RAIRO-Oper. Res. 50 (2016) 733–748
DOI: 10.1051/ro/2016007

RAIRO Operations Research
www.rairo-ro.org

SINGLE-MACHINE PAST-SEQUENCE-DEPENDENT SETUP TIMES SCHEDULING WITH RESOURCE ALLOCATION AND LEARNING EFFECT

Zhanguo Zhu[1,2], Feng Chu[2], Yugang Yu[3] and Linyan Sun[4]

Abstract. This paper addresses single-machine scheduling problem with resource allocation and learning effect in the background of past-sequence-dependent (p-s-d) setup times. In the proposed model of this paper, the actual job processing times are dependent on learning effect and the amount of resource allocated, and the setup times are proportional to the length of the already processed jobs. The resource function used here is a general convex one. The optimal job sequence and the optimal amount of resource allocated to each job are determined jointly for the objective function yielded by a combination of the total completion time, total absolute differences in completion times, and the total resource consumption. Besides, we also discuss some extension and special cases of this problem. It is shown that all the problems under study are polynomially solvable while the complexity results are different.

Mathematics Subject Classification. 90B35.

Received May 20, 2015. Accepted January 21, 2016.

1. Introduction

Operation management (OM) is a multidisciplinary field investigating design, management, and improvement of operation systems that are devoted to the production and delivery of products and services [16, 42]. The operation systems generally include the material, machine, and consumable resources, as well as humans, processes that are used to realize the production and delivery. Planning and scheduling are important processes within this system since they define operational performance and productivity [39]. However, most traditional research on scheduling mainly focuses on material and machines. The issues such as human behaviors, limited resources, and new form of setup times existing in realistic scheduling process that may bring great challenges to traditional scheduling results are involved relative little.

Human behaviors make operations systems increasingly difficult to handle in manufacturing and service industries. A new research domain called "Behavioral Operations Management" (BOM) that investigates how behavioral factors influence operations systems has been very attractive in recent years. A stream of research on

Keywords. Scheduling, p-s-d setup times, resource allocation, learning effect.

[1] College of Economics and Management, Nanjing Agricultural University, Nanjing 210095, P.R. China

[2] Laboratoire d'informatique, biologie intégrative et systèmes complexes (IBISC), EA 4526, Université d'Evry Val d'Essonne, 91020 Evry cedex, France. feng.chu@ibisc.univ-evry.fr

[3] School of Management, University of Science and Technology of China, Hefei, P.R. China

[4] School of Management, Xi'an Jiaotong University, Xi'an, 710049 Shaanxi Province, P.R. China

BOM is to analyze the influences of human behaviors on scheduling that is a specific sub-field of OM. Learning effect of workers, the repetition of same/similar tasks leading to production efficiency improvement, is one of the most important human behaviors. After Biskup [6] and Cheng and Wang [11] initiated to discuss scheduling problems with learning effect, many researchers have highlighted studies on them in different scheduling environments. Two kinds of models are usually used to describe learning effect, job-independent and job-dependent. For the first one, the actual processing time of job J_j if it is scheduled in position r in the sequence is $p_{jr} = p_j r^a$, where p_j is the nominal processing time of job j, and a is the negative learning index [6, 30]. For the second one, the actual processing time is $p_{jr} = p_j r^{a_j}$, where a_j is the negative job-dependent learning index [23, 36]. In addition, there also exist other learning effect models in the literature such as time-dependent [27] and general learning functions [24]. Biskup [7] reviewed comprehensively related learning models and scheduling problems in his survey. In this work, we mainly discuss the learning models introduced above.

Another important issue in scheduling system is resource allocation which means that the actual job processing time is assumed to be a function of the amount of resource allocated such as financial budget, fuel, or additional manpower. Although two different resource functions including the linear one and the convex one are used to describe the combination between job processing time and the resource allocated, the linear function [15, 20] is not widely used as the convex one because of some marginal rules [21]. The convex function have the following specific form: $p_j^A(q_j) = (\frac{p_j}{q_j})^k$, for $j = 1, 2, \ldots, n$, where k is a positive constant, where $p_j^A(q_j)$ is the actual job processing time, q_j is the amount of resource that can be allocated to job J_j. Monma *et al.* [35] were among the first to propose this kind of model. Kaspi and Shabtay [21, 22], and Shabtay *et al.* [40] continued to discussed this convex function in scheduling problems. Shabtay and Steiner [41] provided very detailed survey on scheduling problems with resource allocation. Yin *et al.* [47] contributed to resource allocation related scheduling problem in the context of due-date assignment and batch delivery. There also exists other closely related work on scheduling problems considering resource allocation from different perspectives, such as [10, 12, 13, 48]. In addition, a very new interesting direction of scheduling is the combination of resource allocation and learning effect. Wang *et al.* [43] investigated single scheduling problems with learning effect and resource allocation. Zhu *et al.* [54] studied these problems in group technology environment. However, it is necessary to develop more general form of resource function to describe more exactly the effect of resource on scheduling process. Leyvand *et al.* [32] provided a general resource function as introduced above.

Setup consideration is always a very important issue in scheduling research. Three types of setup times including sequence independent, sequence dependent, and past-sequence-dependent (p-s-d) have been investigated. The first two types are classical ones that depend on the current job or both of the current and the last preceding jobs. The related work is Liu and Cheng [33], Asano and Ohta [4], et al. More details on scheduling problems with such kind of setup times are provided in reviews by Allahverdi *et al.* [2] and Allahverdi *et al.* [3]. The third one motivated by some phenomenons in high tech manufacturing is just involved by researchers in very recent years. Koulamas and Kyparisis [25] initiated this form of setup times which depends on all already scheduled jobs from the current batch. They proved that many scheduling problems with p-s-d setup times and different objective functions are still solvable in polynomial time. Then scheduling problems with this kind of setup times attracted much attention, and a very important stream is to integrate learning effect. Kuo and Yang [28], Wang [44], Wang *et al.* [46], Cheng *et al.* [14], *etc.* discussed respectively scheduling problems with learning effect and p-s-d setup times for different scheduling environment. Other related problems include Biskup and Herrmann [8], Zhu *et al.* [54], and Zhao and Tang [51], *etc.*

The scheduling process is the concurrent effects of machines, consumable resources, workers (learning effect), and other practical settings (setup cost), *etc.* Most existing research just studied one and very few analyzed two above issues. Zhu *et al.* [54] discussed scheduling problem with learning effect and consumable resources from the view of group technology the essence of which is to increase production efficiency by grouping various parts and products with similar designs and/or production process, not involved p-s-d setup times. Wang and Li [45] delt with scheduling problems with learning effect and psd setup times, however, they ignored consumable resources related decisions and cost. To the best of knowledge, there is still no published work on scheduling problems considering consumable resource allocation and the learning effect p-s-d setup times simultaneously, although it make great practical significance. For example, in the electronic industry, some electronic components exposed to certain electromagnetic fields wait for processing in the machines related area. This kind of exposure usually brings the adverse effect to each electronic component. Prior to a components processing, a setup operation, proportional to the degree of the adverse effect, is needed to restore them to a readiness status (remove the waiting time-induced adverse effect before its main processing). Therefore the setup time of an electronic component is dependent on the waiting time in electromagnetic fields that is just the sum of the processing times of electronic components processed before it. Such a setup time that is dependent on the past processed jobs is the past-sequence-dependent setup times. In addition, the processing and set times of each electronic component is not constant and is usually dependent on the operators expediences (learning effect) and consumable resources (such as energy/power). The operators' learning effect brings the reduction of setup time and job processing times. Larger amount of energy/power brings the change of production rate of machines (such as running at higher speeds reducing the processing times of electronic components) and consuming more energy/power (resulting in the increment of the total resources cost that is corresponding to the last item of the total cost function). In this paper we study scheduling problem integrating all above three issues. The objective is to determine the optimal job sequence and the optimal amount of resources allocated to each job jointly for minimizing the toal cost function yielded by the combination of the total completion time, the total absolute differences in completion times, and the total resource consumption.

The remainder of this paper is organized as follows. In Section 2, the problem is described in details. Optimal analysis for scheduling problems with resource allocation, learning effect, and p-s-d setup times is stated in Section 3. Several exension and special cases are represented in Section 4. Numerical example is presented in Section 5. Some conclusions and future work are discussed in the last section.

2. Problem description

The problem investigated can be formally stated as follows. A set of n independent jobs $(J_1, J_2, \ldots, J_n)$ are available for processing at time 0 on a single machine, and preemption is not allowed. $J_{[r]}$ denotes the job scheduled in the rth job position in a sequence $\pi = J_{[1]}, J_{[2]}, \ldots, J_{[r]}, \ldots, J_{[n]}$. Associated with each job J_j, there is a normal processing time p_j, a job-dependent learning effect factor a_j, and certain amount of resource allocated q_j. In this work, for each job J_j, the actual job processing time p^A_{jr} is a function of normal job processing time, resources allocated, and learning effect, which is an extension of the general resource function proposed by Leyvand *et al.* (2010). Supposing job J_i is scheduled in the rth position, the actual job processing time p^A_{jr} has the following form.

$$p^A_{jr} = p_j(q_j) r^{a_j}, \text{ for } j = 1, 2, \ldots, n;\ r = 1, 2, \ldots, n, \tag{2.1}$$

where $q_j \geq 0$ is the amount of resource allocated to job J_j, and $p_j(q_j)$ is the general convex function of q_j which reflects the effect of resources on job processing times.

As assumptions in former research such as Koulamas and Kyparisis [25], Kuo and Yang [28], the p-s-d setup time $s_{[r]}$ before processing each job J_j can be obtained as equations (2.2) if it is scheduled in rth position. The learning effect and resource related issues are integrated into the p-s-d setup time expressions together.

$$s_{[r]} = \varepsilon \sum_{l=1}^{r-1} p^A_{[l]} = \varepsilon \sum_{l=1}^{r-1} p_{[l]}(q_{[l]}) r^{a_{[l]}}, r = 2, \ldots, n, \text{ and } s_{[1]} = 0, \tag{2.2}$$

where ε is a nonnegative constant.

There exists many criterions in scheduling systems, such as makespan, total completion times, and total absolute differences in completion times usually regarded as a measurement of service level or of fairness towards consumers [38], *etc.* Most scheduling studies only focus on one of them. However, in practice the system usually needs to consider multiple criterions [5]. For example, the system concerns total completion times cost, service level to customers, and resources consumed. This paper discuss multiple objective combined by multi-criterions including the total completion time, total absolute differences in completion times, and the total resource consumption as follows:

$$Z(\pi, q) = \alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j, \tag{2.3}$$

where $C_j = C_j(\pi)$ is the completion time of job J_j for a schedule π, $TC = \sum_{j=1}^{n} C_j$ is the total completion times, $TADC = \sum_{i=1}^{n}\sum_{j=i}^{n} \mid C_i - C_j \mid$ is the total absolute differences in completion times, and $\Re_j$ is the unit resource cost, α_1, α_2, and α_3 are given non-negative parameters.

As the three-field notation of Graham *et al.* [17] for scheduling problem, the problem considered can be denoted as $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$, where RALE means "resource allocation and learning effect", and s_{psd} denotes "past-sequence-dependent setup times".

3. Problem analysis

In this section we analyze the optimal solutions for problem $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$. We provide some useful preliminary results first. Then some properties of it are provided, based on which an optimal algorithm is developed to find the optimal job sequence π^*, and resource allocation q^*.

To overcome disadvantages of previous two forms of resource allocation functions in accurately modelling the resource consumption, there are some studies such as Leyvand *et al.* [32] which use more general resource consumption functions. This type of function can vary not only between but also within jobs and only have to satisfy some not very restrictive properties. The adoption of it makes the problem under study more practical. Before using it, we present several properties of $p_j(q_j)$. If we denote the first derivative $dp_j(q_j)/dq_j$ and the second derivative $d^2 p_j(q_j)/d(q_j)^2$ as $p_j'(q_j)$ and $p_j''(q_j)$ respectively, the properties can be presented as follows:

$p_j'(q_j) \leq 0$, $p_j''(q_j) \geq 0$, for $q_j \in [q_j^{\min}, q_j^{\max}]$, where $q_j^{\min}$ and $q_j^{\max}$ are the lower and upper bound of the amount of resource allocated to job j. In addition, there exists only one point q_j such that $p_j'(q_j) = v$, for $v \in [p_j'(q_j^{\min}), p_j'(q_j^{\max})]$, which can be obtained in constant time.

Based on the expressions of TC and TADC in Koulamas and Kyparisis [25], we have the following new forms under the settings in this problem.

$$TC = \sum_{j=1}^{n}(n-j+1)\left(s_{[j]} + p_{[j]}^{A}\right) = \sum_{j=1}^{n}(n-j+1)\left(1+\varepsilon\frac{n-j}{2}\right)p_{[j]}(q_{[j]})j^{a_{[j]}}$$

$$TADC = \sum_{j=1}^{n}(j-1)(n-j+1)\left(s_{[j]} + p_{[j]}^{A}\right) = \sum_{j=1}^{n}\left((j-1)(n-j+1) + \varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)p_{[j]}(q_{[j]})j^{a_{[j]}}.$$

The first part of total cost function (2.3), $\alpha_1 TC + \alpha_2 TADC$ (set $\alpha_1 + \alpha_2 = 1$), can be given as

$$\begin{aligned}\alpha_1 TC + \alpha_2 TADC = {} & \alpha_1 \sum_{j=1}^{n}(n-j+1)\left(1+\varepsilon\frac{n-j}{2}\right)p_{[j]}(q_{[j]})j^{a_{[j]}} \\ & + \alpha_2 \sum_{j=1}^{n}\left((j-1)(n-j+1) + \varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)p_{[j]}\left(q_{[j]}\right)j^{a_{[j]}}.\end{aligned} \tag{3.1}$$

Considering equations (2.3) and (3.1), the total cost function $Z(\pi, q)$ has the following form:

$$
\begin{aligned}
Z(\pi, q) &= \alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j \\
&= \alpha_1 \sum_{j=1}^{n} (n-j+1) \left(1 + \varepsilon \frac{n-j}{2}\right) p_{[j]}(q_{[j]}) j^{a_{[j]}} \\
&\quad + \alpha_2 \sum_{j=1}^{n} \left((j-1)(n-j+1) + \varepsilon \sum_{l=j+1}^{n} (l-1)(n-l+1) \right) p_{[j]}(q_{[j]}) j^{a_{[j]}} \\
&\quad + \alpha_3 \sum_{j=1}^{n} \Re_{[j]} q_{[j]} \\
&= \sum_{j=1}^{n} \lambda_j p_{[j]}(q_{[j]}) j^{a_{[j]}} + \sum_{j=1}^{n} \alpha_3 \Re_{[j]} q_{[j]},
\end{aligned}
\tag{3.2}
$$

where

$$
\lambda_j = \alpha_1 (n-j+1) \left(1 + \varepsilon \frac{n-j}{2}\right) + \alpha_2 \left((j-1)(n-j+1) + \varepsilon \sum_{l=j+1}^{n} (l-1)(n-l+1) \right).
$$

We present the following lemma to show that the resource allocation is a function of certain job sequence, which could provide important information to predetermine optimal resource allocation for every job in certain sequence.

Lemma 3.1. *As a function of any given job sequence, the optimal resource allocation $q_j^*(\pi)$ for job J_j is*

$$
q_{[j]}^* = \begin{cases}
q_{[j]}^{\min}, & \text{if } p'_{[j]}(q_{[j]}^{\min}) \geq \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}}, \quad (3.3) \\
\hat{q}_{[j]}, & \text{if } p'_{[j]}(q_{[j]}^{\min}) < \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}} < p'_{[j]}(q_{[j]}^{\max}), \quad (3.4) \\
q_{[j]}^{\max}, & \text{if } p'_{[j]}(q_{[j]}^{\max}) \leq \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}}. \quad (3.5)
\end{cases}
$$

Proof. The equation (3.2) indicates that the cost function is the function of job sequences and resources allocated to each job. Therefore, for each given job sequence π, take the derivative of (3.2) with respect to $q_{[j]}^*$, let it be equal to 0, and solve it.

For $j = 1, 2, \ldots, n$,

$$
\begin{aligned}
\frac{dZ_{[j]}(\pi, q)}{dq_{[j]}} &= \frac{d\left(\lambda_j p_{[j]}(q_{[j]}) j^{a_{[j]}} + \alpha_3 \Re_{[j]} q_{[j]}\right)}{dq_{[j]}} \\
&= \lambda_j p'_{[j]}(q_{[j]}) j^{a_{[j]}} + \alpha_3 \Re_{[j]}.
\end{aligned}
$$

Let $\lambda_j p'_{[j]}(q_{[j]}) j^{a_{[j]}} + \alpha_3 \Re_{[j]} = 0$, we obtain $p'_{[j]}(q_{[j]}) = \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}}$, and the solution of this equation is $\hat{q}_{[j]}$.

If $p'_{[j]}(q_{[j]}^{\min}) \geq \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}} \geq 0$ which indicates $Z_{[j]}(\pi, q)$ is a non-decreasing function. For $q_{[j]} \in [q_{[j]}^{\min}, q_{[j]}^{\max}]$, the value $q_{[j]}^* = q_{[j]}^{\min}$ minimizes value of $Z_{[j]}(\pi, q)$.

If $p'_{[j]}(q_{[j]}^{\max}) \leq \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}} \leq 0$ which indicates and $Z_{[j]}(\pi, q)$ is a non-increasing function. For $q_{[j]} \in [q_{[j]}^{\min}, q_{[j]}^{\max}]$, the value $q_{[j]}^* = q_{[j]}^{\max}$ minimizes the value of $Z_{[j]}(\pi, q)$.

If $p'_{[j]}(q_{[j]}^{\min}) < \frac{-\alpha_3 \Re_{[j]}}{\lambda_j j^{a_{[j]}}} < p'_{[j]}(q_{[j]}^{\max})$, based on the properties of $p_j(q_j)$, it is concluded that $q_{[j]}^* = \hat{q}_{[j]}$ minimizes $Z_{[j]}(\pi, q)$. $\square$

Lemma 3.1 implies that the optimal resource allocation for a job is dependent only on its position in the job sequence and is independent on the jobs predecessors or successors in the sequence. In other words, the amount of resource allocated can be obtained according to (3.3)–(3.5) once the optimal job sequence is determined. Now we show the optimal job sequence decision method.

We define the binary variables x_{jr} such that $x_{jr} = 1$ if job J_j is scheduled in position r and otherwise $x_{jr} = 0$, for $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, n$. Let $\mathscr{H}_{jr}$ denote the cost incurred by job J_j scheduled in position r which can be obtained as follows:

$$\mathscr{H}_{jr} = \begin{cases} \lambda_r p_j(q_j^{\min}) r^{a_{[j]}} + \Re_j q_j^{\min}, & \text{if } p'_j(q_j^{\min}) \geq \frac{-\alpha_3 \Re_j}{\lambda_r r^{a_{[j]}}}, & (3.6)\\ \lambda_r p_j(\hat{q}_j) r^{a_{[j]}} + \Re_j \hat{q}_j, & \text{if } p'_j(q_j^{\min}) < \frac{-\alpha_3 \Re_j}{\lambda_r r^{a_{[j]}}} < p'_j(q_j^{\max}), & (3.7)\\ \lambda_r p_j(q_j^{\max}) r^{a_{[j]}} + \Re_j q_j^{\max}, & \text{if } p'_j(q_j^{\max}) \leq \frac{-\alpha_3 \Re_j}{\lambda_r r^{a_{[j]}}}. & (3.8) \end{cases}$$

Consequently, the optimal job sequence determining problem can be formulated as the following binary assignment problem:

$$\text{(P1)} \;\; \min \sum_{j=1}^{n} \sum_{r=1}^{n} \mathscr{H}_{jr} x_{jr}$$

subject to

$$\sum_{r=1}^{n} x_{jr} = 1, j = 1, 2, \ldots, n,$$
$$\sum_{j=1}^{n} x_{jr} = 1, r = 1, 2, \ldots, n,$$
$$x_{jr} = 0 \text{ or } 1, \quad j = 1, 2, \ldots, n; \quad r = 1, 2, \ldots, n.$$

The constraints ensure each job is scheduled in one position and each position is taken by only one job.

The preceding analysis for the $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ problem can be summarized with the following optimization algorithm.

Algorithm 1

Step 1. For $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, n$, calculate all values $\mathscr{H}_{jr}$ with equations (3.6)–(3.8).
Step 2. Solve the linear assignment problem (P1) to determine the optimal job sequence π^*.
Step 3. Calculate the optimal amount of resources allocated $q^*_{[j]}(\pi^*)$ using equation (3.3)–(3.5).
Step 4. Obtain the actual job processing times and p-s-d setup times with equations (2.1) and (2.2).

Theorem 3.2. *For the $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ problem, the optimal job sequence π^* and resource allocation $q^*(\pi^*)$ can be obtained in $O(n^3)$ time.*

Proof. Step 1 takes $O(n^2)$ time, Step 2 requires $O(n^3)$ [9], Step 3 takes $O(n)$ time, and Step 4 takes $O(n)$ time. Therefore, the overall complexity of Algorithm 1 is $O(\max(n^2, n^3, n))$, that is at most $O(n^3)$. □

4. Extension and special cases

In this section, we discuss some extension and special cases of the problem under study. The extension is the consideration of deteriorating rate-modifying activity (RMA) meaning that the actual duration of RMA may deteriorate over time during the scheduling of jobs and resource allocation. The special case is to discuss job-independent learning effect and specific resource consumption function. In this case, each job has the same learning factor and the resource allocation function has a specific form. We show that the extension and special case are still polynomially solvable while the complexity is different.

4.1. Extension

It is assumed that there exists a linear deteriorating RMA during the scheduling of jobs and resource allocation. In addition to the optimal job sequence π^* and resource allocation $q^*(\pi^*)$, the optimal position to schedule the deteriorating RMA is also an important decision to be made.

A rate-modifying activity affects the scheduling process much by changing the production rate of the equipments under consideration. Lee and Leon [29] first introduce this concept to the scheduling field, initiating a new type of scheduling problem in the context of realistic application. Then more excellent work focused on this interesting theme. He *et al.* [19] discussed a restricted rate-modifying activity. Lodree and Geiger [34] addressed scheduling problem with rate-modifying activity considering simple linear deterioration. Gordon *et al.* [18] analyzed scheduling problem with a rate-modifying activity in the context of common due date. Zhao *et al.* [52] integrated it to a two-parallel machines environment. Most papers above assumed that the duration of the rate-modifying activity is fixed. Considering this assumption may be invalid in many realistic life, several recent papers concentrated on deteriorating rate-modifying activity [26, 37, 49, 50].

When a deteriorating RMA, the position of which is i, if it is scheduled immediately after the completion of job i, is introduced, the actual processing time and p-s-d setup time can be represented as equations (4.1)–(4.4), where $i = 0, \ldots, n-1$, and $i = 0$ means the RMA is scheduled prior to the first job:

$$p_{[j]}^A = \begin{cases} p_{[j]}(q_{[j]})j^{a_{[j]}}, & j = 1, 2, \ldots, i, \quad (4.1) \\ \beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}, & j = i+1, i+2, \ldots, n, \quad (4.2) \end{cases}$$

$$s_{[j]} = \begin{cases} \varepsilon \sum_{l=1}^{j-1} p_{[l]}(q_{[l]})j^{a_{[l]}}, & j = 1, 2, \ldots, i, \quad (4.3) \\ \varepsilon \sum_{l=1}^{i} p_{[l]}(q_{[l]})j^{a_{[l]}} + \varepsilon \sum_{l=i+1}^{j-1} \beta_{[l]}p_{[l]}(q_{[l]})j^{a_{[l]}}, & j = i+1, \ldots, n, \quad (4.4) \end{cases}$$

where β_j is the improvement rate after the RMA for job J_j. Equations (4.1) denote the actual processing times of jobs scheduled before the RMA, while equations (4.2) mean the actual processing times of jobs scheduled after the RMA. As in Yang *et al.* [50], the RMA duration is $g(t) = \varphi + \gamma t$, where φ denotes the normal RMA time, γ denotes a deterioration factor of the RMA, and t denotes the starting time of the RMA. We denote this problem as $1|RALE, s_{psd}, DRM|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^n \Re_j q_j$, where DRM denotes "a deteriorating RMA".

4.1.1. Preliminary results

The introduction of the deteriorating RMA brings new characteristics to the actual job processing times, p-s-d setup times, the total completion time, and the total absolute difference in completion times. Therefore, the total completion time and the total absolute difference in completion times can be expressed as follows by induction:

$$\begin{aligned} TC = &\sum_{j=1}^{i} \left((n-j+1) + \varepsilon \frac{(n-j)(n-j+1)}{2} \right) p_{[j]}(q_{[j]})j^{a_{[j]}} \\ &+ \sum_{j=i+1}^{n} \left((n-j+1) + \varepsilon \frac{(n-j)(n-j+1)}{2} \right) \beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}} \\ &+ (n-i)\varphi + (n-i) \sum_{j=1}^{i} \left((i-j)\varepsilon + 1 \right) \gamma p_{[j]}(q_{[j]})j^{a_{[j]}} \end{aligned}$$

$$
\begin{aligned}
TADC = &\sum_{j=1}^{i}\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&+\sum_{j=i+1}^{n}\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&+i(n-i)\varphi+i(n-i)\sum_{j=1}^{i}\left((i-j)\varepsilon+1\right)\gamma p_{[j]}(q_{[j]})j^{a_{[j]}}.
\end{aligned}
$$

The inductions are referred to Appendix A.

4.1.2. An optimal solution

In this subsection, we show how to solve the scheduling problem $1|RALE, s_{psd}, DRM|\alpha_1 TC+\alpha_2 TADC+\alpha_3\sum_{j=1}^{n}\Re_j q_j$ involving resource allocation and deteriorating RMA decisions. In such a practical scheduling environment, the resource allocation, the position of deteriorating RMA, feasibility of schedules and associated execution costs are all taken into consideration concurrently.

Considering the above preliminary results, we have that

$$
\begin{aligned}
Z(\pi,q) &= \alpha_1 TC+\alpha_2 TADC+\alpha_3\sum_{j=1}^{n}\Re_j q_j\\
&=\alpha_1\Bigg[\sum_{j=1}^{i}\left((n-j+1)+\varepsilon\frac{(n-j)(n-j+1)}{2}\right)p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&\quad+\sum_{j=i+1}^{n}\left((n-j+1)+\varepsilon\frac{(n-j)(n-j+1)}{2}\right)\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&\quad+(n-i)\varphi+(n-i)\sum_{j=1}^{i}\left((i-j)\varepsilon+1\right)\gamma p_{[j]}(q_{[j]})j^{a_{[j]}}\Bigg]\\
&\quad+\alpha_2\Bigg[\sum_{j=1}^{i}\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&\quad+\sum_{j=i+1}^{n}\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}\\
&\quad+i(n-i)\varphi+i(n-i)\sum_{j=1}^{i}\left((i-j)\varepsilon+1\right)\gamma p_{[j]}(q_{[j]})j^{a_{[j]}}\Bigg]+\alpha_3\sum_{j=1}^{n}\Re_{[j]}q_{[j]}.\\
&=\sum_{j=1}^{i}\phi_j p_{[j]}(q_{[j]})j^{a_{[j]}}+\sum_{j=i+1}^{n}\delta_j\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}+(\alpha_1(n-i)+\alpha_2 i(n-i))\varphi\\
&\quad+\sum_{j=1}^{n}\alpha_3\Re_{[j]}q_{[j]},
\end{aligned}
\tag{4.5}
$$

where

$$
\phi_j=\alpha_1\left((n-j+1)+\varepsilon\frac{(n-j)(n-j+1)}{2}+(n-i)\left((i-j)\varepsilon+1\right)\gamma\right)
$$

$$+\alpha_2\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)+i(n-i)\left((i-j)\varepsilon+1\right)\gamma\right),$$

$$\delta_j=\alpha_1\left((n-j+1)+\varepsilon\frac{(n-j)(n-j+1)}{2}\right)$$
$$+\alpha_2\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right).$$

Like Lemma 3.1, the following lemma shows that the resource allocation is a function of certain job sequence. The difference is that the resource allocation is also dependent on the position of the RMA.

Lemma 4.1. *As a function of any given job sequence, the optimal resource allocation $q_j^*(\pi)$ for job J_j is*

$$q_{[j]}^*=\begin{cases} q_{[j]}^{\min}, & \text{if } p'_{[j]}(q_{[j]}^{\min})\geq\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}, \text{for } \ j=1,2,\ldots,i, & (4.6)\\ q_{[j]}^{\min}, & \text{if } p'_{[j]}(q_{[j]}^{\min})\geq\frac{-\alpha_3\Re_{[j]}}{\delta_j j^{a_{[j]}}\beta_{[j]}}, \text{for } \ j=i+1,\ldots,n, & (4.7)\\ \hat{q}_{[j]}, & \text{if } p'_{[j]}(q_{[j]}^{\min})<\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}<p'_{[j]}(q_{[j]}^{\max}), \text{for } \ j=1,2,\ldots,i, & (4.8)\\ \hat{q}_{[j]}, & \text{if } p'_{[j]}(q_{[j]}^{\min})<\frac{-\alpha_3\Re_{[j]}}{\delta_j j^{a_{[j]}}\beta_{[j]}}<p'_{[j]}(q_{[j]}^{\max}), \text{for } \ j=i+1,\ldots,n, & (4.9)\\ q_{[j]}^{\max}, & \text{if } p'_{[j]}(q_{[j]}^{\max})\leq\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}, \text{for } \ j=1,2,\ldots,i, & (4.10)\\ q_{[j]}^{\max}, & \text{if } p'_{[j]}(q_{[j]}^{\max})\leq\frac{-\alpha_3\Re_{[j]}}{\delta_j j^{a_{[j]}}\beta_{[j]}}, \text{for } \ j=i+1,\ldots,n. & (4.11)\end{cases}$$

Proof. The equation (4.5) indicates that the cost function is the function of job sequences and resources allocated to each job. Therefore, for each given job sequence π, take the derivative of (4.5) with respect to $q_{[j]}^*$, let it be equal to 0, and solve it.

For $j=1,2,\ldots,i$,

$$\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}}=\frac{d\left(\phi_j p_{[j]}(q_{[j]})j^{a_{[j]}}+\alpha_3\Re_{[j]}q_{[j]}\right)}{dq_{[j]}}=\phi_j p'_{[j]}(q_{[j]})j^{a_{[j]}}+\alpha_3\Re_{[j]}=0.$$

We have $p'_{[j]}(q_{[j]})=\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}$ and assume the solution of this equation is $\hat{q}_{[j]}$.

If $p'_{[j]}(q_{[j]}^{\min})\geq\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}}\geq 0$ which indicates that $Z_{[j]}(\pi,q)$ is a non-decreasing function. For $q_{[j]}\in[q_{[j]}^{\min},q_{[j]}^{\max}]$, the value $q_{[j]}^*=q_{[j]}^{\min}$ minimizes the value of $Z_{[j]}(\pi,q)$.

If $p'_{[j]}(q_{[j]}^{\max})\leq\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}}\leq 0$ which indicates $Z_{[j]}(\pi,q)$ is a non-increasing function. For $q_{[j]}\in[q_{[j]}^{\min},q_{[j]}^{\max}]$, the value $q_{[j]}^*=q_{[j]}^{\max}$ minimizes the value of $Z_{[j]}(\pi,q)$.

If $p'_{[j]}(q_{[j]}^{\min})<\frac{-\alpha_3\Re_{[j]}}{\phi_j j^{a_{[j]}}}<p'_{[j]}(q_{[j]}^{\max})$, based on the properties of $p_j(q_j)$, it is concluded that $q_{[j]}^*=\hat{q}_{[j]}$ is the optimal value to minimize the value of $Z_{[j]}(\pi,q)$.

For $j=i+1,i+2,\ldots,n$,

$$\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}}=\frac{d\left(\delta_j\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}+\alpha_3\Re_{[j]}q_{[j]}\right)}{dq_{[j]}}=\delta_j\beta_{[j]}p'_{[j]}(q_{[j]})j^{a_{[j]}}+\alpha_3\Re_{[j]}=0.$$

If $p'_{[j]}(q_{[j]}^{\min})\geq\frac{-\alpha_3\Re_{[j]}}{\delta_j j^{a_{[j]}}\beta_{[j]}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}}\geq 0$ which indicates $Z_{[j]}(\pi,q)$ is a non-decreasing function. For $q_{[j]}\in[q_{[j]}^{\min},q_{[j]}^{\max}]$, the value $q_{[j]}^*=q_{[j]}^{\min}$ minimizes $Z_{[j]}(\pi,q)$.

If $p'_{[j]}(q_{[j]}^{\max}) \leq \frac{-\alpha_3 \Re_{[j]}}{\delta_j j^{a[j]} \beta_{[j]}}$, we obtain $\frac{dZ_{[j]}(\pi,q)}{dq_{[j]}} \leq 0$ which indicates $Z_{[j]}(\pi, q)$ is a non-increasing function. For $q_{[j]} \in [q_{[j]}^{\min}, q_{[j]}^{\max}]$, the value $q_{[j]}^* = q_{[j]}^{\max}$ minimizes $Z_{[j]}(\pi, q)$.

If $p'_{[j]}(q_{[j]}^{\min}) < \frac{-\alpha_3 \Re_{[j]}}{\delta_j j^{a[j]} \beta_{[j]}} < p'_{[j]}(q_{[j]}^{\max})$, based on the properties of $p_j(q_j)$, it is concluded that $q_{[j]}^* = \hat{q}_{[j]}$ minimizes $Z_{[j]}(\pi, q)$. □

Similarly, Lemma 4.1 implies that the amount of resource allocated to a job can be obtained according to (4.6)–(4.11) once the optimal job sequence and the position of the RMA are determined. The optimal job sequence and the position of the RMA decision methods are presented as follows.

We define the binary variables y_{jr} such that $y_{jr} = 1$ if job J_j is scheduled in position r and otherwise $y_{jr} = 0$, for $j = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, n$. Let $\mathscr{Q}_{jr}$ denote the cost incurred by job J_j scheduled in position r.

$$
\mathscr{Q}_{jr} = \begin{cases}
\phi_r p_j(q_j^{\min}) r^{a[j]} + \Re_j q_j^{\min}, & \text{if } p'_j(q_j^{\min}) \geq \frac{-\alpha_3 \Re_{[j]}}{\phi_j r^{a[j]}}, \text{ for } r = 1, 2, \ldots, i, \quad (4.12)\\
\phi_r p_j(\hat{q}_j) r^{a[j]} + \Re_j \hat{q}_j, & \text{if } p'_j(q_j^{\min}) < \frac{-\alpha_3 \Re_{[j]}}{\phi_j r^{a[j]}} < p'_j(q_j^{\max}), \text{ for } r = 1, \ldots, i, \quad (4.13)\\
\phi_r p_j(q_j^{\max}) r^{a[j]} + \Re_j q_j^{\max}, & \text{if } p'_j(q_j^{\max}) \leq \frac{-\alpha_3 \Re_{[j]}}{\phi_j r^{a[j]}}, \text{ for } r = 1, 2, \ldots, i, \quad (4.14)\\
\delta_r p_j(q_j^{\min}) r^{a[j]} + \Re_j q_j^{\min}, & \text{if } p'_j(q_j^{\min}) \geq \frac{-\alpha_3 \Re_{[j]}}{\delta_j r^{a[j]} \beta_{[j]}}, \text{ for } r = i+1, i+2, \ldots, n, \quad (4.15)\\
\delta_r p_j(\hat{q}_j) r^{a[j]} + \Re_j \hat{q}_j, & \text{if } p'_j(q_j^{\min}) < \frac{-\alpha_3 \Re_{[j]}}{\delta_j r^{a[j]} \beta_{[j]}} < p'_j(q_j^{\max}), \text{ for } r = i+1, i+2, \ldots, n, \quad (4.16)\\
\delta_r p_j(q_j^{\max}) r^{a[j]} + \Re_j q_j^{\max}, & \text{if } p'_j(q_j^{\max}) \leq \frac{-\alpha_3 \Re_{[j]}}{\delta_j r^{a[j]} \beta_{[j]}}, \text{ for } r = i+1, i+2, \ldots, n. \quad (4.17)
\end{cases}
$$

The optimal job sequence and the position of the RMA determining problem can be formulated as the following linear programming problem.

$$
\text{(P2)} \ \min \sum_{j=1}^{n} \sum_{r=1}^{n} \mathscr{Q}_{jr} y_{jr} + (\alpha_1(n-i) + \alpha_2 i(n-i))\varphi
$$

subject to

$$
\begin{gathered}
\sum_{r=1}^{n} y_{jr} = 1, j = 1, 2, \ldots, n,\\
\sum_{j=1}^{n} y_{jr} = 1, r = 1, 2, \ldots, n,\\
y_{jr} = 0 \text{ or } 1, j = 1, 2, \ldots, n; r = 1, 2, \ldots, n.
\end{gathered}
$$

The constraints guarantee each job is scheduled in only one position and each position is taken by only one job.

The objective function ($P2$) comprises job sequence and deteriorating RMA related costs. Once the position of the deteriorating RMA is provided, the last two parts are constants and the job sequence determining problem is converted to solve the following binary assignment problem.

$$
\text{(P3)} \ \min \sum_{j=1}^{n} \sum_{r=1}^{n} \mathscr{Q}_{jr} y_{jr}
$$

subject to

$$
\sum_{r=1}^{n} y_{jr} = 1, j = 1, 2, \ldots, n,
$$

$$\sum_{j=1}^{n} y_{jr} = 1, r = 1, 2, \dots, n,$$
$$y_{jr} = 0 \text{ or } 1, j = 1, 2, \dots, n; r = 1, 2, \dots, n.$$

For the problem $1|RALE, s_{psd}, DRM|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$, we propose he following algorithm to summarize the preceding analysis.

Algorithm 2

Step 1. Set $i = 0$.
Step 2. Calculate each value $\mathscr{Q}_{jr}$ with equations (4.12)–(4.17), for $j = 1, 2, \dots, n$ and $r = 1, 2, \dots, n$.
Step 3. Determine a local optimal job sequence π_l^* and record the total cost $Z(\pi_l^*)$ by solving the corresponding linear assignment problem (P3).
Step 4. Set $i = i + 1$. If $i < n$, then go to Step 2. Otherwise go to Step 5.
Step 5. Order the total cost of all local optimal job sequence π_l^*, and the one with the minimal total cost Z^* is denoted as the global optimal job sequence (π^*).
Step 6. Calculate the optimal amount of resources allocated $q_{[j]}^*(\pi^*)$ by equations (4.6)–(4.11).
Step 7. Calculate the actual job processing times and p-s-d setup times with equations (4.1)–(4.4).

Theorem 4.2. *For the* $1|RALE, s_{psd}, DRM|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ *problem, the optimal job sequence* π^* *and resource allocation* $q^*(\pi^*)$ *can be obtained in* $O(n^4)$ *time.*

Proof. The position of the deteriorating RMA may be $1, 2, \dots, n$, so the problem $1|RALE, s_{psd}, DRM|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ is solvable in $O(n^4)$ time. □

4.2. Special cases

In this subsection, we discuss special cases of $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ and $1|RALE, s_{psd}, DRM|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ with the consideration of job independent learning effect and specific convex function (JileConv). In this case, each job has the same learning factor a, and the specific form of convex function is $p_j^A(q_j) = (\frac{p_j}{q_j})^k$, for $j = 1, 2, \dots, n$, as introduced in Section 1. Therefore, the actual processing time influenced by learning effect and resource allocation can be expressed as

$$p_{jr}^A = \left(\frac{p_j}{q_j}\right)^k r^a, \text{ for } j = 1, 2, \dots, n; r = 1, 2, \dots, n. \tag{4.18}$$

Based on the above function and previous results, the total cost function of the problem $1|RALE, s_{psd}$, JileConv$|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ is

$$\begin{aligned} Z(\pi, q) &= \alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j \\ &= \sum_{j=1}^{n} \lambda_j \left(\frac{p_j}{q_j}\right)^k r^a + \sum_{j=1}^{n} \alpha_3 \Re_{[j]} q_{[j]}, \end{aligned} \tag{4.19}$$

The complexity is $O(nlogn)$, and the proof is similar to Wang *et al.* [43].

The total cost function of the problem $1|RALE, s_{psd}, DRM$, JileConv$|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ is

$$
\begin{aligned}
Z(\pi, q) &= \alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j \\
&= \sum_{j=1}^{i} \phi_j \left(\frac{p_j}{q_j}\right)^k r^a + \sum_{j=i+1}^{n} \delta_j \beta_{[j]} \left(\frac{p_j}{q_j}\right)^k r^a + (\alpha_1(n-i) + \alpha_2 i(n-i))\varphi \\
&\quad + \sum_{j=1}^{n} \alpha_3 \Re_{[j]} q_{[j]}.
\end{aligned}
\tag{4.20}
$$

The complexity is $O(n^2 \log n)$, and the proof is similar to Zhu *et al.* [55].

5. Numerical example

In this section, we provide two numerical examples to illustrate the problem and algorithms. To present the examples more clearly, a specific form of resource allocation function, $(\frac{p_j}{q_j})^k, k > 0$, is used in the numerical experiment. The analysis of other forms of resource allocation function is simillar.

Example 1. It is assumed a set of 10 jobs are available for processing. The normal job processing times, learning effect factors, unit resource costs, and the lower and upper bound of the amount of resource allocated to each job are showed in Table 2. Other parameters are set as $\alpha_1 = 3$, $\alpha_2 = 2$, $\alpha_3 = 1$, $k = 2$, $\varepsilon = 1.2$.

According to Algorithm 1, the $1|RALE, s_{psd}|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$ problem can be solved as follows:

First, the value $\mathscr{H}_{jr}$ is obtained with equations (3.6)–(3.8) for $j = 1, 2, \ldots, 10$ and $r = 1, 2, \ldots, 10$.

	rj	1	2	3	4	5	6	7	8	9	10
	1	338.60	320.48	114.71	281.58	137.53	209.91	100.53	198.74	541.48	252.00
	2	305.24	292.31	104.12	268.93	122.55	191.20	88.56	166.43	471.49	220.41
	3	280.84	271.24	96.19	255.93	112.00	178.01	80.38	145.62	425.10	198.61
	4	258.33	252.24	88.73	241.14	102.62	165.16	73.24	128.66	385.45	179.99
$\mathscr{H}_{jr} =$	5	235.62	232.65	81.11	224.06	94.57	151.65	66.30	113.84	347.66	163.18
	6	211.69	210.93	73.01	204.39	87.77	137.00	59.21	100.74	309.51	147.33
	7	185.81	186.58	64.18	181.72	82.06	120.81	51.70	88.57	269.59	129.98
	8	157.19	158.89	54.37	155.45	77.41	102.61	43.54	75.09	226.55	110.45
	9	124.74	126.68	43.14	124.42	73.87	81.61	34.38	59.63	178.51	87.89
	10	98.41	87.13	30.05	91.68	71.51	56.27	24.77	40.87	121.49	60.34

Secondly, by solving the assignment problem (P1), the optimal job sequence π^* can be determined, which is $\pi^* = (J_3, J_7, J_5, J_6, J_8, J_{10}, J_4, J_2, J_1, J_9)$.

Thirdly, the optimal resource allocation is as follows: $q_3^* = 50.98, q_7^* = 14.76, q_5^* = 23.33, q_6^* = 44.04, q_8^* = 65.00, q_{10}^* = 28.06, q_4^* = 40.38, q_2^* = 52.96, q_1^* = 20, q_9^* = 16.20$.

Finally, the actual job processing times and p-s-d setup times can be calculated, and the optimal total cost is 1328.44.

Example 2. $\beta_1 = 0.49$, $\beta_2 = 0.35$, $\beta_3 = 0.34$, $\beta_4 = 0.68$, $\beta_5 = 0.44$, $\beta_6 = 0.13$, $\beta_7 = 0.14$, $\beta_8 = 0.56$, $\beta_9 = 0.27$, $\beta_{10} = 0.21$, $\varphi = 7$, $\gamma = 0.015$. Other input data is the same as in Example 1.

According to Algorithm 2, for each possible position of RMA, $i = 0, \ldots, 9$, we obtain the local optimal job sequence π_i^* and the corresponding total cost $Z(\pi_i^*)$ by solving the linear assignment problem (P3). The results are showed in Table 3.

TABLE 1. Complexity of different problems.

Problems	Complexity
$1\|RALE, s_{psd}\|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$	$O(n^3)$
$1\|RA, s_{psd}, DRM\|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$	$O(n^4)$
$1\|RALE, s_{psd}, JileConv\|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$	$O(n \log n)$
$1\|RALE, s_{psd}, DRM, JileConv\|\alpha_1 TC + \alpha_2 TADC + \alpha_3 \sum_{j=1}^{n} \Re_j q_j$	$O(n^2 \log n)$

TABLE 2. Input data associated with each job.

j	1	2	3	4	5	6	7	8	9	10
p_j	23	43	13	25	8	19	4	31	40	15
a_j	−0.35	−0.2	−0.32	−0.1	−0.4	−0.26	−0.45	−0.3	−0.5	−0.25
R_j	4.3	2	1.5	3	3.2	2.5	4	1	5	3.5
$q_j^{\min}$	20	5	15	25	22	16	5	15	18	7
$q_j^{\max}$	55	85	60	80	40	50	35	65	90	30

TABLE 3. local optimal job sequence and corresponding total cost for all possible positions of RMA.

Position of RMA (i)	Local optimal job sequence (π_i^*)	Total cost $Z(\pi_i^*)$
Not scheduled	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_4, J_2, J_1, J_9)$	1328.44
0	$(J_7, J_3, J_5, J_6, J_8, J_{10}, J_2, J_1, J_9, J_4)$	1237.00
1	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_2, J_1, J_9, J_4)$	1287.94
2	$(J_3, J_5, J_7, J_6, J_8, J_{10}, J_2, J_1, J_9, J_4)$	1394.51
3	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_2, J_1, J_9, J_4)$	1479.98
4	$(J_3, J_7, J_5, J_8, J_6, J_{10}, J_2, J_1, J_9, J_4)$	1531.14
5	$(J_3, J_7, J_5, J_8, J_4, J_6, J_{10}, J_1, J_9, J_2)$	1574.64
6	$(J_3, J_7, J_5, J_8, J_{10}, J_4, J_6, J_1, J_9, J_2)$	1598.32
7	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_4, J_1, J_9, J_2)$	1586.31
8	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_4, J_1, J_9, J_2)$	1528.68
9	$(J_3, J_7, J_5, J_6, J_8, J_{10}, J_4, J_2, J_1, J_9)$	1455.00

Thus, the global optimal job sequnece is $(J_7, J_3, J_5, J_6, J_8, J_{10}, J_2, J_1, J_9, J_4)$ with the minimal total cost. The optimal resource allocation is as follows: $q_7^* = 8.70, q_3^* = 32.30, q_5^* = 22.00, q_6^* = 22.31, q_8^* = 61.37, q_{10}^* = 16.68, q_2^* = 43.83, q_1^* = 20, q_9^* = 18, q_4^* = 25$, and the optimal total cost is 1237.00.

6. CONCLUSIONS

Classical scheduling models have mainly focused on machines or materials. In practical scheduling environment, it does not suffice since the scheduling process is usually the concurrent effects of machines, consumable resources, workers (learning effect), and other practical settings (setup cost), etc and may bring great challenges to traditional scheduling results. Thus, this work fills the gap by studying a single-machine scheduling problem with learning effect and resource allocation in the context of past-sequence-dependent setup times. In such problem, the actual job processing times are dependent on learning effect and the amount of resource allocated, and the setup times are proportional to the length of the already processed jobs. The objective is to determine the optimal job sequence and the optimal amount of resources allocated to each job jointly which minimize a combination of the total completion time, the total absolute differences in completion times, and the total resource consumption. We show that the optimal job sequence and resource allocation can be obtained in $O(n^3)$ time. Then we discuss some extension and special cases of this problem including a deteriorating RMA, job-independent learning effect and a specific convex resource function. It is showed that the special cases are solvable in $O(n^4)$, $O(n \log n)$, and $O(n^2 \log n)$ time respectively, as showed in Table 1.

Most studies on scheduling problems including this paper only involve one agent which has a set of jobs to be processed on a processor. However, scheduling problems consists of multiple agents who compete on the use of the same machine have many practical applications [1,31]. In multiple agents compete systems, different agents interact to perform their respective tasks, negotiating among each other for the usage of common resources over time. Therefore, competing agents related environment will be our future research directions.

7. Appendix A

The p-s-d setup time and actual processing time for job J_j, $j = 1, 2, \ldots, i, i+1, \ldots$, are given by

$$\begin{aligned}
P'_i &= s_{[i]} + p^A_{[i]} = \varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + p^A_{[i]},\\
P'_{[n]} &= s_{[n]} + p^A_{[n]} = \varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + \varepsilon p^A_{[i]} + \varepsilon \beta_{[i+1]} p^A_{[i+1]} + \varepsilon \beta_{[i+2]} p^A_{[i+2]}\\
&\quad + \varepsilon \beta_{[i+3]} p^A_{[i+3]} + \ldots + \varepsilon \beta_{[n-1]} p^A_{[n-1]} + \beta_{[n]} p^A_{[n]}.
\end{aligned}$$

The completion time of each job ($j = 1, 2, \ldots, i, i+1, \ldots, n$) can be denoted as

$$\begin{aligned}
C_{[i]} &= p^A_{[1]} + \left(\varepsilon p^A_{[1]} + p^A_{[2]}\right) + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + p^A_{[3]}\right) + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + p^A_{[4]}\right) + \ldots\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + p^A_{[i]}\right),
\end{aligned}$$

...

$$\begin{aligned}
C_{[n]} &= p^A_{[1]} + \left(\varepsilon p^A_{[1]} + p^A_{[2]}\right) + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + p^A_{[3]}\right) + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + p^A_{[4]}\right) + \ldots\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + p^A_{[i]}\right) + \varphi + \gamma \left(p^A_{[1]} + \left(\varepsilon p^A_{[1]} + p^A_{[2]}\right) + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + p^A_{[3]}\right)\right.\\
&\quad \left. + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + p^A_{[4]}\right) + \ldots + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + p^A_{[i]}\right)\right)\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + \varepsilon p^A_{[i]} + \beta_{[i+1]} p^A_{[i+1]}\right)\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + \varepsilon p^A_{[i]} + \varepsilon \beta_{[i+1]} p^A_{[i+1]} + \beta_{[i+2]} p^A_{[i+2]}\right)\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + \varepsilon p^A_{[i]} + \varepsilon \beta_{[i+1]} p^A_{i+1} + \varepsilon \beta_{[i+2]} p^A_{[i+2]} + \beta_{[i+3]} p^A_{[i+3]}\right) + \ldots\\
&\quad + \left(\varepsilon p^A_{[1]} + \varepsilon p^A_{[2]} + \varepsilon p^A_{[3]} + \ldots + \varepsilon p^A_{[i-1]} + \varepsilon p^A_{[i]} + \varepsilon \beta_{[i+1]} p^A_{[i+1]} + \varepsilon \beta_{[i+2]} p^A_{[i+2]} + \varepsilon \beta_{[i+3]} p^A_{[i+3]} + \ldots\right.\\
&\quad \left. + \varepsilon \beta_{[n-1]} p^A_{[n-1]} + \beta_{[n]} p^A_{[n]}\right).
\end{aligned}$$

The total completion time is obtained based on the induction

$$\begin{aligned}
TC &= \sum_{j=1}^{i} \left((n-j+1) + \varepsilon \frac{(n-j)(n-j+1)}{2}\right) p_{[j]}(q_{[j]}) j^{a_{[j]}}\\
&\quad + \sum_{j=i+1}^{n} \left((n-j+1) + \varepsilon \frac{(n-j)(n-j+1)}{2}\right) \beta_{[j]} p_{[j]}(q_{[j]}) j^{a_{[j]}}\\
&\quad + (n-i)\varphi + (n-i) \sum_{j=1}^{i} \left((i-j)\varepsilon + 1\right) \gamma p_{[j]}(q_{[j]}) j^{a_{[j]}}.
\end{aligned}$$

The total absolute difference in completion times is obtained based on the following induction

$$TADC = \sum_{j=1}^{i} \left((j-1)(n-j+1) + \varepsilon \sum_{l=j+1}^{n} (l-1)(n-l+1)\right) p_{[j]}(q_{[j]}) j^{a_{[j]}}$$

$$+\sum_{j=i+1}^{n}\left((j-1)(n-j+1)+\varepsilon\sum_{l=j+1}^{n}(l-1)(n-l+1)\right)\beta_{[j]}p_{[j]}(q_{[j]})j^{a_{[j]}}$$
$$+i(n-i)\varphi+i(n-i)\sum_{j=1}^{i}\left((i-j)\varepsilon+1\right)\gamma p_{[j]}(q_{[j]})j^{a_{[j]}}.$$

Acknowledgements. This work is partially supported by NSF of China under Grant No. 71201085.

References

[1] A. Agnetis, P.B. Mirchandani, D. Pacciarelli and A. Pacifici, Scheduling problems with two competing agents. *Oper. Res.* **52** (2004) 229–242.

[2] A. Allahverdi, J.N.D. Gupta and T. Aldowaisan, A review of scheduling research involving setup considerations. *OMEGA Int. J. Manage. Sci.* **27** (1999) 219–239.

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

[4] M. Asano and H. Ohta, Scheduling with shutdowns and sequence dependent set-up times. *Int. J. Prod. Res.* **37** (1999) 1661–1676.

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

[6] D. Biskup, Single-machine scheduling with learning considerations. *Eur. J. Oper. Res.* **115** (1999) 173–178.

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

[8] 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.

[9] P. Brucker, *Scheduling Algorithms*, 3rd edition. Springer, Berlin (2001).

[10] T.C.E. Cheng and A. Janiak, Resource optimal control in some single-machine scheduling problems. *IEEE Trans. Autom. Control* **39** (1994) 1243–1246.

[11] T.C.E. Cheng and G.Q. Wang, Single machine scheduling with learning effect considerations. *Ann. Oper. Res.* **98** (2000) 273–290.

[12] T.C.E. Cheng, Z.L. Chen and C.L. Li, Single-machine scheduling with trade-off between number of tardy jobs and resource allocation. *Oper. Res. Lett.* **19** (1996) 237–242.

[13] T.C.E. Cheng, A. Janiak, M.Y. Kovalyov, Single machine batch scheduling with resource dependent setup and processing times. *Eur. J. Oper. Res.* **135** (2001) 177–183.

[14] 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.

[15] R.L. Daniels and R.K. Sarin, Single machine scheduling with controllable processing times and number of jobs tardy. *Oper. Res.* **37** (1989) 981–984.

[16] C. de Snoo, *Coordination in Planning and Scheduling-An Organizational and Behavioral Perspective.* University of Groningen (2011). ISBN: 978-90-367-4502-4.

[17] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.R. KAN, Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* **5** (1979) 287–326.

[18] V.S. Gordon and A.A. Tarasevich, A note: Common due date assignment for a single machine scheduling with the rate-modifying activity. *Comput. Oper. Res.* **36** (2009) 325–328.

[19] Y. He, M. Ji and T.C.E. Cheng, Single machine scheduling with a restricted rate-modifying activity. *Naval Res. Logistics* **52** (2005) 361–369.

[20] A. Janiak and M.Y. Kovalyov, Single machine scheduling subject to deadlines and resource dependent processing times. *Eur. J. Oper. Res.* **94** (1996) 284–91.

[21] M. Kaspi and D. Shabtay, Convex resource allocation for minimizing the makespan in a single machine with job release dates. *Comput. Oper. Res.* **31** (2004) 1481–1489.

[22] M. Kaspi and D. Shabtay, A bicriterion approach to time/cost trade-offs in scheduling with convex resource-dependent job processing times and release dates. *Comput. Oper. Res.* **33** (2006) 3015–3033.

[23] C. Koulamas, A Note on single-machine scheduling with job-dependent learning effects. *Eur. J. Oper. Res.* **207** (2010) 1142–1143.

[24] C. Koulamas and G. Kyparisis, Single-machine and two-machine flowshop scheduling with general learning functions. *Eur. J. Oper. Res.* **178** (2007) 402–407.

[25] C. Koulamas and G.J. Kyparisis, Single-machine scheduling problems with past-sequence-dependent setup times. *Eur. J. Oper. Res.* **187** (2008) 68–72.

[26] M.A. Kubzin and V.A. Strusevich, Planning machine maintenance in two-machine shop scheduling. *Oper. Res.* **54** (2006) 789–800.

[27] W.-H. Kuo and D.-L. Yang, Minimizing the total completion time in a singlemachine scheduling problem with a time-dependent learning effect. *Eur. J. Oper. Res.* **174** (2006) 1184–1190.

[28] W.-H. Kuo and D.-L. Yang, Single-machine scheduling with past-sequence-dependent setup and learning effects. *Inf. Process. Lett.* **102** (2007) 22–26.

[29] C.-L. Lee and V.-J. Leon, Machine scheduling with a ratemodifying activity. *Eur. J. Oper. Res.* **129** (2001) 119–128.

[30] W.-C. Lee and C.-C. Wu, Minimizing total completion time in a two-machine flowshop with a learning effect. *Int. J. Prod. Econ.* **88** (2004) 85–93.

[31] J.Y.-T. Leung, M. Pinedo and G. Wan, Competitive two-agent scheduling and its applications. *Oper. Res.* **58** (2010) 458–469.

[32] Y. Leyvand, D. Shabtay and G. Steiner, A unified approach for scheduling with convex resource consumption functions using positional penalties. *Eur. J. Oper. Res.* **206** (2010) 301–312.

[33] Z. Liu and T.C.E. Cheng, Minimizing total completion time subject to job release dates and preemption penalties. *J. Sched.* **7** (2004) 313–327.

[34] E.J. Lodree Jr. and C.D. Geiger, A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration. *Eur. J. Oper. Res.* **201** (2010) 644–648.

[35] C.L. Monma, A. Schrijver, M.J. Todd and V.K. Wei, Convex resource allocation problems on directed acyclic graphs: Duality, complexity, special cases and extensions. *Math. Oper. Res.* **15** (1990) 736–748.

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

[37] G. Mosheiov and J.B. Sidney, Scheduling a deteriorating maintenance activity on a single machine. *J. Oper. Res. Soc.* **61** (2009) 882–887.

[38] D. Oron, Single machine scheduling with simple linear deterioration to minimize total absolute deviation of completion times. *Comput. Oper. Res.* **35** (2008) 2071–2078.

[39] M.L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, NY (2008).

[40] D. Shabtay, M. Kaspi and G. Steiner, The no-wait two-machine flow-shop scheduling problem with convex resource-dependent processing times. *IIE Trans.* **39** (2007) 539–57.

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

[42] N. Slack, S. Chambers and R. Johnston, *Operations Management*. Pearson Education, Harlow (2007).

[43] D. Wang, M.-Z. Wang and J.-B. Wang, Single-machine scheduling with learning effect and resource-dependent processing times. *Comput. Ind. Eng.* **58** (2010) 458–462.

[44] J.-B. Wang, Single-machine scheduling with past-sequence dependent setup times and time-dependent learning effect. *Comput. Ind. Eng.* **55** (2008) 584–591.

[45] 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.

[46] 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. Appl.* **57** (2009) 9–16.

[47] Y. Yin, T.C.E. Cheng, C.-C. Wu and S.-R. Cheng, Single-machine common due-date scheduling with batch delivery costs and resource-dependent processing times. *Int. J. Prod. Res.* **51** (2013) 5083–5099.

[48] J.J. Yuan, T.C.E. Cheng and C.T. Ng, NP-hardness of the single-variable-resource scheduling problem to minimize the total weighted completion time. *Eur. J. Oper. Res.* **178** (2007) 631–633.

[49] S.-J. Yang and D.-L. Yang, Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities. *Comput. Math. Appl.* **60** (2010) 2161–2169.

[50] S.-J. Yang, D.-L. Yang and T.C.E. Cheng, Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance. *Comput. Oper. Res.* **37** (2010) 1510–1514.

[51] C. Zhao and H. Tang, Single machine scheduling with past-sequence-dependent setup times and deteriorating jobs. *Comput. Ind. Eng.* **59** (2010) 663–666.

[52] C. Zhao, H. Tang and C. Cheng, Two-parallel machines scheduling with rate-modifying activities to minimize total completion time. *Eur. J. Oper. Res.* **198** (2009) 354–357.

[54] Z. Zhu, F. Chu, L. Sun and M. Liu, Scheduling with resource allocation and past-sequence-dependent setup times including maintenance. *International Conference on Networking, Sensing and Control* (2011) 383–387.

[54] Z. Zhu, L. Sun, F. Chu and M. Liu, Single-machine group scheduling with resource allocation and learning effect. *Comput. Ind. Eng.* **60** (2011) 148–157.

[55] Z. Zhu, F. Chu, L. Sun and M. Liu, Single machine scheduling with resource allocation and learning effect considering the rate-modifying activity. *Appl. Math. Model.* **37** (2013) 5371–5380.