Scheduling jobs with resource-dependent ready times and processing times depending on their starting times and positions

Jian Jin[a], Ping Ji[b,1]

[a]School of Government, Beijing Normal University, Beijing 100875, China

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

Abstract

The paper deals with resource allocation scheduling problems in which the processing time of a job is defined by a function of its starting times and its position in a sequence. We also assume that the resource-dependent ready times of jobs are continuous functions of their consumed resource. Our objective is to minimize the resource consumption (the makespan) subject to the makespan constraint (limited resource availability). We prove that these two single-machine scheduling problems can be solved in polynomial time respectively.

Keywords: Scheduling; Resource allocation; Single-machine; Deteriorating job; Learning effect

1 Introduction

In modern scheduling problems and theory, we often encounter settings in which the processing times of jobs may be changed by the phenomenon of deterioration, i.e., any delay in processing a job is penalized by incurring additional time for accomplishing the job. We refer the reader to book Gawiejnowicz [1] for more details on single-machine, parallel-machine and dedicated-machine scheduling problems with deterioration jobs and time-dependent processing times. More recent review on scheduling problems with variable processing times is given in Agnetis et al. [2].

In addition, the scheduling problems and models with resource allocation have been studied by Blazewicz et al. [3], i.e., any job, besides machines, may require for its processing some additional resources. However, the phenomena of scheduling problems with deteriorating jobs and resource allocation can be found in real-life situations, i.e., job processing times are defined by functions of their starting times and positions in the sequence, and ready times of jobs are defined by functions of their consumed resource. "*The application of the model can be found in*

[1] Corresponding author. E-mail addresses: p.ji@polyu.edu.hk

the steel production, more precisely, in the process of preheating ingots by gas to prepare them for hot rolling on the blooming mill. Before the ingots can be hot rolled, they have to achieve the required temperature. However, the preheating time of the ingots depends on their starting temperature, i.e. the longer ingots wait for the start of the preheating process, the lower goes their temperature and therefore longer lasts the preheating process. The preheating time can be shortened by the increase of the gas flow intensity, i.e. the more gas is consumed, the shorter lasts the preheating process. Thus, the ingot preheating time depends on the starting moment of the preheating process and the amount of gas consumed during it" (Bachman and Janiak [4]). Bachman and Janiak [4] first studied single-machine resource allocation scheduling with deteriorating jobs. They showed that the maximum completion time (i.e., makespan) minimization scheduling problem is NP-hard. Zhao and Tang [5] studied scheduling with deteriorating jobs on a single-machine in which the release times of the jobs depend on the amounts of resource allocation. Zhu et al. [6] considered two single-machine scheduling problems with proportional-linear deterioration of job processing times. They proved that these both problems can be solved in polynomial time. Zhang et al. [7] addressed the resource allocation scheduling problems with learning effect and deteriorating jobs on a single machine. They presented the optimal algorithm for the problem of minimizing the makespan with the total resource consumption constraints. Wu et al. [8] considered the same model with Zhang et al. [7], they showed by a counter-example that the published result in Zhang et al. [7] is incorrect and provide the corrected result. Yang et al. [9], Wang and Wang [10], Li et al. [11], Wang and Wang [12], Liu et al. [13], and Wang et al. [14] considered other types of resource allocation scheduling problems. More recent review on parallel-machine scheduling problems related to the ones (i.e., resource allocation, learning effect and deteriorating jobs) are considered in Lu et al [15].

In this paper, we consider scheduling problems with deteriorating jobs and resource allocation. More precisely, a novel hybrid scheduling model is proposed, which simultaneously considers the features of processing times depending on their starting times and positions, and resource-dependent ready times. For two versions of the problem, we derive the structural properties on the schedule of jobs, based on which, a polynomial time algorithm can be proposed to solve the studied problem. The remaining part of this paper is organized as follows. In Section 2 we describe and formulate the problem. In Sections 3 and 4 we consider two resource-dependent scheduling problems and present polynomial time algorithms to solve them optimally. The last section includes conclusions.

2 Problem formulation

The single machine problem is to schedule n jobs $J_1, J_2, ..., J_n$ on one machine. The machine can handle one job at a time and preemption is not allowed. Associated with each job $J_j (j = 1, 2, ..., n)$ is a basic processing time p_j. Let $p_{j,r}(t)$ be the processing time of job J_j if it is started at time t and scheduled in position r in a sequence. In this paper we consider the combination of models of $p_{j,r} = p_j f(r)$ and $p_j(t) = p_j(A \pm Bt) = p_j \times (A \pm Bt)$ (Kononov [16] and Wang and Xia [17]), that is

$$p_{j,r}(t) = p_j(t)f(r) = p_j(A \pm Bt)f(r) = p_j \times (A \pm Bt) \times f(r), \quad (1)$$

where $p_j \geq 0$, $A > 0$, $B \geq 0$, $f(r)$ is a non-decreasing function, and $f(1) = 1$. Mosheiov [18] proposed the model $p_{j,r} = p_j f(r)$, where $f(r) = r^a, a \geq 0$. For the case of $p_{j,r}(t) = p_j(A + Bt)f(r)$, we don't have any extra restrictions. However, for the case of $p_{j,r}(t) = p_j(A - Bt)f(r)$, we assume that the basic processing times satisfy the following condition:

$$0 < p_j B f(r) < 1 \quad (2)$$

and

$$0 < Bf(r)(T_0 + \sum_{i=1}^{n} p_i - p_j) < 1, \quad (3)$$

hold for $1 \leq j \leq n$ and $1 \leq r \leq n$, where $T_0 \geq 0$ is the starting time of the first job. As in Ho et al. [19], Wang [20], and Gawiejnowicz [1], the condition (2) ensures that the decrease of each job processing time is less than one unit for every unit delay in its starting moment, and the condition (3) ensures that all job processing times are positive in a feasible schedule (see also Ho et al. [19], Wang [20], and Gawiejnowicz [1] for detailed explanations). We assume that the release time of job J_j is: $r_j = g(u_j) \geq T_0, \underline{u} \leq u_j \leq \overline{u}$, $j = 1, 2, \ldots, n$, where u_j is the amount of a doubly-constrained non-renewable resource allocated to job J_j, with $0 \leq \underline{u} \leq u_j \leq \overline{u}$, $\underline{u}$ and $\overline{u}$ are known technological constraints and $g : R_+ \to R_+$ (R_+ is the set of non-negative real numbers) is a strictly decreasing continuous function with $g(\overline{u}) \geq 0$, g also is an invertible function.

Let π denote any schedule of the jobs $J_1, J_2, \ldots, J_n$ and the job $J_{[j]}$ be in position j of π. We denote the set of all feasible schedules by Π. By $\tilde{U}$ we denote the set of all possible resource allocations $u = (u_1, u_2, \ldots, u_n)$ satisfying the resource constraint (i.e., $\underline{u} \leq u_j \leq \overline{u}, j = 1, 2, \ldots, n, \sum_{j=1}^{n} u_j \leq \hat{U}$, where $\hat{U}$ is the total amount of the resource available, and $\hat{U} \geq n\underline{u}$).

For a given sequence $\pi = (J_1, J_2, \ldots, J_n)$, $C_j = C_j(\pi)$ represents the completion time for job J_j. We consider the problem of minimizing the total resource consumption with a makespan constraint and the problem of minimizing the makespan with a resource consumption constraint. In the three-field notation scheme $\alpha|\beta|\gamma$ introduced by Graham et al. [21], the two problems are denoted as $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \underline{u} \leq u_j \leq \overline{u}, C_{\max} \leq \hat{C}|\sum u_j$ and $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \sum_{j=1}^{n} u_j \leq \hat{U}, \underline{u} \leq u_j \leq \overline{u}|C_{\max}$, respectively. Zhao and Tang [5] considered the scheduling problems $1|p_{j,r}(t) = p_j r^a, r_j = g(u_j), \underline{u} \leq u_j \leq \overline{u}, C_{\max} \leq \hat{C}|\sum u_j$ and $1|p_{j,r}(t) = p_j r^a, r_j = g(u_j), \sum_{j=1}^{n} u_j \leq \hat{U}, \underline{u} \leq u_j \leq \overline{u}|C_{\max}$. Zhu et al. [6] considered the scheduling problems $1|p_{j,r}(t) = p_j(A + Bt), r_j = g(u_j), \underline{u} \leq u_j \leq \overline{u}, C_{\max} \leq \hat{C}|\sum u_j$ and $1|p_{j,r}(t) = p_j(A + Bt), r_j = g(u_j), \sum_{j=1}^{n} u_j \leq \hat{U}, \underline{u} \leq u_j \leq \overline{u}|C_{\max}$.

Remark: For the case of $p_{j,r}(t) = p_j(A - Bt)f(r)$, the following example is proposed by an anonymous reviewer: there is only one job J_j with basic processing time $p_j = 1$, $A = T_0/[2*(T_0+1)]$, $B = 1/(T_0+1)$ and $f(r) = r$, where $T_0 > 0$. The problem is to minimize the max completion time with resource consumption constraint such that $r_j = g(1)$ where $g(x) = T_0/x$. Since there is only one job, $f(r) = f(1) = 1$ for job J_j, we have $p_{j,r}(t) = p_{j,1}(t) = p_j * (A - Bt) * f(1) = p_j * (A - Bt) = p_j * (T_0/[2 * (T_0 + 1)] - t/(T_0 + 1)) \leq p_j * (T_0/[2 * (T_0 + 1)] - T_0/(T_0 + 1)) < 0$, which shows the actual processing time of J_j is negative. In addition, the setting satisfies the two assumptions, condition (2) and (3), i.e., $0 < p_j Bf(r) = 1/(T_0 + 1) < 1$ and $0 < Bf(r)(T_0 + \sum_{i=1}^{n} p_i - p_j) = Bf(r)T_0 = BT_0 = T_0/(T_0 + 1) < 1$. As in Wang and Xia [17], in order to avoid the emergence of negative actual processing time, we assume that $A = 1$ for the case of $p_{j,r}(t) = p_j(A - Bt)f(r)$.

3 Minimizing the resource consumption

We first give two lemmas, which will be applied further on.

Lemma 1 *For a given schedule $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$ of $1|p_{j,r}(t) = p_j(A \pm Bt)f(r)|\gamma$, if the first job starts at time $T_0 \geq 0$, then*

$$C_{[j]} = (T_0 \pm \frac{A}{B}) \prod_{i=1}^{j} (1 \pm B p_{[i]} f(i)) \mp \frac{A}{B}, j = 1, 2, \ldots, n. \quad (4)$$

Proof: By setting $b_{[j]} = p_{[j]} f(j)$ we get a statement of Theorem 6.20 (i.e., If there hold inequalities $BT_0 + A > 0$ and $1 + p_j B > 0$ for $1 \leq j \leq n$, then the problem $1|p_j(t) = p_j(A + Bt)|C_{\max}$ is solvable in $O(n)$ time, the maximum completion time

$$C_{\max} = \begin{cases} T_0 + A \sum_{i=1}^{n} p_{[i]} & \text{if} \quad B = 0, \\ (T_0 + \frac{A}{B}) \prod_{i=1}^{n} (1 + B p_{[i]}) - \frac{A}{B} & \text{if} \quad B \neq 0, \end{cases}$$

and it does not depend on the schedule of jobs) from the book by Gawiejnovicz [1] and the paper by Kononov [16]. □

Lemma 2 *For the problem* $1|p_{j,r}(t) = p_j(A \pm Bt)f(r)|C_{\max}$*, an optimal schedule can be obtained by the LPT rule, i.e., by sequencing the jobs in non-increasing order of* p_j*.*

Proof. By using job interchanging technique, let π and π' be two job schedules where the difference between π and π' is a pairwise interchange of two adjacent jobs J_i and J_j, i.e., $\pi = (S_1, J_i, J_j, S_2)$ and $\pi' = (S_1, J_j, J_i, S_2)$, where S_1 and S_2 are partial sequences, such that $p_i \geq p_j$. Let t denote the completion time of the last job in S_1. Furthermore, we assume that that there are $r-1$ jobs in S_1. Thus, J_i and J_j are the rth and the $(r+1)$th jobs, respectively, in π. To show π dominates π', it suffices to show that $C_j(\pi) \leq C_i(\pi')$. From Lemma 1, the completion times of jobs J_i and J_j in π are

$$C_i(\pi) = (t \pm \frac{A}{B})(1 \pm Bp_i f(r)) \mp \frac{A}{B}$$

and

$$C_j(\pi) = (t \pm \frac{A}{B})(1 \pm Bp_i f(r))(1 \pm Bp_j f(r+1)) \mp \frac{A}{B}. \quad (5)$$

Similarly, the completion times of jobs J_j and J_i in π' are

$$C_j(\pi') = (t \pm \frac{A}{B})(1 \pm Bp_j f(r)) \mp \frac{A}{B}$$

and

$$C_i(\pi') = (t \pm \frac{A}{B})(1 \pm Bp_j f(r))(1 \pm Bp_i f(r+1)) \mp \frac{A}{B}. \quad (6)$$

Based on equations (5) and (6), we have

$$C_j(\pi) - C_i(\pi') = (p_i - p_j)(A \pm Bt)(f(r) - f(r+1)) \leq 0$$

for $p_i \geq p_j$ and $f(r) \leq f(r+1)$, thus π dominates π'. □

Given π and $u \in \tilde{U}$, we have

$$r_{[1]} = g(u_{[1]}),$$

$$C_{[1]}(\pi, u) = (r_{[1]} \pm \frac{A}{B})(1 \pm Bp_{[1]}f(1)) \mp \frac{A}{B},$$

$$r_{[j]} = g(u_{[j]}),$$

$$C_{[j]}(\pi, u) = \max_{1 \leq i \leq j} \left\{ \left(\max\{r_{[i]}, C_{[i-1]}\} \pm \frac{A}{B} \right) \prod_{k=i}^{j} (1 \pm Bp_{[k]} f(k)) \right\} \mp \frac{A}{B}, j = 2, \ldots, n.$$

Thus

$$C_{\max}(\pi, u) = \max_{1 \le i \le n} \left\{ \left(\max\{r_{[i]}, C_{[i-1]}\} \pm \frac{A}{B} \right) \prod_{k=i}^{n} (1 \pm Bp_{[k]} f(k)) \right\} \mp \frac{A}{B}.$$

Also, for $\pi \in \Pi$ and $u \in \tilde{U}$,

$$U(\pi, u) = \sum_{j=1}^{n} u_{[j]}.$$

Let $\hat{C}$ be a given makespan, our goal is to find $\pi^* \in \Pi$ and $u^* \in \tilde{U}$ that minimizes $\sum_{j=1}^{n} u_j$, subject to $C_{\max}(\pi^*, u^*) \le \hat{C}$. Since releasing jobs sooner consumes more resources, hence the completion time of the last job is $\hat{C}$. Consider the permutation $\pi = (J_1, J_2, \ldots, J_n)$ of jobs in non-increasing order of processing time.

So, a sequence π is feasible only if

$$C_{\max} = (g(\overline{u}) \pm \frac{A}{B}) \prod_{i=1}^{n} (1 \pm Bp_i f(i)) \mp \frac{A}{B} \le \hat{C}.$$

For $\pi = (J_1, J_2, \ldots, J_n)$, we denote by u^* the resource allocation, our problem is as follows:

$$U(\pi, u^*) = \min_{u \in \tilde{U}} \{U(\pi, u)\},$$

subject to $C_{\max}(\pi, u^*) \le \hat{C}$.

If $C_{\max}(\pi, \cdot) = \hat{C}$, then the first job starts at time $T = (\hat{C} \pm \frac{A}{B}) / \prod_{i=1}^{n} (1 \pm Bp_{[i]} f(i)) \mp \frac{A}{B}$, and the optimal resource allocation can be obtained by the following algorithm:

Algorithm 1.

1) For a given schedule $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, calculate $T = (\hat{C} \pm \frac{A}{B}) / \prod_{i=1}^{n} (1 \pm Bp_{[i]} f(i)) \mp \frac{A}{B}$.

2) If $T \ge g(\underline{u})$, set $u^*_{[j]} = \underline{u}, j = 1, \ldots, n$, stop. Otherwise, goto 3).

3) If $(T \pm \frac{A}{B})(1 \pm Bp_{[1]} f(1)) \mp \frac{A}{B} \ge g(\underline{u})$, set $u^*_{[1]} = g^{-1}(T), u^*_{[j]} = \underline{u}, j = 2, \ldots, n$, stop. Otherwise, let k be the maximum natural number satisfying $(T \pm \frac{A}{B}) \prod_{i=1}^{k-1} (1 \pm Bp_{[i]} f(i)) \mp \frac{A}{B} \le g(\underline{u})$, then

$$\begin{aligned} u^*_{[1]} &= g^{-1}(T), \\ u^*_{[j]} &= g^{-1} \left((T \pm \frac{A}{B}) \prod_{i=1}^{j-1} (1 \pm Bp_{[i]} f(i)) \mp \frac{A}{B} \right), j = 2, \ldots, k, \\ u^*_{[j]} &= \underline{u}, j = k+1, \ldots, n. \end{aligned}$$

Theorem 1 *For* $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \underline{u} \leq u_j \leq \overline{u}, C_{\max} \leq \hat{C}|\sum u_j$, *an optimal schedule* π^* *can be obtained by the LPT rule, and an optimal resource allocation* u^* *can be obtained by Algorithm 1.*

Proof. (i) Consider first the case of $p_{j,r}(t) = p_j(A+Bt)f(r)$. Let's prove that in optimal schedule jobs should be released in non-increasing order of processing time.

Consider an optimal sequence π in which there are two adjacent jobs J_j and J_k with J_j being followed by J_k and $p_j < p_k$. We assume that J_j and J_k are the rth and the $(r+1)$th jobs, respectively, in π. Let the completion time of job J_k be C_0. Performing an adjacent pair-wise interchange on jobs J_j and J_k to get another schedule π', then

$$C_k(\pi) = C_0, r_k(\pi) = (C_0+\frac{A}{B})/(1+Bp_kf(r+1))-\frac{A}{B}, r_j(\pi) = (C_0+\frac{A}{B})/[(1+Bp_jf(r))(1+Bp_kf(r+1))]-\frac{A}{B};$$

$$C_j(\pi') = C_0, r_j(\pi') = (C_0+\frac{A}{B})/(1+Bp_jf(r+1))-\frac{A}{B}, r_k(\pi') = (C_0+\frac{A}{B})/[(1+Bp_kf(r))(1+Bp_jf(r+1))]-\frac{A}{B}.$$

Since $p_j < p_k$, $f(r) \leq f(r+1)$ and $(1+Bp_jf(r))(1+Bp_kf(r+1)) \geq (1+Bp_kf(r))(1+Bp_jf(r+1))$, we have $r_j(\pi') > r_k(\pi), r_k(\pi') \geq r_j(\pi)$; hence $g^{-1}(r_j(\pi')) + g^{-1}(r_k(\pi')) < g^{-1}(r_k(\pi)) + g^{-1}(r_j(\pi))$.

(ii) For the case of $p_{j,r}(t) = p_j(A - Bt)f(r)$, similarly to the case (i), we have

$$C_k(\pi) = C_0, r_k(\pi) = (C_0-\frac{A}{B})/(1-Bp_kf(r+1))+\frac{A}{B}, r_j(\pi) = (C_0-\frac{A}{B})/[(1-Bp_jf(r))(1-Bp_kf(r+1))]+\frac{A}{B};$$

$$C_j(\pi') = C_0, r_j(\pi') = (C_0-\frac{A}{B})/(1-Bp_jf(r+1))+\frac{A}{B}, r_k(\pi') = (C_0-\frac{A}{B})/[(1-Bp_kf(r))(1-Bp_jf(r+1))]+\frac{A}{B}.$$

Since $p_j < p_k$, $f(r) \leq f(r+1)$, $(C_0 - \frac{A}{B}) < 0$ and $(1 - Bp_jf(r))(1 - Bp_kf(r+1)) \leq (1 - Bp_kf(r))(1 - Bp_jf(r+1))$, we have $r_j(\pi') > r_k(\pi), r_k(\pi') \geq r_j(\pi)$; hence $g^{-1}(r_j(\pi')) + g^{-1}(r_k(\pi')) < g^{-1}(r_k(\pi)) + g^{-1}(r_j(\pi))$.

The release times of the jobs processed after jobs J_j and J_k are not affected by the interchange, and the release times of the jobs processed before jobs J_j and J_k are unchanged. Hence the value of the objective function under π' is strictly less than under π. This contradicts the optimality of π and proves the theorem. $\square$

If the values of f, g and g^{-1} in Algorithm 1 can be calculated in $O(h(n))$ time respectively, and $h(n)$ is a polynomial in n, then (π^*, u^*) can be calculated in polynomial time $O(\max\{h(n), n \log n\})$.

4 Minimizing the makespan

From Lemma 2 and the results of Section 3, for $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \sum_{j=1}^{n} u_j \leq \hat{U}, \underline{u} \leq u_j \leq \overline{u}|C_{\max}$ problem, we also only need to consider the schedule in which the jobs are processed by the LPT rule.

Without loss of generality, we assume that $\pi = (J_1, J_2, \ldots, J_n)$ is the sequence by the LPT rule.

First we prove that only schedules without idle time need to be considered.

Lemma 3 *There exists an optimal schedule for the problem* $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \sum_{j=1}^{n} u_j \leq \hat{U}, \underline{u} \leq u_j \leq \overline{u}|C_{\max}$*, in which the machine is not idle between the processing of jobs.*

Proof. Assume that we have an optimal schedule where the machine is idle between the processing of jobs J_i and J_j. Thus, we can move job J_i and the jobs scheduled before J_i to the right until job J_j without increasing the value $C_{\max}$ and the resource consumption amount u_i. □

Under the condition $u_j^* = \underline{u}, j = 1, 2, \ldots, n$, then $C_{\max}(\pi, \underline{u}) = (g(\underline{u}) \pm \frac{A}{B}) \prod_{i=1}^{n}(1 \pm Bp_i f(i)) \mp \frac{A}{B}$. If we increase the amount of resources allocated to J_1, then the release time r_1 will be smaller and the makespan will be smaller, too. Let the maximum amount of resources allocated to job J_1 be $u_1^{\max}$, i.e., $u_1^{\max} = \min\{U - (n-1)\underline{u}, \overline{u}\}$.

Let $\bar{r}_1 = g(u_1^{\max})$, and the completion time of J_1 is $(\bar{r}_1 \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B}$.

If $(\bar{r}_1 \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B} \geq g(\underline{u})$, we have

$$u_1^* = u_1^{\max},$$
$$u_j^* = \underline{u}, j = 2, \ldots, n,$$
$$C_{\max} = (\bar{r}_1 \pm \frac{A}{B}) \prod_{i=1}^{n}(1 \pm Bp_i f(i)) \mp \frac{A}{B}.$$

If $(\bar{r}_1 \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B} < g(\underline{u})$, then there must be a natural number k such that

$$C_j(\pi, u_\pi^*) \leq g(\underline{u}), j = 1, 2, \ldots, k-1$$
$$C_j(\pi, u_\pi^*) > g(\underline{u}), j = k, \ldots, n,$$
$$u_j^* = \underline{u}, j = k+1, \ldots, n,$$

where u_π^* is the optimal resource allocation for the sequence π.

If there is a k such that $r_k = g(\underline{u})$, suppose that the release time of J_j is r_j, $j = 1, 2, \ldots, k$, then

$$r_1 = (g(\underline{u}) \pm \frac{A}{B}) / \prod_{i=1}^{k-1}(1 \pm Bp_i f(i)) \mp \frac{A}{B},$$
$$r_2 = (r_1 \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B},$$
$$\ldots,$$
$$r_{k-1} = (r_1 \pm \frac{A}{B}) \prod_{i=1}^{k-2}(1 \pm Bp_i f(i)) \mp \frac{A}{B},$$
$$r_k = (r_1 \pm \frac{A}{B}) \prod_{i=1}^{k-1}(1 \pm Bp_i f(i)) \mp \frac{A}{B} = g(\underline{u}), \tag{7}$$
$$u_j = g^{-1}(r_j), j = 1, 2, \ldots, k,$$
$$u_j = \underline{u}, j = k+1, \ldots, n,$$
$$\sum_{j=1}^{k-1} u_j + (n-(k-1))\underline{u} \leq \hat{U}. \tag{8}$$

Let $D = g(\underline{u}) - C_{k-1}(\pi, u_\pi^*)$. From (4)

$$r_1^* = \left(g(\underline{u}) - D \pm \frac{A}{B}\right) / \prod_{i=1}^{k-1}(1 \pm Bp_i f(i)) \mp \frac{A}{B},$$
$$r_2^* = (r_1^* \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B},$$
$$\ldots,$$
$$r_{k-1}^* = (r_1^* \pm \frac{A}{B}) \prod_{i=1}^{k-2}(1 \pm Bp_i f(i)) \mp \frac{A}{B},$$
$$r_k^* = (r_1^* \pm \frac{A}{B}) \prod_{i=1}^{k-1}(1 \pm Bp_i f(i)) \mp \frac{A}{B} = g(\underline{u}) - D, \tag{9}$$
$$u_j^* = g^{-1}(r_j^*), j = 1, 2, \ldots, k,$$
$$u_j^* = \underline{u}, j = k+1, \ldots, n. \tag{10}$$
$$\sum_{j=1}^{k} u_j^* + (n-k)\underline{u} = \hat{U}, \tag{11}$$

where D can be determined by Eqs. (9), (10) and (11), r_j^* is the optimal ready time of job J_j.

Remark. If k can be determined, from Eq. (9), $u_j^* = g^{-1}(r_j^*)(j = 1, 2, \ldots, k)$ can be calculated, and u_j^* is a function of D, hence the Eq. (11) has a solution and this solution is the only one.

If k is the maximum natural number satisfying the following inequality: $\sum_{j=1}^{k-1} u_j + (n-(k-1))\underline{u} \le \hat{U}$, then $D = g(\underline{u}) - C_{k-1}(\pi, u_\pi^*)$.

Based on the above analysis, the optimal resource allocation u_j^* of $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \sum_{j=1}^n u_j \le \hat{U}, \underline{u} \le u_j \le \overline{u}|C_{\max}$ can be obtained by the following algorithm:

Algorithm 2.

1) Sequencing the jobs by the LPT rule, and let $u_1^{\max} = \min\{\hat{U} - (n-1)\underline{u}, \overline{u}\}$, $\bar{r}_1 = g(u_1^{\max})$.

2) If $(\bar{r}_1 \pm \frac{A}{B})(1 \pm Bp_1 f(1)) \mp \frac{A}{B} \ge g(\underline{u})$, then $u_1^* = u_1^{\max}, u_j^* = \underline{u}, j = 2, \ldots, n$, and stop. Otherwise, goto 3).

3) Let k be the maximum natural number satisfying the following inequality:

$$\sum_{j=1}^{k-1} u_j + (n-(k-1))\underline{u} \le \hat{U},$$

where $u_j = g^{-1}(r_j)$, and r_j satisfies form (7).

4) The resource allocation is

$$u_j^* = g^{-1}(r_j^*), j = 1, 2, \ldots, k,$$
$$u_j^* = \underline{u}, j = k+1, \ldots, n,$$

where r_j^* can be determined by form (9) and D satisfies Eq. (11).

Theorem 2 *For $1|p_{j,r}(t) = p_j(A \pm Bt)f(r), r_j = g(u_j), \sum_{j=1}^n u_j \le \hat{U}, \underline{u} \le u_j \le \overline{u}|C_{\max}$ problem, an optimal schedule π^* and an optimal resource allocation u^* can be obtained by Algorithm 2.*

If we are able to calculate all the values of f, g, g^{-1} and D in Algorithm 2 in $O(h(n))$ time, then finding π^* and u^* needs $O(\max\{h(n), n\log n\})$ time.

Example 1. $n = 4, p_1 = 4, p_2 = 3, p_3 = 2, p_4 = 1, A = B = 1, f(r) = r^a, a = 0.2, \underline{u} = 1, \overline{u} = 20, \hat{U} = 24$ and $r_j = g(u_j) = \frac{28}{u_j}$.

According to the result of Algorithm 2, $g(\underline{u}) = 28, u_1^{\max} = \min\{\hat{U} - (n-1)\underline{u}, \overline{u}\} = 20$, $\bar{r}_1 = g(u_1^{\max}) = \frac{7}{5}, (\bar{r}_1 + \frac{A}{B})(1 + Bp_1 f(1)) - \frac{A}{B} = 11 < g(\underline{u}) = 28$. Then, we solve $k-1$:

$k-1 = 1, r_1 = \frac{24}{5}, u_1 = \frac{35}{6}, \sum_{j=1}^{k-1} u_{\pi(j)} + (n-(k-1))\underline{u} = \frac{53}{6} \le \hat{U}$;

$k-1 = 2, r_1 = 0.3045, u_1 = 91.9540, r_2 = 5.5225, u_2 = 5.0702$, $\sum_{j=1}^{k-1} u_{\pi(j)} + (n-(k-1))\underline{u} = 99.0242 > \hat{U}$;

So $k-1 = 1$, i.e., $k = 2$. Furthermore, by Eq. (11) we solve D:

$r_1^* = \frac{24-D}{5}, u_1^* = \frac{140}{24-D}; r_2^* = 28 - D, u_2^* = \frac{28}{28-D}; u_3^* = u_4^* = 1$, then, $D = 16.8190$, $u_1^* = 19.4959, u_2^* = 2.5042, u_3^* = 1, u_4^* = 1$ and $C_{\max}\{r_1^*|(J_1, J_2, J_3, J_4)\} = 437.5962$.

5 Conclusions

This paper studied scheduling problems with time-and-position-dependent deterioration on a single-machine. For two resource allocation scheduling problems, we showed that these problems can polynomially solvable if f, g and g^{-1} can be calculated in polynomial time. It is known that minimizing max completion time for the general problem is NP-hard. We conjecture that minimizing resource consumption for the general problem is also NP-hard. Future research may focus on considering these two general problems, addressing flow shop (or unrelated parallel machine) scheduling problems with processing times depending on their starting times, positions and resource, studying multiagent scheduling or group technology scheduling.

Acknowledgements–This research was supported by a grant from the Research Grants Council of Hong Kong Special Administrative Region, China (PolyU 517011).

References

[1] Gawiejnowicz, S. (2008) *Time-dependent scheduling.* Springer, Berlin, Heidelberg.

[2] Agnetis, A., Billaut, J.-C., Gawiejnowicz, S., Pacciarelli, D. and Soukhal, A. (2014) *Multiagent scheduling.* Springer, Berlin, Heidelberg.

[3] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G. and Weglarz, J. (2001) *Scheduling computer and Manufacturing Processes.* Springer, Berlin, Heidelberg.

[4] Bachman, A. and Janiak, A. (2000) Scheduling deteriorating jobs dependent on resources for the makespan minimization. Operations Research Proceedings 2000, Dresden, September, pp. 29-34. Springer, Berlin, Heidelberg.

[5] Zhao, C.-L. and Tang H.-Y. (2005) Single machine scheduling problems with deteriorating jobs. Applied Mathematics and Computation, **161**, 865-874.

[6] Zhu, V.C.Y., Sun, L.Y., Sun, L.H. and Li, X.H. (2010) Single-machine scheduling time-dependent jobs with resource-dependent ready times. Computers & Industrial Engineering, **58**, 84-87.

[7] Zhang, X.-G., Yan, G.-L. and Tang, G.-C. (2011) Single-machine scheduling problems with release time of jobs depending on resource allocated. International Journal of Advanced Manufacturing Technology, **57**, 1175-1181.

[8] Wu, Y.-B., Piao, Z.-Y., Liu, H. and Ji, P. (2013) A note on "single-machine scheduling problems with release time of jobs depending on resource allocated". International Journal of Advanced Manufacturing Technology, **65**, 689-693.

[9] Yang, D.-L., Lai, C.-J. and Yang, S.-J. (2014) Scheduling problems with multiple common due windows assignment and controllable processing times on a single machine. International Journal of Production Economics, **150**, 96-103.

[10] Wang, J.-B. and Wang, M.-Z. (2014) Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times. Asia-Pacific Journal of Operational Research, **31(5)**, 1450036 (15 pages).

[11] Li, G., Luo, M.-L., Zhang, W.-J. and Wang, X.-Y. (2015) Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation. International Journal of Production Research, **53(4)**, 1228-1241.

[12] Wang, J.-B. and Wang, J.-J. (2015) Research on scheduling with job-dependent learning effect and convex resource dependent processing times. International Journal of Production Research, **53**, 5826-5836.

[13] Liu, L., Wang, J.-J. and Wang, X.-Y. (2016) Due-window assignment scheduling with resource processing times to minimise total resource consumption cost. International Journal of Production Research, **54**, 1186-1195.

[14] Wang, J.-B., Liu, M., Yin, N. and Ji, P. (2017) Scheduling jobs with controllable processing time, truncated job-dependent learning and deterioration effects. Journal of Industrial and Management Optimization, **13(2)**, 1025-1039.

[15] Lu, Y.-Y., Jin, J., Ji, P. and Wang, J.-B. (2016) Resource-dependent scheduling with deteriorating jobs and learning effect on unrelated parallel machines. Neural Computing & Applications, **27**, 1993-2000.

[16] Kononov, A. (1998) A single machine scheduling problems with processing times proportional to an arbitrary function. Discrete Analysis and Operations Research, **5**, 17-37 (in Russian).

[17] Wang, J.-B. and Xia, Z.-Q. (2005) Scheduling jobs under decreasing linear deterioration. Information Processing Letters, **94,** 63-69.

[18] Mosheiov, G. (2001) Parallel machine scheduling with a learning effect. Journal of the Operational Research Society, **52,** 1165-1169.

[19] Ho, K.I.-J., Leung, J.Y.-T. and Wei, W.-D. (1993) Complexity of scheduling tasks with time-dependent execution times. Information Processing Letters, **48**, 315-320.

[20] Wang, J.-B. (2007) Flow shop scheduling problems with decreasing linear deterioration under dominant machines. Computers & Operations Research, **34**, 2043-2058.

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