Resource-dependent scheduling with deteriorating jobs and learning effects on unrelated parallel-machine

Yuan-Yuan Lu[a,1], Jian Jin[b], Ping Ji[c], Ji-Bo Wang[d]

[a]College of Mathematics, Jilin Normal University, Siping, Jilin 136000, China;

[b]School of Government, Beijing Normal University, Beijing 100875, China;

[c]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong;

[d]School of Science, Shenyang Aerospace University, Shenyang 110136, China

Abstract

The focus of this paper is to analyze unrelated parallel-machine resource allocation scheduling problem with learning effect and deteriorating jobs. The goal is to find the optimal sequence of jobs and the optimal resource allocation separately for minimizing the cost function includes the total load, the total completion time, the total absolute deviation of completion time, and the total resource cost. We show that the problem is polynomial time solvable if the number of machines is a given constant.

Keywords: Scheduling; Parallel-machine; Learning effect; Deteriorating jobs; Resource allocation

1 Introduction

In classical scheduling theory and model, the job processing times are assumed to be fixed and constant values (Pinedo [18]). However, we often encounter settings in which the job processing times may be changed by the phenomenon of deterioration, and/or learning, and/or resource allocation. Extensive surveys of different scheduling models and problems involving deteriorating jobs (time-dependent processing times), and/or learning, and/or resource allocation can be found in Gawiejnowicz [5], Shabtay and Steiner [20] and Biskup [2]. More recently, Wang and Wang [27] considered single machine scheduling problems with nonlinear deterioration. They showed that the makespan minimization problem remains polynomially solvable. Xu et al. [37], Lu et al. [14], and Wang and Wang [32] considered single machine group scheduling with deteriorating jobs. Yin et al. [41] considered scheduling problems with sum-of-logarithm-processing-times based deterioration. They proved that single machine makespan minimization problem can be solved in polynomial time. For the total completion time minimization problem, they also gave some results. Yin et al. [39], Yin et al. [40], and Yin et al. [43] considered scheduling problems with time-dependent processing time (deteriorating jobs). Wang and Wang [28] considered single machine scheduling with convex resource dependent processing times. For the total amount of resource consumed minimization problem subject to a constraint on total

[1] Corresponding author. E-mail addresses: luyuanyuan_jilin@163.com

weighted flow time, they proposed a branch-and-bound algorithm and a heuristic algorithm. Wei et al. [36] and Wang and Wang [29] considered single machine scheduling with time-and-resource-dependent processing times. Wang et al. [34], Hsu et al. [10], and Wang [23] considered single machine scheduling with learning effects. Eren [4], and Hsu et al. [8] considered parallel machine scheduling with learning effects. Wang and wang [26] considered flow shop scheduling with learning effects. Jiang et al. [12], Wang [22], Wang et al. [24], Wang and wang [30], and Wang et al. [25] considered single machine scheduling with learning effect and deteriorating jobs. Huang et al. [11] considered parallel machines scheduling with learning effect and deteriorating jobs. Wang et al. [33], Yin et al. [42], and Zhang et al. [44] considered resource allocation scheduling problem with learning effect and deteriorating jobs. Hsu and Yang [9] considered unrelated parallel-machine scheduling problems with deteriorating jobs and resource-dependent processing time. For two resource consumption functions and two multi-objective functions, they proved that the proposed problems were polynomial time solvable respectively, if the number of machines is fixed. Wang and Wang [31] considered unrelated parallel machines scheduling problems with deteriorating jobs and learning effect. Rudek et al. [19] considered multiprocessor scheduling problems with time-dependent processing times. For the workspan criterion, they constructed some polynomial time algorithms.

Wang [22], and Yang and Kuo [38] considered the single machine model $p_j = (a_j + \alpha t)r^b$, where a_j is the original (normal) processing time of job J_j, p_j is the actual processing time of job J_j, r is the position of job J_j when scheduled on the machine, $\alpha \geq 0$ is the deterioration rate, and $b \leq 0$ is the learning index of job J_j. Shabtay and Steiner [21] considered single machine scheduling model $p_j = a_j - \beta_j u_j$, where u_j is the amount of a non-renewable resource allocated to job J_j, with $0 \leq u_j \leq \bar{u}_j < \frac{a_j}{\beta_j}$, where $\bar{u}_j$ denote the maximum amount of resource allocated to job J_j and β_j is the positive compression rate of job J_j. Wang and Wang [31] considered unrelated parallel machines scheduling model $p_{ij} = (a_{ij} + \alpha t)r^b$, where a_{ij} is the original (normal) processing time of job J_j on machine M_i, p_{ij} is the actual processing time of job J_j on machine M_i, r is the position of job J_j when scheduled on machine M_i, $\alpha \geq 0$ is the deterioration rate, and $b \leq 0$ is the learning index of job J_j. "*The phenomena of deterioration, learning effect, and resource allocation occurring simultaneously can be found in the manual production of glass crafts by a skilled craftsman. Silicon-based raw material is first heated up (i.e., need to consume resource) in an oven until it becomes a lump of malleable dough from which the craftsman cuts pieces and shapes them according to different designs into different glass craft products. The initial time to heat up the raw material to the threshold temperature at which it can be shaped is long and so the first piece (i.e., job) has a long processing time, which includes both the heating time (i.e., the deterioration effect) and the shaping time (i.e., the normal processing time). The second piece requires a shorter time to re-heat the dough to the threshold temperature (i.e., a* *smaller deterioration effect). Similarly, the later a piece is cut from the dough, the shorter is its heating time to reach the threshold temperature. On the other hand, the pieces that are shaped later require shorter shaping times because the craftsman's productivity improves as a result of learning (Cheng et al. [3]).* This paper consider the unrelated parallel-machine scheduling problem with position, time and resource dependent processing times at the same time. This model stems from Wang and Wang [31] and Shabtay and Steiner [21].

2 Problem formulation

The problem considered in this paper can be formally described as follows. There are n independent jobs $\{J_1, J_2, \ldots, J_n\}$ to be processed on m unrelated parallel machines $\{M_1, M_2, \ldots, M_m\}$. Each of them is available at time 0. The machine can handle one job at a time, and preemption is not allowed. Let n_i denote the number of jobs assigned to $M_i (i = 1, 2, \ldots, m)$ and $P(n, m) = (n_1, n_2, \ldots, n_m)$ denote a job-allocation vector, where $(n_1 + n_2 + \ldots n_m = n)$. We assume, as in most practical situations, that $m < n$ and m is a given constant. Each job can be processed on any one of the m unrelated parallel machines.

Associated with each job $J_j (j = 1, 2, ..., n)$ on machine M_i, there is a normal processing time a_{ij}. Let p_{ij} denote the actual processing time for job J_j on machine M_i. In this paper, we consider the following unrelated parallel-machine scheduling model:

$$p_{ij} = (a_{ij} + \alpha t) f(r) - \theta_{ij} u_{ij}, \tag{1}$$

where $f(r)$ represents a factor that depends on the position of a job in the processing sequence, r is the position of job J_j when scheduled on machine M_i, t is the starting time of job J_j on machine M_i, $\alpha \geq 0$ is a common deterioration rate for all the jobs, $\theta_{ij} \geq 0$ is the positive compression rate of job J_j on machine M_i, and u_{ij} is the amount of resource that can be allocated to job J_j on machine M_i, with $0 \leq u_{ij} \leq m_{ij} < \frac{a_{ij} f(n)}{\theta_{ij}}$, where m_{ij} is the upper bound on the amount of resource that can be allocated to job J_j on machine M_i. If the values $f(r)$, $r = 1, 2, \ldots, n$, form a non-decreasing (non-increasing) sequence, we deal with a positional deterioration (learning) effect; i.e., $1 = f(1) \leq f(2) \leq \ldots \leq f(n)$ $(1 = f(1) \geq f(2) \geq \ldots \geq f(n))$.

Let $J_{i[j]}$ denote the jth job on machine M_i $C_{i[j]}$ denote the completion time of job $J_{i[j]}$ and $W_{i[j]}$ denote the waiting time of job $J_{i[j]}$. As in Hsu and Yang [9], let $C^i_{\max} = \max\{C_{ij} | j = 1, 2, \ldots, n_i\}$, $TC^i = \sum_{j=1}^{n_i} C_{ij}$ $(TW^i = \sum_{j=1}^{n_i} W_{ij})$, $TADC^i = \sum_{j=1}^{n_i} \sum_{l=j}^{n_i} |C_{ij} - C_{il}|$ $(TADW^i = \sum_{j=1}^{n_i} \sum_{l=j}^{n_i} |W_{ij} - W_{il}|)$ be the makespan of jobs, the total completion (waiting) times, and the total absolute differences in completion (waiting) times on machine M_i, where $W_{ij} = C_{ij} - p_{ij}$ be the waiting time of job J_j on machine M_i. Then, the total load, the total completion (waiting) time, and the total absolute deviation of job completion (waiting) time on all machines are $\sum_{i=1}^{m} C^i_{\max}$, $\sum_{i=1}^{m} TC^i$ $(\sum_{i=1}^{m} TADC^i)$, $\sum_{i=1}^{m} TW^i$ $(\sum_{i=1}^{m} TADW^i)$, respectively.

Criteria TC^i and TW^i (and $TADC^i$ and $TADW^i$) are strictly related, since $W_{ij} = C_{ij} - p_{ij}$. Thus, each result concerning TC^i ($TADC^i$) will concern TW^i ($TADW^i$) (Bagchi [1], Mor and G. Mosheiov [15]). Therefore, our goal is only to determine the optimal resource allocations and the optimal sequence of jobs on all machines so that the corresponding value of the following cost function be optimal:

$$F = \delta_1 \sum_{i=1}^{m} C_{\max}^i + \delta_2 \sum_{i=1}^{m} TC^i + \delta_3 \sum_{i=1}^{m} TADC^i + \delta_4 \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij}, \tag{2}$$

where weights $\delta_1 \geq 0, \delta_2 \geq 0$, $\delta_3 \geq 0$ and $\delta_4 \geq 0$ are given constants (the decision-maker selects the weights $\delta_1, \delta_2, \delta_3, \delta_4$) and G_{ij} is the per time unit cost associated with the resource allocation. Then, using the three-field notation introduced by Graham et al. [7], the corresponding scheduling problem is denoted by $Rm|\text{LDRA}|F$, where LDRA denotes Learning-Deteriorating-Resource Allocation (i.e., the model of Eq. (1)).

3 Optimal resource allocation

In this section, we will prove that the proposed problems can be solved in polynomial time. Note that $C_{i[j]} = \sum_{l=1}^{j} p_{i[l]}$, $C_{\max}^i = \sum_{j=1}^{n_i} p_{i[j]}$, $TC^i = \sum_{j=1}^{n_i} C_{i[j]}$ and $TADC^i = \sum_{j=1}^{n_i} (j-1)(n_i - j + 1) p_{i[j]}$ (Kanet [13]).

Let $p_{i[r]}$ and $a_{i[r]}$ denote the actual processing time and the normal processing time of a job when it is scheduled in position r on machine M_i, respectively. Then the completion times of jobs can be expressed as follows (by induction):

$$\begin{aligned}
C_{i[1]} &= a_{i[1]} f(1) - \theta_{i[1]} u_{i[1]} \\
C_{i[2]} &= a_{i[1]} f(1) - \theta_{i[1]} u_{i[1]} + (a_{i[2]} + \alpha(a_{i[1]} f(1) - \theta_{i[1]} u_{i[1]})) f(2) - \theta_{[2]} u_{i[2]} \\
&= a_{i[2]} f(2) - \theta_{[2]} u_{[2]} + (1 + \alpha f(2))(a_{i[1]} f(1) - \theta_{i[1]} u_{i[1]}) \\
&\cdots \\
C_{i[j]} &= \sum_{k=1}^{j} \prod_{l=k+1}^{j} (1 + \alpha f(l))(a_{i[k]} f(k) - \theta_{i[k]} u_{i[k]}) \\
&\cdots \\
C_{i[n_i]} &= \sum_{k=1}^{n_i} \prod_{l=k+1}^{n_i} (1 + \alpha f(l))(a_{i[k]} f(k) - \theta_{i[k]} u_{i[k]})
\end{aligned} \tag{3}$$

Let $i[r]$ denote the rth job on machine M_i, from (3), the actual processing time of job $J_{i[r]}$ can be expressed as follows:

$$\begin{aligned}
p_{i[r]} &= (a_{i[r]} + \alpha C_{i[r-1]}) f(r) - \theta_{i[r]} u_{i[r]} \\
&= a_{i[r]} f(r) - \theta_{i[r]} u_{i[r]} + \alpha f(r) \left(\sum_{k=1}^{r-1} \prod_{l=k+1}^{r-1} (1 + \alpha f(l))(a_{i[k]} f(k) - \theta_{i[k]} u_{i[k]}) \right),
\end{aligned} \tag{4}$$

Let $i[r]$ denote where $C_{[0]} = 0$.

From (2) and (4), we have

$$\begin{aligned}
F &= \delta_1 \sum_{i=1}^{m} C^i_{\max} + \delta_2 \sum_{i=1}^{m} TC^i + \delta_3 \sum_{i=1}^{m} TADC^i + \delta_4 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}u_{ij} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n_i} [\delta_1 + \delta_2(n_i+1-j) + \delta_3(j-1)(n_i-j+1)]p_{i[j]} + \delta_4 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{i[j]}u_{i[j]} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n_i} \omega_{ij}p_{i[j]} + \delta_4 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{i[j]}u_{i[j]} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n_i} \omega_{ij}\left(a_{i[r]}f(r) - \theta_{i[r]}u_{i[r]} + \alpha f(r)\left(\sum_{k=1}^{r-1}\prod_{l=k+1}^{r-1}(1+\alpha f(l))(a_{i[k]}f(k) - \theta_{i[k]}u_{i[k]})\right)\right) \\
&\quad +\delta_4 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{i[j]}u_{i[j]} \\
&= \sum_{i=1}^{m}[\omega_{i1}(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]}) \\
&\quad +\omega_{i2}(a_{i[2]}f(2) - \theta_{i[2]}u_{i[2]} + \alpha f(2)(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]})) \\
&\quad +\omega_{i3}(a_{i[3]}f(3) - \theta_{i[3]}u_{i[3]} + \alpha f(3)(a_{i[2]}f(2) - \theta_{i[2]}u_{i[2]} + (1+\alpha f(2))(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]}))) \\
&\quad +\omega_{i4}(a_{i[4]}f(4) - \theta_{i[4]}u_{i[4]} + \alpha f(4)(a_{i[3]}f(3) - \theta_{i[3]}u_{i[3]} + (1+\alpha f(3))(a_{i[2]}f(2) - \theta_{i[2]}u_{i[2]}) \\
&\quad +(1+\alpha f(2))(1+\alpha f(3))(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]}))) \\
&\quad +\ldots \\
&\quad +\omega_{i,n_i-1}(a_{i[n_i-1]}f(n_i-1) - \theta_{i[n_i-1]}u_{i[n_i-1]} + \alpha f(n_i-1)(a_{i[n_i-2]}f(n_i-2) - \theta_{i[n_i-2]}u_{i[n_i-2]} \\
&\quad +(1+\alpha f(n_i-2))(a_{i[n_i-3]}f(n_i-3) - \theta_{i[n_i-3]}u_{i[n_i-3]}) \\
&\quad +\ldots + \prod_{l=3}^{n_i-2}(1+\alpha f(l))(a_{i[2]}f(2) - \theta_{i[2]}u_{i[2]}) + \prod_{l=2}^{n_i-2}(1+\alpha f(l))(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]}))) \\
&\quad +\omega_{in_i}(a_{i[n_i]}f(n_i) - \theta_{i[n_i]}u_{i[n_i]} + \alpha f(n_i)(a_{i[n_i-1]}f(n_i-1) - \theta_{i[n_i-1]}u_{i[n_i-1]} \\
&\quad +(1+\alpha f(n_i-1))(a_{i[n_i-2]}f(n-2) - \theta_{i[n_i-2]}u_{i[n_i-2]}) \\
&\quad +\ldots + \prod_{l=3}^{n_i-1}(1+\alpha f(l))(a_{i[2]}f(2) - \theta_{i[2]}u_{i[2]}) + \prod_{l=2}^{n_i-1}(1+\alpha f(l))(a_{i[1]}f(1) - \theta_{i[1]}u_{i[1]})))] \\
&\quad +\delta_4 \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{i[j]}u_{i[j]} \\
&= \sum_{i=1}^{m}[(\omega_{i1} + \alpha f(2)\omega_{i2} + \alpha f(3)(1+\alpha f(2))\omega_{i3} + \alpha f(4)(1+\alpha f(2))(1+\alpha f(3))\omega_{i4}
\end{aligned}$$

$$+\ldots+\alpha f(n_i)\prod_{l=2}^{n_i-1}(1+\alpha f(l))\omega_{in_i})(a_{i[1]}f(1)-\theta_{i[1]}u_{i[1]})$$
$$+\left(\omega_{i2}+\alpha f(3)\omega_{i3}+\alpha f(4)(1+\alpha f(3))\omega_{i4}+\ldots+\alpha f(n_i)\prod_{l=3}^{n_i-1}(1+\alpha f(l))\omega_{in_i}\right)$$
$$\times(a_{i[2]}f(2)-\theta_{i[2]}u_{i[2]})$$
$$+\left(\omega_{i3}+\alpha f(4)\omega_{i4}+\alpha f(5)(1+\alpha f(4))\omega_{i5}+\ldots+\alpha f(n_i)\prod_{l=4}^{n_i-1}(1+\alpha f(l))\omega_{in_i}\right)$$
$$\times(a_{i[3]}f(3)-\theta_{i[3]}u_{i[3]})$$
$$+\ldots+(\omega_{i,n_i-1}+\alpha f(n_i)\omega_{in_i})(a_{i[n_i-1]}f(n_i-1)-\theta_{i[n_i-1]}u_{i[n_i-1]})$$
$$+\omega_{in}(a_{i[n_i]}f(n_i)-\theta_{i[n_i]}u_{i[n_i]})]+\delta_4\sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{i[j]}u_{i[j]}$$
$$=\sum_{i=1}^{m}\sum_{j=1}^{n_i}\Omega_{ij}f(j)a_{i[j]}+\sum_{i=1}^{m}\sum_{j=1}^{n_i}(\delta_4G_{i[j]}-\theta_{i[j]}\Omega_{ij})u_{i[j]}, \quad (6)$$

where $\omega_{ij}=\delta_1+\delta_2(n_i+1-j)+\delta_3(j-1)(n_i-j+1)$ and

$$\Omega_{i1} = \omega_{i1}+\alpha f(2)\omega_{i2}+\alpha f(3)(1+\alpha f(2))\omega_{i3}+\alpha f(4)(1+\alpha f(2))(1+\alpha f(3))\omega_{i4}$$
$$+\ldots+\alpha f(n_i)\prod_{l=2}^{n_i-1}(1+\alpha f(l))\omega_{in_i}$$
$$\ldots$$
$$\Omega_{ik} = \omega_{ik}+\alpha f(k+1)\omega_{i,k+1}+\alpha f(k+2)(1+\alpha f(k+1))\omega_{i,k+2}+\ldots+\alpha f(n_i)\prod_{l=k+1}^{n_i-1}(1+\alpha f(l))\omega_{in_i}$$
$$\ldots$$
$$\Omega_{in_i} = \omega_{in_i}.$$

Theorem 1 *Given a sequence, for the problem $Rm|LDRA|F$, the optimal resource allocation can be determined as follows:*

$$u^*_{i[j]}=\begin{cases}0, & \text{if } \delta_4G_{i[j]}-\theta_{i[j]}\Omega_{ij}>0,\\ u_{i[j]}, & \text{if } \delta_4G_{i[j]}-\theta_{i[j]}\Omega_{ij}=0,\\ m_{i[j]}, & \text{if } \delta_4G_{i[j]}-\theta_{i[j]}\Omega_{ij}<0,\end{cases} \quad (7)$$

where $0\le u_{i[j]}\le m_{i[j]}$ and $u^*_{i[j]}, i=1,2,\ldots,m; j=1,2,\ldots,n_i$, represents the optimal resource allocation of the job in position j on machine M_i.

Proof. For the $Rm|$LDRA$|F$ problem, substituting (1) for $p_{i[j]}$ into (2) and taking the derivative by $u_{i[j]}$ to Eq. (6), we have $\frac{df(\pi,u)}{du_{i[j]}}=\delta_4G_{i[j]}-\theta_{i[j]}\Omega_{ij}$ for $i=1,2,\ldots,m$ and $j=1,2,\ldots,n_i$.

Then, for any sequence, the optimal resource allocation of a job in a position with a negative $\delta_4 G_{i[j]} - \theta_{i[j]}\Omega_{ij}$ should be its upper bound on the amount of resource $m_{i[j]}$, and the optimal resource allocation of a job in a position with a positive $\delta_4 G_{i[j]} - \theta_{i[j]}\Omega_j$ should be 0. If $\delta_4 G_{i[j]} - \theta_{i[j]}\Omega_{ij} = 0$, then the optimal resource allocation of the job in this position may be any value between 0 and $m_{i[j]}$. □

4 Optimal sequences

In order to obtain the optimal sequence, for a given job-allocation vector $P(n, m) = (n_1, n_2, \ldots, n_m)$, we formulate the $Rm|\text{LDRA}|F$ problem as an assignment problem.

Let

$$\lambda_{ijr} = \begin{cases} \Omega_{ir} f(r) a_{ij}, & \text{if} \quad \delta_4 G_{ij} - \theta_{ij}\Omega_{ir} \geq 0, \\ \Omega_{ir} f(r) a_{ij} + (\delta_4 G_{ij} - \theta_{ij}\Omega_{ir}) m_{ij}, & \text{if} \quad \delta_4 G_{ij} - \theta_{ij}\Omega_{ir} < 0. \end{cases} \tag{8}$$

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 $x_{ijr} = 0$, otherwise. As in Panwalkar and Rajagopalan [17], the optimal matching of jobs to positions requires a solution for the following assignment problem:

$$\min \sum_{i=1}^{m} \sum_{r=1}^{n_i} \sum_{j=1}^{n} \lambda_{ijr} x_{ijr} \tag{9}$$

subject to

$$\sum_{i=1}^{m} \sum_{r=1}^{n_i} x_{ijr} = 1, \quad j = 1, 2, \ldots, n,$$

$$\sum_{j=1}^{n} x_{ijr} = 1, \quad i = 1, 2, \ldots, m; r = 1, 2, \ldots, n_i,$$

$$x_{ijr} = 0 \text{ or } 1, \quad i = 1, 2, \ldots, m; r = 1, 2, \ldots, n_i; j = 1, 2, \ldots, n.$$

The constraints make sure that each job is scheduled exactly once and each position on each machine is taken by one job.

Recall that solving an assignment problem of size n requires an effort of $O(n^3)$ (using the well-known Hungarian method).

Now we give an optimal algorithm for the problem $Rm|\text{LDRA}|F$.

Algorithm 1

For $n_1 = 0, 1, 2, \ldots, n$.

For $n_2 = 1, 2, \ldots, n - n_1$.

For $n_k = 1, 2, \ldots, n - \sum_{i=1}^{k-1} n_i$.

For $n_m = 1, 2, \ldots, n - \sum_{i=1}^{m-1} n_i$.

Find the minimum total cost for $P(n, m) = (n_1, n_2, \ldots, n_m)$ using assignment problem (9).

Find $(n_1^*, n_2^*, \ldots, n_m^*)$ corresponding to the lowest total cost to determine the optimal job sequence, and denoted by $\pi^* = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$.

Calculate the optimal resources by using equation (7).

Next, the question is how many $P(n, m) = (n_1, n_2, \ldots, n_m)$ vectors exist. Note that n_i may be $0, 1, 2, \ldots, n$ for $i = 1, 2, \ldots, m$. So if we get the numbers of jobs on the first $m - 1$ machines, the number of jobs processed on the last machine is then determined uniquely due to $n_1 + n_2 + \ldots + n_m = n$. Therefore, the upper bound of the number of $P(n, m)$ vectors is $(n + 1)^{m-1}$. Based on the above analysis, we have the following result.

Theorem 2 *The problem $Rm|LDRA|F$ can be solved by Algorithm 1 in $O(n^{m+2})$ time, i.e., the problem is polynomially solvable because m is a constant.*

Proof. To solve the Rm|LDRA|F problem, a maximum number $(n + 1)^{m-1}$ of assignment problems need to be solved, and each assignment problem can be solved in $O(n^3)$ time (using the well-known Hungarian method). Hence, the total time of the Rm|LDRA|F problem is solved in $O(n^{m+2})$ time. □

Remark: Similarly, the problem Rm|LDRA|F can be solved in $O(n^{m+2})$ time, i.e., the problem is polynomially solvable because m is a constant.

The following instance gives the working of Theorem 2 for the problem Rm|LDRA|F.

Example 1. Let $f(r) = r^b, m = 2, n = 5, \alpha = 0.1, b = -0.3, \delta_1 = \delta_2 = \delta_3 = \delta_4 = 1$, and the parameters for each job as given in Table 1.

Table 1. Data of Example 1

J_j	J_1	J_2	J_3	J_4	J_5
a_{1j}	35	26	19	37	28
a_{2j}	25	37	28	20	26
θ_{1j}	3	2	3	5	4
θ_{2j}	5	4	2	3	2
m_{1j}	5	6	3	4	4
m_{2j}	2	5	7	3	6
G_{1j}	12	17	15	14	16
G_{2j}	14	15	16	13	17

Solution. When $n_1 = 0, n_2 = 5$, the positional weights on machine M_2 are: $\omega_{21} = 6, \omega_{22} = 9, \omega_{23} = 10, \omega_{24} = 9, \omega_{25} = 6$, $\Omega_{21} = 8.6542, \Omega_{22} = 10.7787, \Omega_{23} = 10.9884, \Omega_{24} = 9.3702, \Omega_{25} =$

6. The optimal schedule on machine M_2 is $[J_4, J_1, J_3, J_2, J_5]$, and the optimal resources are $u_{24} = 3, u_{21} = 2, u_{23} = 7, u_{22} = 5, u_{25} = 0$. The total cost is 665.3226.

When $n_1 = 1, n_2 = 4$, the positional weights on machines M_1 and M_2 are: $\omega_{11} = 2, \omega_{21} = 5, \omega_{22} = 7, \omega_{23} = 7, \omega_{24} = 5$, $\Omega_{11} = 2, \Omega_{21} = 6.4953, \Omega_{22} = 7.8571, \Omega_{23} = 7.3299, \Omega_{24} = 5$. The optimal schedule on machine M_1 is $[J_3]$, and on machine M_2 is $[J_1, J_4, J_2, J_5]$, and the optimal resources are $u_{13} = 0, u_{21} = 2, u_{24} = 3, u_{22} = 5, u_{25} = 0$. The total cost is 401.7947.

When $n_1 = 2, n_2 = 3$, the positional weights on machines M_1 and M_2 are: $\omega_{11} = 3, \omega_{12} = 3, \omega_{21} = 4, \omega_{22} = 5, \omega_{23} = 4$, $\Omega_{11} = 3.2437, \Omega_{12} = 3, \Omega_{21} = 4.7172, \Omega_{22} = 5.2877, \Omega_{23} = 4$. The optimal schedule on machine M_1 is $[J_3, J_2]$, and on machine M_2 is $[J_4, J_1, J_5]$, and the optimal resources are $u_{13} = 0, u_{12} = 0, u_{24} = 3, u_{21} = 2, u_{25} = 0$. The total cost is 315.8946.

When $n_1 = 3, n_2 = 2$, the positional weights on machines M_1 and M_2 are: $\omega_{11} = 4, \omega_{12} = 5, \omega_{13} = 4, \omega_{21} = 3, \omega_{22} = 3$, $\Omega_{11} = 4.7172, \Omega_{12} = 5.2877, \Omega_{13} = 4, \Omega_{21} = 3.2437, \Omega_{22} = 3$. The optimal schedule on machine M_1 is $[J_5, J_3, J_2]$, and on machine M_2 is $[J_1, J_4]$, and the optimal resources are $u_{15} = 4, u_{13} = 0, u_{12} = 0, u_{21} = 2, u_{24} = 0$. The total cost is 363.0957.

When $n_1 = 4, n_2 = 1$, the positional weights on machines M_1 and M_2 are: $\omega_{11} = 5, \omega_{12} = 7, \omega_{13} = 7, \omega_{14} = 5, \omega_{21} = 2$, $\Omega_{11} = 6.4953, \Omega_{12} = 7.8571, \Omega_{13} = 7.3299, \Omega_{14} = 5, \Omega_{21} = 2$. The optimal schedule on machine M_1 is $[J_3, J_5, J_4, J_2]$, and on machine M_2 is $[J_1]$, and the optimal resources are $u_{13} = 3, u_{15} = 4, u_{14} = 4, u_{12} = 0, u_{21} = 2$. The total cost is 465.7744.

When $n_1 = 5, n_2 = 0$, the positional weights on machine M_1 are: $\omega_{11} = 6, \omega_{12} = 9, \omega_{13} = 10, \omega_{14} = 9, \omega_{15} = 6$, $\Omega_{11} = 8.6542, \Omega_{12} = 10.7787, \Omega_{13} = 10.9884, \Omega_{14} = 9.3702, \Omega_{15} = 6$. The optimal schedule on machine M_1 is $[J_3, J_5, J_4, J_1, J_2]$, and the optimal resources are $u_{13} = 5, u_{15} = 4, u_{14} = 4, u_{11} = 5, u_{12} = 0$. The total cost is 628.9458.

Hence, The optimal schedule on machine M_1 is $[J_3, J_2]$, and on machine M_2 is $[J_4, J_1, J_5]$, and the optimal resources are $u_{13} = 0, u_{12} = 0, u_{24} = 3, u_{21} = 2, u_{25} = 0$. The total cost is 315.8946.

Next, some computational experiments are conducted to test the problem $Rm|\text{LDRA}|F$ against computational time by using Algorithm 1. Algorithm 1 was coded in VC++ 6.0 and implemented on a Pentium-V with 2G CPU personal computer. The normal processing times a_{ij} were generated from a uniform distribution over [1, 100], θ_{ij} from a uniform distribution over [1, 10], G_{ij} from a uniform distribution over [1, 10], and m_{ij} from a uniform distribution over $[0, \frac{a_{ij} n^b}{\theta_{ij}}]$. Let $f(r) = r^b$, $m = 2, m = 3, m = 4, m = 5, m = 6$ and $n = 10, n = 20, n = 30, n = 40, n = 50, n = 60$, $\alpha = 0.01$, $b = -0.3, \delta_1 = \delta_2 = \delta_3 = \delta_4 = 1$. The mean computational time (in second) is computed for 50 test problems in each condition (see Table 2). For the problem $Rm|\text{LDRA}|F$, the results shown in Table 2 reveal that Algorithm 1 can solve a medium-scaled case. Since the proposed algorithm is inefficient for greater values of m, some fast heuristics can be provided and analysed, please refer to reference Okołowski and Gawiejnowicz [16].

Table 2. The CPU time (in second) for Algorithm 1

n	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
10	0.017	0.012	0.043	0.128	0.359
20	0.083	0.465	0.278	14.289	64.887
30	0.571	4.368	36.970	269.244	1649.266
40	2.336	21.941	240.321	2287.474	-
50	6.788	78.619	1079.599	11889.214	-
60	11.194	219.455	-	-	-

5 Conclusions

This research considered unrelated parallel-machine resource allocation problem with learning effect and deteriorating jobs. The objective function is to minimize a cost function containing total load, total completion time, total absolute differences in completion times and total resource cost. We have showed that the proposed problem is polynomial time solvable when the number of machines m is a fixed constant. In future research, we plan to explore more general position-time-resource-dependent processing times models, consider other types of process compressibility independently and/or simultaneously, and extend the problems to flow shop, job shop (Weckman et al. [35] and Geyik and Dosdoğru [6]) machine settings or group technology environments.

Acknowledgements— This research was supported by the Science and Technology Development Project of Jilin province of China (Grant No. 20140520057JH), The Hong Kong Polytechnic University (Project 4-BCBJ) and the National Natural Science Foundation of China (Grant No. 71471120).

References

[1] U.B. Bagchi, Simultaneous minimization of mean and variation of flow-time and waiting time in single machine systems, Operations Research, Vol.37, pp.118-125, 1989.

[2] D. Biskup, A state-of-the-art review on scheduling with learning effects, European Journal of Operational Research, Vol.188, pp.315-329, 2008.

[3] T.C.E. Cheng, C.-C. Wu, W.-C. Lee, Some scheduling problems with deteriorating jobs and learning effects, Computers & Industrial Engineering, Vol.54, pp.972-982, 2008.

[4] T. Eren, A bicriteria parallel machine scheduling with a learning effect of setup and removal times, Applied Mathematical Modelling, Vol.33, pp.1141-1150, 2009.

[5] S. Gawiejnowicz, Time-Dependent Scheduling, Springer, Berlin, 2008.

[6] F. Geyik, A.T. Dosdoğru, Process plan and part routing optimization in a dynamic flexible job shop scheduling environment: an optimization via simulation approach, Neural Computing and Applications, Vol.23, pp.1631-1641, 2013.

[7] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics, Vol.5, pp.287-326, 1979.

[8] C.-J. Hsu, W.-H. Kuo, and D.-L. Yang, Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects, Applied Mathematical Modelling. Vol. 35, pp.1492-1496, 2011.

[9] C.-J. Hsu, D.-L. Yang, Unrelated parallel-machine scheduling with position-dependent deteriorating jobs and resource-dependent processing time, Optimization Letters, Vol.8, pp.519-531, 2014.

[10] C.-J. Hsu, S.-J. Yang, and D.-L. Yang, Two due date assignment problems with position-dependent processing time on a single-machine, Computers & Industrial Engineering, Vol.60, pp.796-800, 2011.

[11] X. Huang, M.-Z. Wang, P. Ji, Parallel machines scheduling with deteriorating and learning effects, Optimization Letters, Vol.8, pp.493-500, 2014.

[12] Z. Jiang, F. Chen, H. Kang, Single-machine scheduling problems with actual time-dependent and job-dependent learning effect, European Journal of Operational Research, Vol.227, pp.76-80, 2013.

[13] J.J. Kanet, Minimizing variation of flow time in single machine systems, Management Science, Vol.27, pp.1453-1459, 1981.

[14] Y.-Y. Lu, J.-J. Wang, J.-B. Wang, Single machine group scheduling with decreasing time-dependent processing times subject to release dates, Applied Mathematics and Computation, Vol.234, pp.286-292, 2014.

[15] B. Mor and G. Mosheiov, Total absolute deviation of job completion times on uniform and unrelated machines, Computers and Operations Research, Vol.38, pp.660-665, 2011.

[16] D. Okołowski, S. Gawiejnowicz, Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect, Computers & Industrial Engineering, Vol.59, pp.272-279, 2010.

[17] S.S. Panwalkar, and R. Rajagopalan, Single machine sequencing with controllable processing times, European Journal of Operational Research, Vol.59, pp.298-302, 1992.

[18] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, Upper Saddle River, NJ: Prentice-Hall, 2002.

[19] R. Rudek, A. Rudek, A. Kozik, The solution algorithms for the multiprocessor scheduling with workspan criterion, Expert Systems with Applications, Vol.40, pp.2799-2806, 2013.

[20] D. Shabtay, and G. Steiner, A survey of scheduling with controllable processing times, Discrete Applied Mathematics, Vol.155, pp.1643-1666, 2007.

[21] D. Shabtay and G. Steiner, The single-machine earlinesstardiness scheduling problem with due date assignment and resource-dependent processing times, Annals of Operations Research, vol.159, no.1, pp. 25-40, 2008.

[22] J.-B. Wang, A note on scheduling problems with learning effect and deteriorating jobs, International Journal of Systems Science, Vol.37, pp.827-833, 2006.

[23] J.-B. Wang, Single-machine scheduling with a sum-of-actual-processing-time-based learning effect, Journal of the Operational Research Society, Vol.61, pp.172-177, 2010.

[24] J.-B. Wang, C.-J. Hsu, and D.-L. Yang, Single-machine scheduling with effects of exponential learning and general deterioration, Applied Mathematical Modelling, Vol.37: 2293-2299, 2013.

[25] J.-B. Wang, L. Liu, and C. Wang, Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs, Applied Mathematical Modelling, Vol.37, pp.8394-8400, 2013.

[26] J.-B. Wang, and M.-Z. Wang, Worst-case behavior of simple sequencing rules in flow shop scheduling with general position-dependent learning effects, Annals of Operations Research, Vol.191, pp.155-169, 2011.

[27] J.-B. Wang, and M.-Z. Wang, Single-machine scheduling with nonlinear deterioration, Optimization Letters, Vol.6, pp.87-98, 2012.

[28] J.-B. Wang, and M.-Z. Wang, Single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time, Computers & Operations Research, Vol.39, pp.492-497, 2012.

[29] X.-R. Wang, J.-J. Wang, Single-machine scheduling with convex resource dependent processing times and deteriorating jobs, Applied Mathematical Modelling, Vol.37, 2388-2393, 2013.

[30] X.-Y. Wang, and J.-J. Wang, Scheduling problems with past-sequence-dependent setup times and general effects of deterioration and learning, Applied Mathematical Modelling, Vol.37, pp.4905-4914, 2013.

[31] X.-Y. Wang, J.-J. Wang, Scheduling deteriorating jobs with a learning effect on unrelated parallel machines, Applied Mathematical Modelling, Vol.38, pp.5231-5238, 2014.

[32] J.-B. Wang, J.-J. Wang, Single machine group scheduling with time dependent processing times and ready times, Information Sciences, Vol.275, pp.226-231, 2014.

[33] J.-B. Wang, M.-Z. Wang, and P. Ji, Scheduling jobs with processing times dependent on position, starting time and allotted resource, Asia-Pacific Journal of Operational Research, vol.29, Article ID 1250030, 2012.

[34] D. Wang, M.-Z. Wang, and J.-B. Wang, Single-machine scheduling with learning effect and resource-dependent processing times, Computers & Industrial Engineering, Vol.59, pp.458-462, 2010.

[35] G. Weckman, A.A. Bondal, M.M. Rinder, W.A. Young II, Applying a hybrid artificial immune systems to the job shop scheduling problem, Neural Computing and Applications, Vol.21, pp.1465-1475, 2012.

[36] C.-M. Wei, J.-B. Wang, and P. Ji, Single-machine scheduling with time-and-resource-dependent processing times, Applied Mathematical Modelling, Vol.36, pp.792-798, 2012.

[37] Y.-T. Xu, Y. Zhang, X. Huang, Single-machine ready times scheduling with group technology and proportional linear deterioration, Applied Mathematical Modelling, Vol.38, pp.384-391, 2014.

[38] D.-L. Yang, W.-H Kuo, Single-machine scheduling with both deterioration and learning effects, Annals of Operations Research, Vol. 172, pp. 315-327, 2009.

[39] Y. Yin, T.C.E. Cheng, C.-C. Wu, Scheduling with time-dependent processing times, Mathematical Problems in Engineering, Vol.2014, Article ID 201421, 2 pages, 2014.

[40] Y. Yin, T.C.E. Cheng, L. Wan, C.-C. Wu, J. Liu, Two-agent single-machine scheduling with deteriorating jobs, Computers & Industrial Engineering, Vol.81, pp.177-185, 2015.

[41] N. Yin, L. Kang, P. Ji, J.-B. Wang, Single machine scheduling with sum-of-logarithm-processing-times based deterioration, Information Sciences, Vol.274, pp.303-309, 2014.

[42] N. Yin, L. Kang, and X.-Y. Wang, Single-machine group scheduling with processing times dependent on position, starting time and allotted resource, Applied Mathematical Modelling, Vol.38, pp.4602-4613, 2014.

[43] Y. Yin, W.-H. Wu, T.C.E. Cheng, C.-C. Wu, Single-machine scheduling with time-dependent and position-dependent deteriorating jobs, International Journal of Computer Integrated Manufacturing (2014), http://dx.doi.org/10.1080/0951192X.2014.900872.

[44] G.-Q. Zhang, J.-J. Wang, and Y.-J. Liu, Scheduling jobs with variable job processing times on unrelated parallel machines, The Scientific World Journal, Vol.2014, Article ID 242107, 7 pages, 2014.