

UNRELATED-PARALLEL MACHINE SCHEDULING WITH SIMULTANEOUS CONSIDERATIONS OF RESOURCE-DEPENDENT PROCESSING TIMES AND RATE-MODIFYING ACTIVITIES

TENG-RUEY CHANG¹, HSIN-TAO LEE¹, DAR-LI YANG² AND SUH-JENQ YANG¹

¹Department of Industrial Management
Nan Kai University of Technology
No. 568, Zhongzheng Rd., Caotun Township, Nantou County 54243, Taiwan
{ t237; htlee; sjyang }@nkut.edu.tw

²Department of Information Management
National Formosa University
No. 64, Wunhua Rd., Huwei Township, Yunlin 632, Taiwan
dlyang@nfu.edu.tw

Received September 2013; revised January 2014

ABSTRACT. *In this paper we investigate unrelated parallel-machine scheduling problems with simultaneous considerations of resource-dependent processing times and rate-modifying activities. The scheduler has an option to perform a rate-modifying activity on each machine to improve their production efficiency. We examine two types of resource allocation, namely the linear resource allocation model and the convex resource allocation model. We aim to find the optimal resource allocations, the optimal rate-modifying activity positions, and the optimal job sequence to minimize the cost function containing the total completion time plus the resource allocation and the cost function containing the total machine load plus the resource allocation, respectively. If the number of machines is fixed, we show that the problem under study can be formulated as an assignment problem and thus can be solved in a polynomial time algorithm.*

Keywords: Scheduling, Unrelated parallel-machine, Resource allocation, Rate-modifying activity

1. **Introduction.** Scheduling a rate-modifying activity becomes a popular topic among researchers in the last decade. The rate-modifying activity is an activity that changes the production efficiency of a machine. Lee and Leon [11] were among the pioneers who brought the concept of the rate-modifying activity into the field of scheduling. Lee and Leon considered several single-machine scheduling problems in this class to minimize the makespan, flowtime, the weighted flowtime, and the maximum lateness. They assumed that the machine may have at most one rate-modifying activity during the planning horizon. They proposed polynomial or pseudopolynomial algorithms to solve the problems under consideration. He et al. [7] studied a restricted version of the problem introduced by Lee and Leon [11]. They showed that some related problems become NP-hard with the restriction. Zhao et al. [36] extended some objectives studied by Lee and Leon [11] to the identical parallel-machine environment. For the problem to minimize the total completion time, they provided a polynomial algorithm to solve it. For the problem to minimize the weighted completion time, they introduced a pseudopolynomial dynamic programming algorithm to solve the case where the jobs satisfy an agreeable condition. Lodree and Geiger [13] considered scheduling with a rate-modifying activity under the assumption that a job's processing time is time-dependent on a single-machine. The goal was to derive the optimal policy for assigning a single rate-modifying activity in a sequence to

minimize the makespan. They showed that under given conditions, the optimal policy is to schedule the rate-modifying activity in the middle of the job sequence. Hsu et al. [8] extended the model studied by Zhao et al. [36] to an unrelated parallel-machine setting. For given the number of machines, they proposed an efficient polynomial time algorithm such that the total completion time minimization problem can be solved in a lower order algorithm. Zhao et al. [35] considered the parallel-machine scheduling with deteriorating jobs and rate-modifying activities to minimize the total completion time. They proved that the problem remains polynomially solvable. Yang et al. [33] showed that the time complexity of the model studied by Hsu et al. [8] can be reduced. More recent papers which considered scheduling jobs with the rate-modifying activity include Ji and Cheng [10], Wang et al. [27], Zhao and Tang [37], Yin et al. [34], Yang and Yang [31], and Rustogi and Strusevich [18].

On the other hand, in most scheduling studies, the job processing times are treated as constant parameters. However, in many practical situations the scheduler can control the processing times by the addition of resources, such as worker, energy, gas, fuel, and catalyzer, to the job operations. Resource-dependent processing times appear whenever resources can be employed to adjust processing requirements. For example, in the steel industry, the ingot preheated process before hot rolling consumes lots of energy. The manufacture can improve the efficiency of the production by allocating some extra energy. Another practical example comes from a case where a manufacturer can improve the production efficiency by adding the workers. In these situations, job scheduling and resource allocation decisions should be coordinated carefully to achieve the most efficient system performance. Pioneering research in the area of scheduling with resource-dependent processing times was conducted by Vickson [23,24] and Van Wassenhove and Baker [22]. Janiak [9] described an interesting application of a scheduling problem with resource-dependent processing times in steel mills. Trick [21] concerned another interesting application of resource allocation arising from scheduling in a machine-tooling environment, where the job processing time is a function of the feed rate and the spindle speed used for each operation. Comprehensive surveys of different scheduling problems concerning resource-dependent processing time are presented by Nowicki and Zdrzalka [15], Chudzik et al. [4], and Shabtay and Steiner [19]. For new trends in scheduling with resource-dependent processing time, we refer the reader to Wang et al. [25], Wang and Wang [26], Wang and Wang [28], Wei et al. [29], Zhu et al. [39], Rudek and Rudek [17] and Oron [16].

Parallel-machine scheduling problems with resource-dependent processing time are common in practice. Alidaee and Ahmadian [1] considered parallel-machine scheduling problems with controllable job processing times. They aimed to minimize the total processing cost plus the total flow time and the total processing cost plus the total weighted earliness and weighted tardiness, respectively. They showed that both problems can be solved in polynomial time algorithms. Cheng et al. [3] focused on unrelated parallel-machine scheduling problems where the job processing times can be compressed by a convex function. The objectives were to minimize the total compression cost plus the total flow time and the total compression cost plus the sum of earliness and tardiness costs, respectively. They provided polynomial time algorithms for solving both problems. Su and Lien [20] considered the problem of scheduling a set of jobs on parallel machines when the processing time of each job depends on the amount of resource consumed. They aimed to find the allocation of resources to jobs and jobs to machines to minimize the makespan. The problem has been proven to be NP-hard even for the fixed job processing times. They

proposed a heuristic algorithm for solving the problem. Lee and Yang [12] studied multi-objective scheduling problems involving deterioration effects and resource allocations simultaneously on an unrelated parallel-machine setting. They developed polynomial time algorithms for all the problems studied if the number of machines is given. Edis et al. [5] presented a review and discussion of studies on parallel-machine scheduling problems with additional resources.

It is natural to study scheduling problems combining resource-dependent processing time and rate-modifying activity. To the best of our knowledge, only Zhu et al. [38] and Yang et al. [30] studied the problem of this type. Zhu et al. [38] addressed single-machine scheduling problems in which the actual job processing times are determined by resource allocation function, its position in a sequence and a rate-modifying activity simultaneously. They showed that all the problems studied are polynomially solvable. Yang et al. [30] considered unrelated parallel-machine scheduling involving controllable processing times and rate-modifying activities simultaneously. If the number of machines is a given constant, they proposed an efficient polynomial-time algorithm to solve the proposed problem.

Motivated by the observation in a product assembly process, the processing time of a job depends both on the number of labors allocated to process the job and the labors' skills. The production efficiency can be improved after a training course for the labors. The training course can be considered as a type of rate-modifying activity. Consequently, in this paper we investigate unrelated parallel-machine scheduling problems with simultaneous considerations of resource-dependent processing times and rate-modifying activities. We aim to determine the optimal resource allocations, the optimal rate-modifying activity positions, and the optimal job sequence to minimize two cost functions. The rest of this paper is organized as follows. In Section 2 we introduce and formulate the problem. Some preliminary results for further analysis are provided in Section 3. We find the optimal solutions for the objective functions in Sections 4, 5, 6 and 7. The last section concludes this paper.

2. Problem Formulation. In this section we first introduce the notation to be used throughout the paper, followed by formulation of the problem.

- n : the number of jobs;
- m : the number of machines, $m < n$;
- J_j : job j , $j = 1, 2, \dots, n$;
- M_i : machine i , $i = 1, 2, \dots, m$;
- n_i : the number of jobs assigned to process on machine M_i , $n = \sum_{i=1}^m n_i$, $i = 1, 2, \dots, m$;
- a_{ij} : the normal processing time of job J_j scheduled *before* a rate-modifying activity on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
- b_{ij} : the normal processing time of job J_j scheduled *after* a rate-modifying activity on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
- v_{ij} : the compression rate of job J_j on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
- u_{ij} : the amount of resource allocated to job J_j on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
- \bar{u}_{ij} : the upper bound on the amount of resource allocated to job J_j on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
- p_{ijr} : the actual processing time of job J_j scheduled in the r th position on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ and $r = 1, 2, \dots, n_i$;
- p_{ijh} : the actual processing time of job J_j scheduled in the h th position *to the last job* on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ and $h = 1, 2, \dots, n_i$;
- k : a positive constant for the convex resource allocation model;

- G_{ij} : the per unit time cost associated with the resource allocation to job J_j on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
 C_{ij} : the completion time of job J_j on machine M_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$;
 C_{\max}^i : the makespan of machine M_i , $i = 1, 2, \dots, m$;
 S_i : the set of jobs assigned to machine M_i , $i = 1, 2, \dots, m$;
 S : the set of all feasible schedules.

Subscript

$[ir']$: the job scheduled in the r th position on machine M_i ;

$[ih]$: the job scheduled in the h th position to the last job on machine M_i .

We consider a set of n independent jobs to be processed on a set of m unrelated parallel machines. Each job J_j , $j = 1, 2, \dots, n$, becomes available for processing at time zero. Preemption is not allowed and each machine is only able to process one job at a time. Let $S = (S_1, S_2, \dots, S_m)$ be a schedule for the machines. Then $S_i \cap S_j = \emptyset$, $\forall i \neq j$, and $\bigcup_{i=1}^m S_i = \{J_1, \dots, J_n\}$. The scheduler has an option to perform a rate-modifying activity on each machine to improve their production efficiency. We assume that each machine may have at most one rate-modifying activity during the planning horizon. We further assume that the rate-modifying activity can be performed immediately *after* completing the processing of any job and the duration of the rate-modifying activity on machine M_i is t_i . From a practical point of view, however, scheduling a rate-modifying activity before we process any job on a machine is meaningless. So, in this paper we do not consider the case of scheduling a rate-modifying activity before we process any job on a machine. For convenience, we say that the rate-modifying activity is scheduled in position k_i ($0 < k_i \leq n_i$) on machine M_i if it is performed immediately *after* the completion of the job scheduled in the k_i th position on machine M_i .

In this study, the processing time of a job is controllable by additional resources. We examine two different types of resource allocation. In the first one that describes the bounded linear function, the actual processing time of job J_j scheduled in position r is given by the following function:

$$p_{ijr'} = \begin{cases} a_{ij} - v_{ij}u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad r \leq k_i, \\ b_{ij} - v_{ij}u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad r > k_i, \end{cases} \quad (1)$$

where $a_{ij} > b_{ij} > 0$, $0 \leq u_{ij} \leq \bar{u}_{ij} < \frac{b_{ij}}{v_{ij}}$, and $v_{ij} > 0$. The second model concerns a convex decreasing function whereby if job J_j is scheduled in the r th position on machine M_i , its actual processing time is given by

$$p_{ijr'} = \begin{cases} \left(\frac{a_{ij}}{u_{ij}} \right)^k, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad r \leq k_i, \\ \left(\frac{b_{ij}}{u_{ij}} \right)^k, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad r > k_i, \end{cases} \quad (2)$$

where $a_{ij} > b_{ij} > 0$, $u_{ij} > 0$, and k is a positive constant.

The objective of this study is to determine the optimal resource allocations, the optimal rate-modifying activity positions, and the optimal job sequence such that one of the following two cost functions is minimized: the cost function containing the total completion time plus the resource allocation and the cost function containing the total machine load [32] plus the resource allocation, i.e.,

$$TC = \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} + \sum_{i=1}^m \sum_{j=1}^{n_i} G_{ij}u_{ij} \quad (3)$$

and

$$TL = \sum_{i=1}^m C_{\max}^i + \sum_{i=1}^m \sum_{j=1}^{n_i} G_{ij} u_{ij}. \tag{4}$$

Following the three-field notation in Graham et al. [6], we denote our problems as $Rm|rm, lin|TC$, $Rm|rm, con|TC$, $Rm|rm, lin|TL$, and $Rm|rm, con|TL$, respectively, where rm indicates the rate-modifying activity and lin and con represent the linear resource allocation model and the convex resource allocation model, respectively.

3. Preliminary Analysis. In this section we present two important lemmas for an optimal schedule of the problem under study.

Lemma 3.1. [14] *The number of nonnegative integer solutions to $x_1 + x_2 + \dots + x_m = n$ is $C(n + m - 1, m - 1) = \frac{(n+m-1)!}{(m-1)!n!}$.*

Lemma 3.2. [8] *The number $C(n + m, m)$ is bounded from above by $\frac{(2n)^m}{m!}$.*

4. Minimization of $Rm|rm, lin|TC$. In this section we consider the $Rm|rm, lin|TC$ problem. First, we denote that the rate-modifying activity is scheduled in position l_i ($0 \leq l_i < n_i$) on machine M_i if it is scheduled immediately *before* the job scheduled in the l_i th position *to the last job* on machine M_i . Note that if $l_i = n_i$, it indicates that we perform the rate-modifying activity *before* we process any job on machine M_i ; while if $l_i = 0$, it means that there is no rate-modifying activity scheduled on machine M_i . In addition, from a practical point of view, scheduling a rate-modifying activity before we process any job on a machine is meaningless. Then, the actual processing time of job J_j if it is scheduled in the h th position *to the last job* on machine M_i is given by:

$$p_{ijh} = \begin{cases} b_{ij} - v_{ij}u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad h \leq l_i, \\ a_{ij} - v_{ij}u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad h > l_i. \end{cases} \tag{5}$$

Let (l_1, l_2, \dots, l_m) be the rate-modifying activity position vector for the machines when the rate-modifying activity is scheduled immediately *before* the job scheduled in the l_i th position *to the last job* on machine M_i and $p_{[ih]}$ be the actual processing time a job scheduled in the h th position *to the last job* on machine M_i , for $i = 1, 2, \dots, m$ and $h = 1, 2, \dots, n_i$. Thus, for a given vector (l_1, l_2, \dots, l_m) , if we substitute $C_{[ih]} = \sum_{r=1}^h p_{[ir]}$ into (3), then we obtain that

$$\begin{aligned} TC &= \sum_{i=1}^m \left(\sum_{h=1}^{l_i} h(b_{[ih]} - v_{[ih]}u_{[ih]}) + \sum_{h=l_i+1}^{n_i} h(a_{[ih]} - v_{[ih]}u_{[ih]}) \right) \\ &\quad + \sum_{i=1}^m l_i t_i + \sum_{i=1}^m \sum_{h=1}^{n_i} G_{[ih]} u_{[ih]} \\ &= \sum_{i=1}^m \left[\sum_{h=1}^{l_i} (hb_{[ih]} + (G_{[ih]} - hv_{[ih]})u_{[ih]}) + \sum_{h=l_i+1}^{n_i} (ha_{[ih]} + (G_{[ih]} - hv_{[ih]})u_{[ih]}) \right] \\ &\quad + \sum_{i=1}^m l_i t_i. \end{aligned} \tag{6}$$

Note that $[ih]$ denotes the job scheduled in the h th position *to the last job* on machine M_i . Obviously, for a given vector (l_1, l_2, \dots, l_m) , we can ignore the times required to perform the rate-modifying activities, as it represents a constant in (6).

Lemma 4.1. *Given a sequence, the optimal resource allocation for the $Rm|rm, lin|TC$ problem can be determined by*

$$u_{[ih]}^* = \begin{cases} \bar{u}_{[ih]}, & \text{if } G_{[ih]} < hv_{[ih]}, \\ 0, & \text{if } G_{[ih]} \geq hv_{[ih]}, \end{cases} \tag{7}$$

where $u_{[ih]}^*$ denotes the optimal resource allocation of a job scheduled in the h th position to the last job on machine M_i .

Proof: From (6), we see that for any sequence, the optimal resource allocation of a job in a position with $G_{[ih]} - hv_{[ih]} < 0$ should be its upper bound on the amount of resource $\bar{u}_{[ih]}$, and the optimal resource allocation of a job in a position with $G_{[ih]} - hv_{[ih]} \geq 0$ should be 0. It should be noted that if $u_{[ih]} = 0$, it means that no additional resource is allocated on the job. This completes the proof.

Next, we define $y_{ijs} = 1$ if job J_j is in the s th position *to the last job* processed on machine M_i and $y_{ijs} = 0$ otherwise. Let w_{ijs} be the positional weight of job J_j scheduled in the s th position *to the last job* on machine M_i . Then, the $Rm|rm, lin|TC$ problem can be formulated as follows:

$$\text{Minimize } \sum_{i=1}^m \sum_{j=1}^n \sum_{s=1}^n w_{ijs} y_{ijs}. \tag{8}$$

$$\text{subject to } \sum_{j=1}^n y_{ijs} = 1, \quad i = 1, 2, \dots, m, \quad s = 1, 2, \dots, l_i, \tag{9}$$

$$\sum_{j=1}^n y_{ijs} \leq 1, \quad i = 1, 2, \dots, m, \quad s = l_i + 1, l_i + 2, \dots, n, \tag{10}$$

$$\sum_{i=1}^m \sum_{s=1}^n y_{ijs} = 1, \quad j = 1, 2, \dots, n, \tag{11}$$

$$\sum_{j=1}^n y_{ij1} \geq \sum_{j=1}^n y_{ij2} \geq \dots \geq \sum_{j=1}^n y_{ijn}, \quad i = 1, 2, \dots, m, \tag{12}$$

$$y_{ijs} \in \{0, 1\}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s = 1, 2, \dots, n, \tag{13}$$

where

$$w_{ijs} = \begin{cases} sb_{ij} + (G_{ij} - sv_{ij})u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s \leq l_i, \\ sa_{ij} + (G_{ij} - sv_{ij})u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s > l_i, \end{cases} \tag{14}$$

and

$$u_{ij}^* = \begin{cases} \bar{u}_{ij}, & \text{if } G_{ij} < sv_{ij}, \\ 0, & \text{if } G_{ij} \geq sv_{ij}. \end{cases} \tag{15}$$

Constraint (9) ensures that each position ($s \leq l_i, i = 1, 2, \dots, m$) on each machine is taken by one job. Constraint (10) makes sure that each position ($s \geq l_i + 1, i = 1, 2, \dots, m$) on each machine is taken by at most one job. Constraint (11) ensures that each job is scheduled exactly once. Constraint (12) makes sure that on every machine, the unassigned positions must precede all the assigned positions, so that $y_{ijs} = 1$ if and only if job J_j is indeed in the s th position *to the last job* on machine M_i .

Lemma 4.2 is useful to find the optimal solution for the $Rm|rm, lin|TC$ problem.

Lemma 4.2. *In (14), $w_{ij1} \leq w_{ij2} \leq \dots \leq w_{ijl_i}$ and $w_{ij(l_i+1)} \leq w_{ij(l_i+2)} \leq \dots \leq w_{ijn}$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.*

Proof: Let f and g be two positions to the last job processed on machine M_i and $f > g$.

Case a: $f \leq l_i$ (thereby $g < l_i$)

If $f \leq l_i$, from (14) and (15), we obtain the following results:

- (i) If $G_{ij} \geq fv_{ij}$, then we have that $u_{ij}^* = 0$ and $w_{ijf} = fb_{ij} > gb_{ij} = w_{ijg}$.
- (ii) If $G_{ij} < gv_{ij}$, then we obtain that $u_{ij}^* = \bar{u}_{ij}$ and $w_{ijf} = fb_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij} = f(b_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} > g(b_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} = w_{ijg}$.
- (iii) If $gv_{ij} \leq G_{ij} < fv_{ij}$, then we know that $w_{ijf} = fb_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij}$ and $w_{ijg} = gb_{ij}$. We see that $w_{ijf} - w_{ijg} = fb_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij} - gb_{ij} = f(b_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} - gb_{ij} > f(b_{ij} - v_{ij}\bar{u}_{ij}) + gv_{ij}\bar{u}_{ij} - gb_{ij} = f(b_{ij} - v_{ij}\bar{u}_{ij}) - g(b_{ij} - v_{ij}\bar{u}_{ij}) > 0$. Thus, $w_{ijf} > w_{ijg}$.

Case b: $g \geq l_i + 1$ (thereby $f > l_i + 1$)

If $g \geq l_i + 1$, from (14) and (15), we have the following results:

- (i) If $G_{ij} \geq fv_{ij}$, then we obtain that $u_{ij}^* = 0$ and $w_{ijf} = fa_{ij} > ga_{ij} = w_{ijg}$.
- (ii) If $G_{ij} < gv_{ij}$, then we have that $u_{ij}^* = \bar{u}_{ij}$ and $w_{ijf} = fa_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij} = fa_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij} = f(a_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} > g(a_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} = w_{ijg}$.
- (iii) If $gv_{ij} \leq G_{ij} < fv_{ij}$, then we have that $w_{ijf} = fa_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij}$ and $w_{ijg} = ga_{ij}$. We see that $w_{ijf} - w_{ijg} = fa_{ij} + (G_{ij} - fv_{ij})\bar{u}_{ij} - ga_{ij} = f(a_{ij} - v_{ij}\bar{u}_{ij}) + G_{ij}\bar{u}_{ij} - ga_{ij} \geq f(a_{ij} - v_{ij}\bar{u}_{ij}) + gv_{ij}\bar{u}_{ij} - ga_{ij} = f(a_{ij} - v_{ij}\bar{u}_{ij}) + g(a_{ij} - v_{ij}\bar{u}_{ij}) = (f - g)(a_{ij} - v_{ij}\bar{u}_{ij}) > 0$. Thus, $w_{ijf} > w_{ijg}$.

So, we have that $w_{ij1} \leq w_{ij2} \leq \dots \leq w_{ijl_i}$ and $w_{ij(l_i+1)} \leq w_{ij(l_i+2)} \leq \dots \leq w_{ijn}$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. This completes the proof.

In what follows we prove that the $Rm|rm, lin|TC$ problem can be optimally solved in $O(n^{m+3})$ time.

Theorem 4.1. *The $Rm|rm, lin|TC$ problem can be solved in $O(n^{m+3})$ time.*

Proof: From Lemma 4.2, we know that $w_{ij1} \leq w_{ij2} \leq \dots \leq w_{ijl_i}$ and $w_{ij(l_i+1)} \leq w_{ij(l_i+2)} \leq \dots \leq w_{ijn}$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$, and thus $\sum_{j=1}^n y_{ij1} \geq \sum_{j=1}^n y_{ij2} \geq \dots \geq \sum_{j=1}^n y_{ijl_i}$ and $\sum_{j=1}^n y_{ij(l_i+1)} \geq \sum_{j=1}^n y_{ij(l_i+2)} \geq \dots \geq \sum_{j=1}^n y_{ijn}$, for $i = 1, 2, \dots, m$. In addition, from constraints (9) and (10), we know that $\sum_{j=1}^n y_{ijl_i} (= 1) \geq \sum_{j=1}^n y_{ij(l_i+1)} (\leq 1)$ for $i = 1, 2, \dots, m$. Thus, the inequality $\sum_{j=1}^n y_{ij1} \geq \sum_{j=1}^n y_{ij2} \geq \dots \geq \sum_{j=1}^n y_{ijn}$ holds, for $i = 1, 2, \dots, m$. As a result, we can remove constraint (12) from the formulation without affecting the optimal solution value of the problem. Hence, the $Rm|rm, lin|TC$ problem can be re-formulated as the following assignment problem and thus can be solved in $O(mn^3)$ time [2]:

$$\begin{aligned} & \text{Minimize } \sum_{i=1}^n \sum_{j=1}^m \sum_{s=1}^n w_{ijs} y_{ijs} \\ & \text{subject to (9), (10), (11) and (13).} \end{aligned}$$

Moreover, since there are n jobs to be assigned to m unrelated parallel machines, we obtain that $0 < l_1 + l_2 + \dots + l_m \leq n$. Let $l_{m+1} = n - (l_1 + l_2 + \dots + l_m) \geq 0$. This means that $l_1 + l_2 + \dots + l_{m+1} = n$. By Lemma 3.1, the number of nonnegative integer solutions to $l_1 + l_2 + \dots + l_{m+1} = n$ is $C(n + m, m)$. By Lemma 3.2, the number $C(n + m, m)$ is bounded from above by $(2n)^m / m!$. Therefore, we conclude that the $Rm|rm, lin|TC$ problem can be solved in $O(n^{m+3})$ time. This completes the proof.

Clearly, if the number of machines m is fixed, then the $Rm|rm, lin|TC$ problem is polynomially solvable.

From the above analysis, we propose the following algorithm to solve the $Rm|rm, lin|TC$ problem.

Algorithm 1.

Step 1: For all the possible vectors (l_1, l_2, \dots, l_m) , calculate w_{ijs} by using (14), for $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ and $s = 1, 2, \dots, n$.

Step 2: Solve the corresponding assignment problem to obtain the local optimal schedule and the corresponding total cost for each possible vector (l_1, l_2, \dots, l_m) .

Step 3: The global optimal solution for the problem is the one with the minimum value of the total cost TC .

Step 4: Calculate the optimal resources by using (7).

Step 5: Calculate the actual processing times by using (5).

The following example illustrates applying Algorithm 1 to find the optimal solution of an 11 jobs instance. For given the position of the rate-modifying activity on each machine, i.e., (l_1, l_2) , we solve the corresponding assignment problem using LINGO version 11.0 on a personal computer with Intel Core i7-2600 CPU @3.40GHz and 8GB RAM under Windows 7.

Example 4.1. *There are 11 jobs to be processed on two identical parallel machines. The set of job parameters is presented in Table 1. The duration of the rate-modifying activity is 2.0 for both machines.*

TABLE 1. Job parameters for Example 4.1

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}
a_j	28.0	16.0	11.0	14.0	22.0	27.0	19.0	15.0	25.0	18.0	20.0
b_j	21.0	9.6	9.9	11.9	14.3	21.6	18.05	10.5	12.5	14.4	18.0
\bar{u}_j	5.5	4.2	2.5	3.1	5.2	3.8	4.5	3.5	3.6	4.8	5.6
v_j	2.4	1.2	3.1	1.5	2.2	2.5	1.6	1.8	1.4	2.1	1.9
G_j	10.0	16.0	8.0	14.0	12.0	17.0	6.0	15.0	11.0	13.0	18.0

TABLE 2. Optimal job sequences and total cost for all the possible positions of rate-modifying activities for Example 4.1

(l_1, l_2)	Job sequences	TC
(0, 0)	$M_1 = (J_3, J_7, J_8, J_{11}, J_9, J_6), M_2 = (J_4, J_2, J_{10}, J_5, J_1)$	582.5
(0, 1)	$M_1 = (J_3, J_7, J_8, J_{11}, J_6, J_1), M_2 = (J_4, J_2, J_{10}, J_5, J_9)$	574.0
(0, 2)	$M_1 = (J_3, J_7, J_8, J_{11}, J_5, J_6), M_2 = (J_4, J_2, J_{10}, J_9, J_1)$	554.5
(0, 3)	$M_1 = (J_3, J_7, J_2, J_{10}, J_{11}, J_6), M_2 = (J_4, J_8, J_9, J_5, J_1)$	533.6
(0, 4)	$M_1 = (J_3, J_7, J_8, J_{10}, J_{11}, J_6), M_2 = (J_4, J_2, J_9, J_5, J_1)$	510.0
(1, 1)	$M_1 = (J_3, J_7, J_8, J_{11}, J_6, J_1), M_2 = (J_4, J_2, J_{10}, J_5, J_9)$	569.0
(1, 2)	$M_1 = (J_7, J_8, J_{11}, J_5, J_1), M_2 = (J_3, J_4, J_2, J_{10}, J_9, J_6)$	551.1
(1, 3)	$M_1 = (J_3, J_7, J_8, J_{10}, J_{11}, J_1), M_2 = (J_4, J_2, J_9, J_5, J_6)$	530.2
(1, 4)	$M_1 = (J_3, J_7, J_8, J_{10}, J_{11}, J_1), M_2 = (J_4, J_2, J_9, J_5, J_6)$	506.6
(2, 2)	$M_1 = (J_3, J_7, J_2, J_{11}, J_9, J_1), M_2 = (J_4, J_8, J_{10}, J_5, J_6)$	537.7
(2, 3)	$M_1 = (J_3, J_7, J_{10}, J_{11}, J_9, J_1), M_2 = (J_4, J_8, J_2, J_5, J_6)$	522.5
(2, 4)	$M_1 = (J_3, J_7, J_8, J_{11}, J_{10}, J_1), M_2 = (J_4, J_2, J_9, J_5, J_6)$	503.4
(3, 3)	$M_1 = (J_3, J_7, J_{10}, J_9, J_{11}, J_1), M_2 = (J_4, J_8, J_2, J_5, J_6)$	513.0
(3, 4)	$M_1 = (J_3, J_7, J_8, J_9, J_{11}, J_1), M_2 = (J_4, J_2, J_5, J_{10}, J_6)$	495.7
(4, 4)	$M_1 = (J_3, J_7, J_8, J_9, J_{10}, J_1), M_2 = (J_4, J_2, J_5, J_{11}, J_6)$	479.7^a

^a: the minimum total cost

TABLE 3. The optimal resource allocations and the actual processing times for Example 4.1

	M_1						M_2				
r	1	2	3	4	5	6	1	2	3	4	5
$J_{[ir']}$	J_3	J_7	J_8	J_9	J_{10}	J_1	J_4	J_2	J_5	J_{11}	J_6
$u_{[ir']}^*$	2.5	4.5	0	0	0	0	0	0	0	0	0
$p_{[ir']}$	3.25	11.8	10.5	12.5	14.4	21.0	14.0	9.6	14.3	18.0	21.6

$J_{[ir']}$: a job scheduled in the r th position on machine M_i ; $u_{[ir']}^*$: the optimal resource allocated on job $J_{[ir']}$; $p_{[ir']}$: the actual processing time of job $J_{[ir']}$.

Solution. Table 2 shows the optimal job sequences and the total cost for all the possible positions of rate-modifying activities of the problem. From Table 2, we see that the optimal solution for this example is obtained when $(l_1, l_2) = (4, 4)$, the job sequences on machines M_1 and M_2 are $M_1 = (J_3, J_7, J_8, J_9, J_{10}, J_1)$ and $M_2 = (J_4, J_2, J_5, J_{11}, J_6)$, respectively. The minimum total cost is 479.7. It should be noted that $(l_1, l_2) = (4, 4)$ means that the rate-modifying activity is located immediately after the completion of job J_7 on machine M_1 and the rate-modifying activity is performed immediately after the completion of job J_4 on machine M_2 . Furthermore, we obtain the optimal resource allocations and the actual processing times of jobs for this example and the results are summarized in Table 3. From Table 3, we know that the actual processing times of jobs J_3 and J_7 are compressed by additional resources.

5. Minimization of $Rm|rm, con|TC$. In this section we investigate the $Rm|rm, con|TC$ problem. Following the analysis of the $Rm|rm, lin|TC$ problem, the actual processing time of job J_j if it is scheduled in the h th position to the last job on machine M_i is defined by:

$$p_{ijh} = \begin{cases} \left(\frac{b_{ij}}{u_{ij}}\right)^k, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad h \leq l_i, \\ \left(\frac{a_{ij}}{u_{ij}}\right)^k, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i, \quad h > l_i. \end{cases} \tag{16}$$

For a given vector (l_1, l_2, \dots, l_m) , we see that

$$TC = \sum_{i=1}^m \left[\sum_{h=1}^{l_i} h \left(\frac{b_{[ih]}}{u_{[ih]}}\right)^k + \sum_{h=l_i+1}^{n_i} h \left(\frac{a_{[ih]}}{u_{[ih]}}\right)^k \right] + \sum_{i=1}^m l_i t_i + \sum_{i=1}^m \sum_{h=1}^{n_i} G_{[ih]} u_{[ih]}. \tag{17}$$

Again, we can ignore the term of $\sum_{i=1}^m l_i t_i$ in (17).

Lemma 5.1. *Given a sequence, the optimal resource allocation, $u_{[ih]}^*$, for the $Rm|rm, con|TC$ problem is*

$$u_{[ih]}^* = \begin{cases} \left(\frac{hk}{G_{[ih]}}\right)^{\frac{1}{k+1}} b_{[ih]}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad h \leq l_i, \\ \left(\frac{hk}{G_{[ih]}}\right)^{\frac{1}{k+1}} a_{[ih]}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad h > l_i. \end{cases} \tag{18}$$

Proof: From (17), we take the first derivative of TC with respect to $u_{[ir]}$ and let it be equal to 0. Then, we obtain that

$$u_{[ih]}^* = \begin{cases} \left(\frac{hk}{G_{[ih]}}\right)^{\frac{1}{k+1}} b_{[ih]}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, h \leq l_i, \\ \left(\frac{hk}{G_{[ih]}}\right)^{\frac{1}{k+1}} a_{[ih]}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, h > l_i. \end{cases}$$

Since (17) is a convex function, (18) provides necessary and sufficient conditions for optimality. This completes the proof.

Furthermore, by substituting (18) into (17), we see that

$$TC = \sum_{i=1}^m \sum_{h=1}^{n_i} \left(k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}\right) \theta_{[ih]} \phi_{ih}, \tag{19}$$

where

$$\theta_{[ih]} = \begin{cases} (G_{[ih]} b_{[ih]})^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, h \leq l_i, \\ (G_{[ih]} a_{[ih]})^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, h > l_i, \end{cases} \tag{20}$$

and

$$\phi_{ih} = h^{\frac{1}{k+1}}, \tag{21}$$

for $i = 1, 2, \dots, m$ and $h = 1, 2, \dots, n_i$.

In what follows we show that the $Rm|rm, con|TC$ problem can be solved in $O(n^{m+3})$ time.

Theorem 5.1. *The $Rm|rm, con|TC$ problem can be solved in $O(n^{m+3})$ time.*

Proof: Following the analysis of the $Rm|rm, lin|TC$ problem, we define $y_{ijs} = 1$ if job J_j is in the s th position to the last job processed on machine M_i and $y_{ijs} = 0$ otherwise. Then, the $Rm|rm, con|TC$ problem can be formulated as follows:

$$\text{Minimize } \sum_{i=1}^m \sum_{j=1}^n \sum_{s=1}^n w_{ijs} y_{ijs}. \tag{22}$$

$$\text{subject to } \sum_{j=1}^n y_{ijs} = 1, \quad i = 1, 2, \dots, m, \quad s = 1, 2, \dots, l_i, \tag{23}$$

$$\sum_{j=1}^n y_{ijs} \leq 1, \quad i = 1, 2, \dots, m, \quad s = l_i + 1, l_i + 2, \dots, n, \tag{24}$$

$$\sum_{i=1}^m \sum_{s=1}^n y_{ijs} = 1, \quad j = 1, 2, \dots, n, \tag{25}$$

$$\sum_{j=1}^n y_{ij1} \geq \sum_{j=1}^n y_{ij2} \geq \dots \geq \sum_{j=1}^n y_{ijn}, \quad i = 1, 2, \dots, m, \tag{26}$$

$$y_{ijs} \in \{0, 1\}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s = 1, 2, \dots, n, \tag{27}$$

where

$$w_{ijs} = \begin{cases} \left(k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}\right) (G_{ij} b_{ij})^{\frac{k}{k+1}} s^{\frac{1}{k+1}}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s \leq l_i, \\ \left(k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}\right) (G_{ij} a_{ij})^{\frac{k}{k+1}} s^{\frac{1}{k+1}}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad s > l_i. \end{cases} \tag{28}$$

From (28), we know that $w_{ij1} \leq w_{ij2} \leq \dots \leq w_{ijl_i}$ and $w_{ij(l_i+1)} \leq w_{ij(l_i+2)} \leq \dots \leq w_{ijn}$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$, and thus $\sum_{j=1}^n y_{ij1} \geq \sum_{j=1}^n y_{ij2} \geq \dots \geq \sum_{j=1}^n y_{ijl_i}$ and $\sum_{j=1}^n y_{ij(l_i+1)} \geq \sum_{j=1}^n y_{ij(l_i+2)} \geq \dots \geq \sum_{j=1}^n y_{ijn}$, for $i = 1, 2, \dots, m$. Therefore, by the proof of Theorem 4.1, we obtain that the $Rm|rm, con|TC$ problem can be solved in $O(n^{m+3})$ time. This completes the proof.

We can obtain the optimal value of the $Rm|rm, con|TC$ problem in a similar manner of Algorithm 1.

6. Minimization of $Rm|rm, lin|TL$. In this section we consider the $Rm|rm, lin|TL$ problem. We denote by the subscript $[ir']$ the job scheduled in the r th position on machine M_i , for $j = 1, 2, \dots, m$ and $r = 1, 2, \dots, n_i$. Let (k_1, k_2, \dots, k_m) be the rate-modifying activity position vector for the machines when the rate-modifying activity is scheduled immediately *after* the completion of the job scheduled in the k_i th position on machine M_i and $p_{[ir']}$ be the actual processing time a job scheduled in the r th position on machine M_i , for $i = 1, 2, \dots, m$ and $r = 1, 2, \dots, n_i$. Thus, for a given vector (k_1, k_2, \dots, k_m) , if we substitute $C_{[ir']} = \sum_{h=1}^r p_{[ih']}$ into (4), then we obtain that

$$\begin{aligned}
 TL &= \sum_{i=1}^m \left[\sum_{r=1}^{k_i} (a_{[ir']} - v_{[ir']}u_{[ir']}) + \sum_{r=k_i+1}^{n_i} (b_{[ir']} - v_{[ir']}u_{[ir']}) \right] \\
 &\quad + \sum_{i=1}^m t_i + \sum_{i=1}^m \sum_{r=1}^{n_i} G_{[ir']}u_{[ir']} \\
 &= \sum_{i=1}^m \left[\sum_{r=1}^{k_i} (a_{[ir']} + (G_{[ir']} - v_{[ir']}u_{[ir']}) + \sum_{r=k_i+1}^{n_i} (b_{[ir']} + (G_{[ir']} - v_{[ir']}u_{[ir']}) \right] \\
 &\quad + \sum_{i=1}^m t_i.
 \end{aligned} \tag{29}$$

Observe that $\sum_{i=1}^m t_i$ is a constant in (29). In addition, the rate-modifying activity position vector (k_1, k_2, \dots, k_m) is bounded from above by $(2n)^m/m!$. Then, performing a similar analysis of $Rm|rm, lin|TC$, we have the following results:

Lemma 6.1. *Given a sequence, the optimal resource allocation for the $Rm|rm, lin|TL$ problem can be determined by*

$$u_{[ir']}^* = \begin{cases} \bar{u}_{[ir]}, & \text{if } G_{[ir']} < v_{[ir]}, \\ 0, & \text{if } G_{[ir']} \geq v_{[ir]}, \end{cases} \tag{30}$$

where $u_{[ir']}^*$ denotes the optimal resource allocation of a job scheduled in the r th position on machine M_i .

Theorem 6.1. *The $Rm|rm, lin|TL$ problem can be solved in $O(n^{m+3})$ time.*

Proof: We define $x_{ijr'} = 1$ if job J_j is in the r th position on machine M_i and $x_{ijr'} = 0$ otherwise. Then, the $Rm|rm, lin|TL$ problem can be formulated as follows:

$$\text{Minimize } \sum_{i=1}^m \sum_{j=1}^n \sum_{r=1}^{n_i} z_{ijr'} x_{ijr'}. \tag{31}$$

$$\text{subject to } \sum_{j=1}^n x_{ijr'} = 1, \quad i = 1, 2, \dots, m, \quad r = 1, 2, \dots, k_i, \tag{32}$$

$$\sum_{j=1}^n x_{ijr'} \leq 1, \quad i = 1, 2, \dots, m, \quad r = k_i + 1, k_i + 2, \dots, n, \tag{33}$$

$$\sum_{i=1}^m \sum_{r=1}^n x_{ijr'} = 1, \quad j = 1, 2, \dots, n, \tag{34}$$

$$\sum_{j=1}^n x_{ij1'} \geq \sum_{j=1}^n x_{ij2'} \geq \dots \geq \sum_{j=1}^n x_{ijn'}, \quad i = 1, 2, \dots, m, \tag{35}$$

$$x_{ijr'} \in \{0, 1\}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad r = 1, 2, \dots, n, \tag{36}$$

where

$$z_{ijr'} = \begin{cases} a_{ij} + (G_{ij} - v_{ij})u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad r \leq k_i, \\ b_{ij} + (G_{ij} - v_{ij})u_{ij}, & i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad r > k_i, \end{cases} \tag{37}$$

and

$$u_{ij}^* = \begin{cases} \bar{u}_{ij}, & \text{if } G_{ij} < v_{ij}, \\ 0, & \text{if } G_{ij} \geq v_{ij}, \end{cases} \tag{38}$$

where $z_{ijr'}$ is the positional weight of job J_j scheduled in the r th position on machine M_i . From (37), we know that $z_{ij1'} = z_{ij2'} = \dots = z_{ijk'_i}$ and $z_{ij(k_i+1)'} = z_{ij(k_i+2)'} = \dots = z_{ijn'}$, for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$, and thus $\sum_{j=1}^n x_{ij1'} = \sum_{j=1}^n x_{ij2'} = \dots = \sum_{j=1}^n x_{ijk'_i}$ and $\sum_{j=1}^n x_{ij(k_i+1)'} = \sum_{j=1}^n x_{ij(k_i+2)'} = \dots = \sum_{j=1}^n x_{ijn'}$, for $i = 1, 2, \dots, m$. Therefore, by the proof of Theorem 4.1, we obtain that the $Rm|rm, lin|TL$ problem can be solved in $O(n^{m+3})$ time. This completes the proof.

Again, we can obtain the optimal value of the $Rm|rm, lin|TL$ problem in a similar manner of Algorithm 1.

7. Minimization of $Rm|rm, con|TL$. In this section we study the $Rm|rm, con|TL$ problem. Following the analysis of the $Rm|rm, lin|TL$ problem, we obtain that

$$TL = \sum_{i=1}^m \left[\sum_{r=1}^{k_i} \left(\frac{a_{[ir']}}{u_{[ir']}} \right)^k + \sum_{r=k_i+1}^{n_i} \left(\frac{b_{[ir']}}{u_{[ir']}} \right)^k \right] + \sum_{i=1}^m t_i + \sum_{i=1}^m \sum_{r=1}^{n_i} G_{[ir']} u_{[ir']}. \tag{39}$$

Again, the term of $\sum_{i=1}^m t_i$ in (39) is a constant. Then, we have the following lemma.

Lemma 7.1. *Given a sequence, the optimal resource allocation, $u_{[ir']}^*$, for the $Rm|rm, con|TL$ problem is*

$$u_{[ir']}^* = \begin{cases} \left(\frac{k}{G_{[ir']}} \right)^{\frac{1}{k+1}} a_{[ir']}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad r \leq k_i, \\ \left(\frac{k}{G_{[ir']}} \right)^{\frac{1}{k+1}} b_{[ir']}^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad r > k_i. \end{cases} \tag{40}$$

Proof: The proof is similar to that of Lemma 5.1.

Furthermore, by substituting (40) into (39), we see that

$$TL = \left(k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}} \right) \sum_{i=1}^m \sum_{r=1}^{n_i} \varphi_{[ir']}, \tag{41}$$

where

$$\varphi_{[ir']} = \begin{cases} (G_{[ir']} a_{[ir']})^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad r \leq k_i, \\ (G_{[ir']} b_{[ir']})^{\frac{k}{k+1}}, & i = 1, 2, \dots, m, \quad r > k_i. \end{cases} \tag{42}$$

Theorem 7.1. *The $Rm|rm, con|TL$ problem can be solved in $O(n^{m+3})$ time.*

Proof: The proof is similar to that of Theorem 6.1. Obviously, we can solve the $Rm|rm, con|TL$ problem in a similar manner of Algorithm 1.

8. Conclusions. In this paper we investigated scheduling problems with simultaneous considerations of resource-dependent processing times and rate-modifying activities on an unrelated parallel-machine setting. The linear and convex resource allocation models are examined, respectively. We aimed to find the optimal resource allocations, the optimal rate-modifying activity positions, and the optimal job sequence such that two cost functions can be minimized, namely the cost function containing the total completion time plus the resource allocation and the cost function containing the total machine load plus the resource allocation. If the number of machines is given, we introduced polynomial time algorithms for all the problems studied. It is worthy of future research to consider the problem with variable rate-modifying activity durations or multiple rate-modifying activities, or in more complicated machine setting, or optimizing other performance measures.

Acknowledgments. This research is supported by the National Science Council of Taiwan, under Grant No. NSC102-2221-E-252-010-MY2. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] B. Alidaee and A. Ahmadian, Two parallel machine sequencing problems involving controllable job processing times, *European Journal of Operational Research*, vol.70, pp.335-341, 1993.
- [2] P. Brucker, *Scheduling Algorithms*, Springer-Verlag Inc., New York, 2001.
- [3] T. C. E. Cheng, Z.-L. Chen and C.-L. Li, Parallel-machine scheduling with controllable processing times, *IIE Transactions*, vol.28, pp.177-180, 1996.
- [4] K. Chudzik, A. Janiak and M. Lichtenstein, Scheduling problems with resource allocation, in *Scheduling in Computer and Manufacturing Systems*, A. Janiak (ed.), Warszawa, Poland, 2006.
- [5] E. B. Edis, C. Oguz and I. Ozkarahan, Parallel machine scheduling with additional resources: Notation, classification, models and solution methods, *European Journal of Operational Research*, vol.230, pp.449-463, 2013.
- [6] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics*, vol.5, pp.287-326, 1979.
- [7] Y. He, M. Ji and T. C. E. Cheng, Single machine scheduling with a restricted rate-modifying activity, *Naval Research Logistics*, vol.52, pp.361-369, 2005.
- [8] C.-J. Hsu, T. C. E. Cheng and D.-L. Yang, Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time, *Information Sciences*, vol.181, pp.4799-4803, 2011.
- [9] A. Janiak, Minimization of the blooming mill standstills-mathematical model, suboptimal algorithms, *Mechanika*, vol.8, no.2, pp.37-49, 1989.
- [10] M. Ji and T. C. E. Cheng, Scheduling with job-dependent learning effects and multiple rate-modifying activities, *Information Processing Letters*, vol.110, pp.460-463, 2010.
- [11] C.-L. Lee and V.-J. Leon, Machine scheduling with a rate-modifying activity, *European Journal of Operational Research*, vol.128, pp.119-128, 2001.
- [12] H.-T. Lee and S.-J. Yang, Parallel machines scheduling with deterioration effects and resource allocations, *Journal of the Chinese Institute of Industrial Engineers*, vol.29, pp.534-543, 2012.
- [13] E. J. Lodree and C. D. Geiger, A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration, *European Journal of Operational Research*, vol.201, pp.644-648, 2010.
- [14] J. L. Mott, A. Kandel and T. P. Baker, *Discrete Mathematics for Computer Scientists & Mathematicians*, Prentice-Hall, New Jersey, 1986.
- [15] E. Nowicki and S. Zdrzalka, A survey of results for sequencing problems with controllable processing times, *Discrete Applied Mathematics*, vol.26, pp.271-287, 1990.

- [16] D. Oron, Scheduling controllable processing time jobs in a deteriorating environment, *Journal of the Operational Research Society*, vol.65, pp.49-56, 2014.
- [17] A. Rudek and R. Rudek, A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models, *Computers & Mathematics with Applications*, vol.62, pp.1870-1878, 2011.
- [18] K. Rustogi and V. A. Strusevich, Combining time and position dependent effects on a single machine subject to rate-modifying activities, *Omega*, vol.42, pp.166-178, 2014.
- [19] D. Shabtay and G. Steiner, A survey of scheduling with controllable processing times, *Discrete Applied Mathematics*, vol.155, pp.1643-1666, 2007.
- [20] L.-H. Su and C.-Y. Lien, Scheduling parallel machines with resource-dependent processing times, *International Journal of Production Economics*, vol.117, pp.256-266, 2009.
- [21] M. Trick, Scheduling multiple variable-speed machines, *Operations Research*, vol.42, pp.234-248, 1994.
- [22] L. N. Van Wassenhove and K. R. Baker, A bicriterion approach to time/cost trade-offs in sequencing, *European Journal of Operational Research*, vol.11, pp.48-52, 1982.
- [23] R. G. Vickson, Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine, *Operations Research*, vol.28, pp.1155-1167, 1980.
- [24] R. G. Vickson, Two single-machine sequencing problems involving controllable job processing times, *AIIE Transactions*, vol.12, pp.258-262, 1980.
- [25] 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.
- [26] 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.
- [27] J.-J. Wang, J.-B. Wang and F. Liu, Parallel machines scheduling with a deteriorating maintenance activity, *Journal of the Operational Research Society*, vol.62, pp.1898-1902, 2011.
- [28] X.-R. Wang and J.-J. Wang, Single-machine scheduling with convex resource dependent processing times and deteriorating jobs, *Applied Mathematical Modelling*, vol.37, pp.2388-2393, 2013.
- [29] C.-M. Wei, J.-B. Wang and P. Ji, Single-machine scheduling with time-and-resource-dependent processing times, *Applied Mathematical Modelling*, vol.62, pp.792-798, 2012.
- [30] D.-L. Yang, T. C. E. Cheng and S.-J. Yang, Parallel-machine scheduling with controllable processing times and rate-modifying activities to minimize total cost involving total completion time and job compressions, *International Journal of Production Research*, 2013.
- [31] D.-L. Yang and S.-J. Yang, Unrelated parallel-machine scheduling with multiple rate-modifying activities, *Information Sciences*, vol.235, pp.280-286, 2013.
- [32] S.-J. Yang, Parallel-machine scheduling with simultaneous considerations of position-dependent deterioration effects and maintenance activities, *Journal of the Chinese Institute of Industrial Engineers*, vol.28, pp.270-280, 2011.
- [33] S.-J. Yang, C.-J. Hsu and D.-L. Yang, Note on "Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time", *Information Sciences*, vol.260, pp.215-217, 2014.
- [34] Y. Yin, T. C. E. Cheng, D. Xu and C.-C. Wu, Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost, *Computers & Industrial Engineering*, vol.63, pp.223-234, 2012.
- [35] C.-L. Zhao, J.-Y. Guo and C.-J. Hsu, Parallel-machine scheduling with deteriorating jobs and rate-modifying activities to minimize total completion time, *Journal of the Chinese Institute of Industrial Engineers*, vol.28, pp.485-491, 2011.
- [36] C.-L. Zhao, H.-Y. Tang and C.-D. Cheng, Two-parallel machines scheduling with rate-modifying activities to minimize total completion time, *European Journal of Operational Research*, vol.198, pp.354-357, 2009.
- [37] C.-L. Zhao and H.-Y. Tang, A note to due-window assignment and single machine scheduling with deteriorating jobs and a rate-modifying activity, *Computers & Operations Research*, vol.39, pp.1300-1303, 2012.
- [38] Z. Zhu, F. Chu, L. Sun and M. Liu, Single machine scheduling with resource allocation and learning effect considering the rate-modifying activity, *Applied Mathematical Modelling*, vol.37, pp.5371-5380, 2013.
- [39] Z. Zhu, L. Sun, F. Chu and M. Liu, Single-machine group scheduling with resource allocation and learning effect, *Computers & Industrial Engineering*, vol.60, pp.148-157, 2011.