. For single-machine scheduling problems, we demonstrated that many problems can be solved in polynomial time. Further research might involve considering scheduling with other models of controllable processing times and learning effects, or considering flow shop scheduling with controllable processing times and job-dependent learning effects.

Acknowledgements. We are grateful to two anonymous referees for their helpful comments on earlier versions of this note.

References

[1] G.I. Adamopoulos and C.P. Pappis, Single machine scheduling with flow allowances. *J. Oper. Res. Soc.* **47** (1996) 1280–1285.

[2] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli and A. Soukhal, Multiagent Scheduling: Models and Algorithms. Springer-Verlag, Berlin (2014).

[3] A. Azzouz, M. Ennigrou and L.B. Said, Scheduling problems under learning effects: classification and cartography. *Int. J. Prod. Res.* **56** (2018) 1642–1661.

[4] U.B. Bagchi, Simultaneous minimization of mean and variation of flow-time and waiting time in single machine systems. *Oper. Res.* **37** (1989) 118–125.

[5] T.C.E. Cheng, C. Oğaz and X.D. Qi, Due-date assignment and single machine scheduling with compressible processing times. *Int. J. Prod. Econ.* **43** (1996) 29–35.

[6] S. Gawiejnowicz, A note on scheduling on a single processor with speed dependent on a number of executed jobs. *Inf. Process. Lett.* **57** (1996) 297–300.

[7] S. Gawiejnowicz, Models and Algorithms of Time-Dependent Scheduling. Springer-Verlag, Berlin (2020).

[8] A.V. Goldberg and R.E. Tarjan, A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.* **35** (1988) 921–940.

[9] A.V. Goldberg, S.A. Plotkin and P.M. Vaidya, Sublinear-time parallel algorithms for matching and related problems. In: 29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA (1988) 174–185.

[10] A.V. Goldberg, S.A. Plotkin, D.B. Shmoys and E. Tardos, Using interior-point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.* **21** (1992) 140–150.

[11] H. Hoogeveen and G.J. Woeginger, Some comments on sequencing with controllable processing times. *Computing* **68** (2002) 181–192.

[12] J.J. Kanet, Minimizing variation of flow time in single machine systems. *Manage. Sci.* **27** (1981) 1453–1459.

[13] S. Karhia and D. Shabtay, Single machine scheduling to minimise resource consumption cost with a bound on scheduling plus due date assignment penalties. *Int. J. Prod. Res.* **56** (2018) 3080–3096.

[14] H.W. Kuhn, The Hungarian method for the assignment problem. *Nav. Res. Logistics Q.* **2** (1955) 83–97.

[15] G. Li, M.-L. Luo, W.-J. Zhang and X.-Y. Wang, Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. *Int. J. Prod. Res.* **53** (2015) 1228–1241.

[16] L. Li, P. Yan, P. Ji and J.-B. Wang, Scheduling jobs with simultaneous considerations of controllable processing times and learning effect. *Neural Comput. App.* **29** (2018) 1155–1162.

[17] S.D. Liman, S.S. Panwalkar and S. Thongmee, A single machine scheduling problem with common due window and controllable processing times. *Ann. Oper. Res.* **70** (1997) 145–154.

[18] 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.

[19] S.-S. Lin, Due-window assignment and resource allocation scheduling with truncated learning effect and position-dependent weights. *Discrete Dyn. Nat. Soc.* **2020** (2020) 9260479.

[20] W.-W. Liu and C. Jiang, Due date assignment scheduling involving job-dependent learning effects and convex resource allocation. *Eng. Optim.* **52** (2020) 74–89.

[21] L. Liu, J.-J. Wang and X.-Y. Wang, Due-window assignment scheduling with resource processing times to minimise total resource consumption cost. *Int. J. Prod. Res.* **54** (2016) 1186–1195.

[22] 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.

[23] G. Mosheiov and D. Oron, Job-dependent due-window assignment based on a common flow allowance. *Found. Comput. Decis. Sci.* **35** (2010) 185–195.

[24] G. Mosheiov and J.B. Sidney, Scheduling with general job-dependent learning curves. *Eur. J. Oper. Res.* **147** (2003) 665–670.

[25] E. Nowicki and S. Zdrzalka, A survey of results for sequencing problems with controllable processing times. *Discrete Appl. Math.* **26** (1990) 271–287.

[26] S.S. Panwalkar and R. Rajagopalan, Single-machine sequencing with controllable processing times. *Eur. J. Oper. Res.* **59** (1992) 298–302.

[27] S.S. Panwalker, M.L. Smith and A. Seidmann, Common due-date assignment to minimize total penalty for the one machine scheduling problem. *Oper. Res.* **30** (1982) 391–399.

[28] A. Seidmann, S.S. Panwalkar and M.L. Smith, Optimal assignment of due dates for a single processor scheduling problem. *Int. J. Prod. Res.* **19** (1981) 393–399.

[29] D. Shabtay and G. Steiner, A survey of scheduling with controllable processing times. *Discrete Appl. Math.* **155** (2007) 1643–1666.

[30] V.A. Strusevich and K. Rustogi, Scheduling with Time-Changing Effects and Rate-Modifying Activities. Springer-Verlag, Berlin (2017).

[31] L. Tai, Optimizing batch-processing operations with batch-position-based learning effects. *RAIRO:OR* **55** (2020) S253–S269.

[32] G. Wan, B.P.C. Yen and C.L. Li, Single machine scheduling to minimize total compression plus weighted flow cost is NP-hard. *Inf. Process. Lett.* **79** (2001) 273–280.

[33] J.-B. Wang, L. Liu and C. Wang, Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs. *Appl. Math. Modell.* **37** (2013) 8394–8400.

[34] J.-B. Wang, D.-Y. Lv, J. Xu, P. Ji and F. Li, Bicriterion scheduling with truncated learning effects and convex controllable processing times. *Int. Trans. Oper. Res.* **28** (2021) 1573–1593.

[35] J.-B. Wang and J.-J. Wang, Research on scheduling with job-dependent learning effect and convex resource dependent processing times. *Int. J. Prod. Res.* **53** (2015) 5826–5836.

[36] J.-B. Wang and Z.-Q. Xia, Single machine scheduling problems with controllable processing times and total absolute differences penalties. *Eur. J. Oper. Res.* **177** (2007) 638–645.

[37] Y.-B. Wu, L. Wan and X.-Y. Wang, Study on due-window assignment scheduling based on common flow allowance. *Int. J. Prod. Econ.* **165** (2015) 155–157.

[38] D.-L. Yang, C.-J. Lai and S.-J. Yang, Scheduling problems with multiple due windows assignment and controllable processing times on a single machine. *Int. J. Prod. Econ.* **150** (2014) 96–103.

[39] N. Yin and X.-Y. Wang, Single-machine scheduling with controllable processing times and learning effect. *Int. J. Adv. Manuf. Technol.* **54** (2011) 743–748.

[40] S. Zdrzalka, Scheduling jobs on a single machine with release dates, delivery times and controllable processing times: worst-case analysis. *Oper. Res. Lett.* **10** (1991) 519–524.