Theoretical Computer Science 462 (2012) 80–87

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Scheduling of deteriorating jobs with release dates to minimize the maximum lateness

Cuixia Miao [a,*], Yuzhong Zhang [b], Cuilian Wu [b]

[a] School of Mathematical Sciences, Qufu Normal University, Qufu, Shandong, 273165, People's Republic of China
[b] School of Management Sciences, Qufu Normal University, Rizhao, Shandong, 276826, People's Republic of China

ARTICLE INFO

Article history:
Received 7 April 2012
Received in revised form 23 July 2012
Accepted 28 August 2012
Communicated by D.-Z. Du

Keywords:
Scheduling
Deteriorating job
NP-hard
Maximum lateness

ABSTRACT

In this paper, we consider the problem of scheduling n deteriorating jobs with release dates on a single (batching) machine. Each job's processing time is a simple linear function of its starting time. The objective is to minimize the maximum lateness. When the machine can process only one job at a time, we first show that the problem is NP-hard even if there are only two distinct release dates. Then we present a 2-approximation algorithm for the case where all jobs have negative due dates. Furthermore, we prove that the earliest due date (EDD) rule provides an optimal solution to the case where all jobs have agreeable release dates, due dates and deteriorating rates, and that the EDD rule gives the worst order for the general case, respectively. When the machine can process up to $b(b = \infty)$ jobs simultaneously as a batch, i.e., the unbounded parallel-batch scheduling model, we show that the problem is NP-hard and present one property of the optimal schedule for the case where all jobs have agreeable release dates and due dates.

1. Introduction

Traditional scheduling problems assume that the processing times of jobs are constant. However, in the real world, the processing times may change. Examples can be found in steel production and fire fighting, amongst others, where any delay in processing a task may increase its completion time. The reader is referred to Kunnathur and Gupta [13] and Sundararaghavan and Kunnathur [24] for other examples. This scheduling has been extensively studied in the last decade (see [19,20,14,4,6,11,12], . . .).

Brucker et al. [1] defined that a parallel-batch machine is a machine that can process up to b jobs simultaneously as a batch; the processing time of the batch is equal to the longest time of any job in the batch. All jobs contained in the same batch start and complete at the same time. Once processing of a batch is initiated, it cannot be interrupted and other jobs cannot be introduced into the batch until processing is completed. Parallel-batch scheduling is motivated by burn-in operations in semiconductor manufacturing, and it has two distinct models: the *bounded model*, in which the bound b for each batch size is effective, i.e., $b < n$, and the *unbounded model*, in which there is effectively no limit on the size of batch, i.e., $b = \infty$ or $b \geq n$, where n denotes the number of jobs and b denotes the batch capacity. This processing system has been extensively studied in the last decade (see [15,25,22,26,16,17,7], amongst others). All the above-mentioned results concerning parallel-batch scheduling assume that the processing times are constant.

But job deterioration and parallel-batch processing coexist in many realistic scheduling situations. Examples can be found in steel production. Qi et al. [23] considered the unbounded parallel-batch scheduling problem with deteriorating jobs on a single machine. Li et al. [18] and Miao et al. [21] also considered the parallel-batch scheduling of deteriorating jobs.

* Corresponding author. Tel.: +86 15589179059.
E-mail address: miaocuixia@126.com (C. Miao).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.08.022

In this paper, we study the scheduling of deteriorating jobs with release dates on a single machine and on a single batching machine. The objective is to minimize the maximum lateness.

1.1. Problem description

The problem considered in this paper can be formally described as follows. We are given a set of independent and non-preemptively deteriorating jobs $J = \{J_1, \ldots, J_n\}$ that are to be scheduled on a single machine or an unbounded parallel-batch machine. Each job J_j $(j = 1, \ldots, n)$ has a release date r_j, a deteriorating rate $\alpha_j > 0$ and a due date d_j. We assume, as Mosheiov [19], that the actual processing time of job J_j is $p_j = \alpha_j t$, where $t \geq r_j$ is the starting time of J_j. Here, we assume $\min\{r_j | j = 1, \ldots, n\} = t_0 > 0$ (when $t_0 = 0$, the completion time of each job will be 0). The objective is to minimize the maximum lateness $L_{\max} = \max\{L_j | j = 1, \ldots, n\} = \max\{C_j - d_j | j = 1, \ldots, n\}$, where C_j denotes the completion time of job J_j. For a given batch B, we denote its deteriorating rate and release date by $\alpha(B)$ and $r(B)$, respectively, and $C(B)$ denotes the completion time. Then $\alpha(B) = \max\{\alpha_j | J_j \in B\}$, $r(B) = \max\{r_j | J_j \in B\}$, and $C(B) = (1 + \alpha(B))S(B)$, where $S(B)$ is the starting time of batch B. Using the 3-field notation of Graham et al. [9], we denote our problems as $1|r_j, p_j = \alpha_j t|L_{\max}$ and $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$.

1.2. Relevant previous work

Mosheiov [19] proved that the earliest due date (EDD) rule provides the optimal schedule for problem $1|p_j = \alpha_j t|L_{\max}$, and Bachman and Janiak [2] proved that the EDD rule also provides the optimal schedule for problem $1|p_j = a_j + ka_j t|L_{\max}$. Bachman and Janiak [3] showed that the problem $1|p_j = a_j + \alpha_j t|L_{\max}$ is NP-hard, and Hsu and Lin [10] designed a branch-and-bound algorithm for this problem. Cheng and Ding [5] designed an $O(n^6 logn)$ algorithm for the special problem $1|p_j = a_j + \alpha t|L_{\max}$. Note that in these papers the jobs have identical release dates. Apart from these five publications, we do not know about any others in which this scheduling model for the minimizing maximum lateness of deteriorating jobs has been considered.

For the unbounded parallel-batch scheduling of minimizing the maximum lateness, Brucker et al. [1] designed an $O(n^2)$ dynamic programming algorithm for problem $1|p - batch, b = \infty|L_{\max}$; Cheng and Ding [6] proved that the parallel-batch problem $1|p - batch, r_j, b = \infty|L_{\max}$ is NP-hard, and they also considered some special cases. Note that the processing time of each job is fixed in their problems.

Qi et al. [23] gave an $O(n^3)$ dynamic programming algorithm for $1|p - batch, r_j = t_0, p_j = \alpha_j t, b = \infty|L_{\max} \leq 0$ in which the jobs have identical release dates. Until now, this is the only paper considering the minimization problem of maximum lateness for deteriorating jobs on a parallel-batch machine.

To the best of our knowledge, minimization problems of maximum lateness for deteriorating jobs with release dates on a single machine and on a single batching machine have never been discussed.

1.3. Organization of the paper

In Section 2, we prove that the problem $1|r_j, p_j = \alpha_j t|L_{\max}$ is NP-hard even if there are only two distinct release dates, and give a 2-approximation algorithm for the case where all jobs have negative due dates, and we also discuss the EDD rule. In Section 3, we prove that the parallel-batch scheduling problem $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$ is NP-hard and present one property of the optimal schedule for one special case. We conclude the paper and suggest some interesting topics for future research in Section 4.

2. Problem $1|r_j, p_j = \alpha_j t|L_{\max}$

2.1. NP-hardness proof

In this subsection, the first result derived is presented. The following lemma will be used in what follows.

Lemma 2.1.1 ([19]). *For the single machine scheduling problem $1|p_j = \alpha_j t|C_{\max}$, if $\pi = \{J_{[1]}, J_{[2]}, \ldots, J_{[n]}\}$, the starting time of job $J_{[1]}$ is t_0; then the completion time of job J_j and the makespan are $C_j = t_0 \prod_{i=1}^{j}(1 + \alpha_{[i]})$ and $C_{\max}(\pi) = t_0 \prod_{j=1}^{n}(1 + \alpha_{[j]})$, respectively.*

Theorem 2.1.2. *The problem $1|r_j, p_j = \alpha_j t|L_{\max}$ is NP-hard even if there are only two distinct release dates.*

Proof. We use a reduction from the Subset-Product Problem, which is NP-hard (see [8]).

An instance I of the Subset-Product Problem is formulated as follows. Given positive integers $x_1, \ldots, x_m$ such that $\prod_{j=1}^{m} x_j = A^2$, does there exist a subset N_1 of set $N = \{1, \ldots, m\}$ such that $\prod_{j \in N_1} x_j = A$?

In the above instance, we can omit the element $j \in N$ with $x_j = 1$, because it will not affect the product of any subset. Therefore, without loss of generality, we can assume that $x_j \geq 2$ for every $j \in N$.

For any given instance I of the Subset-Product Problem, we construct a corresponding instance II of our problem as follows.

• There are in total $n = m + 1$ jobs.

• The deteriorating rates, release dates and due dates of jobs are defined by $r_j = 1, \alpha_j = x_j - 1$ and $d_j = A^3$ for $1 \le j \le m$, and $r_{m+1} = A, \alpha_{m+1} = A - 1$ and $d_{m+1} = A^2$.

• The threshold value is defined by $G = 0$.

It is clear that the reduction can be done in polynomial time, and it is easy to verify that $\alpha_j > 0$ for $j = 1, \ldots, m+1$.

Now we prove that instance I has a solution if and only if instance II has a schedule π with $L_{\max}(\pi) \le G$.

To prove the necessity, suppose that instance I has a solution $N_1 \subseteq N$ such that $\prod_{j \in N_1} x_j = A$.

For convenience, denote by J_{N_1} ($J_{N \setminus N_1}$) the set of jobs corresponding to the elements of set N_1 ($N \setminus N_1$). As a result, a schedule $\pi = \{J_{N_1}, J_{m+1}, J_{N \setminus N_1}\}$ satisfying the following conditions is obtained.

• Jobs in J_{N_1} are scheduled from time $t = 1$ and completed at time A since $r_j = 1$ for $1 \le j \le m$ and $\prod_{j \in N_1}(1 + \alpha_j) = \prod_{j \in N_1} x_j = A$. We denote the last job in J_{N_1} as J_k. Then, from Lemma 2.1.1, the completion time of job J_k is $C_k = 1 \cdot \prod_{j \in N_1}(1 + \alpha_j) = \prod_{j \in N_1} x_j = A$.

• Schedule job J_{m+1} from time $t = A$; this is feasible since $r_{m+1} = A$ and the completion time of the last job in J_{N_1} is $C_k = A$. Therefore, $C_{m+1} = A \cdot (1 + \alpha_{m+1}) = A^2$.

• Schedule jobs in $J_{N \setminus N_1}$ from time $t = C_{m+1} = A^2$ without idle time on the machine. We denote the last job in $J_{N \setminus N_1}$ as $J_{k'}$. Therefore, the completion time is $C_{k'} = C_{m+1} \cdot \prod_{j \in N \setminus N_1}(1 + \alpha_j) = A^2 \cdot \prod_{j \in N \setminus N_1} x_j = A^3$.

Note that $d_j = A^3$ $(1 \le j \le m)$ and $d_{m+1} = A^2$. Thus,

$$L_{\max} = \max_{1 \le j \le m+1}\{C_j - d_j\} = \max\{A^3 - A^3, A^2 - A^2\} = 0 = G.$$

Hence, instance II has a solution.

To prove the sufficiency, suppose that there is a schedule π satisfying $L_{\max}(\pi) \le G = 0$. We are ready to prove that instance I has a solution.

First, job J_{m+1} must start exactly at its release date $r_{m+1} = A$ and finish at A^2 in schedule π, since $L_{\max}(\pi) \le G = 0$. Second, there are no idle intervals from time $t = 1$ to time $t = A$ and from time $t = A^2$ to time $t = A^3$. Otherwise, there must exist at least one job J_j with $C_j > A^3 = d_j$ since $\prod_{j=1}^{m}(1 + \alpha_j) = \prod_{j=1}^{m} x_j = A^2$, a contradiction.

Thus, there exist some jobs to be scheduled in the interval $[1, A)$ without any idle time, i.e., there exists a set $N_1 \subseteq N$ such that the completion time of the last job in J_{N_1} is $1 \cdot \prod_{j \in N_1}(1 + \alpha_j) = \prod_{j \in N_1} x_j = A$. Thus, instance I has a solution. This completes the proof. □

2.2. *A 2-approximation algorithm of the case where all jobs have negative due dates*

In this subsection, we assume that all due dates are negative. If the maximum lateness $L_{\max} = \max_{j=1,\ldots,n}\{C_j - d_j\} \le 0$, the optimization problem is not particularly amenable to obtaining near-optimal solutions. If there were a ρ-approximation algorithm, then, for any input with optimal value 0, the algorithm must still find a schedule of objective function value at most $\rho \cdot 0 = 0$, and hence this would imply that P = NP. One easy workaround to this is to assume that all due dates are negative. Thus, the case of negative due dates is rational. We shall give a 2-approximation algorithm for this special case.

Assume $S \subseteq J$ and let $r(S) = \min_{j \in S}\{r_j\}, \alpha(S) = \prod_{j \in S}(1 + \alpha_j)$, and $d(S) = \max_{j \in S}\{d_j\}$. Let $L^*_{\max}$ denote the optimal value. At each moment that the machine is idle, start processing next an available job with the earliest due date. This is known as the EDD rule. Here, a job is available at time t if its release date is less than or equal to t.

Theorem 2.2.1. *The EDD rule is a 2-approximation algorithm for the problem $1|r_j, p_j = b_j t|L_{\max}$ with negative due dates.*

Proof. Let π be a schedule constructed by applying the EDD rule. Let job J_j be a job of maximum lateness schedule π, i.e., $L_{\max}(\pi) = C_j - d_j$.

We first provide a good lower bound on the optimal value for problem $1|r_j, p_j = b_j t|L_{\max}$ as follows.

Claim 2.2.2. *$L^*_{\max} \ge r(S)\alpha(S) - d(S)$ for each subset $S \subseteq J$.*

Proof of Claim 2.2.2. Consider any optimal schedule π^*,and view this simply as a schedule for jobs in the subset S. Let job J_i be the last job in S to be processed in π^*. We have that the completion time of J_i holds $C_i \ge r(S)\alpha(S)$ since none of the jobs in S can be processed before $r(S)$. Note that the due date of job J_i $d_i \le d(S)$. Thus, the lateness of job J_i in this schedule holds $L_i \ge r(S)\alpha(S) - d(S)$. Thus, $L^*_{\max} \ge r(S)\alpha(S) - d(S)$. □

Find the earliest point in time $t_0 \le C_j$ such that the machine was processing without any idle time for the entire period $[t_0, C_j)$. Let S be the set of jobs processed in the interval $[t_0, C_j)$. We know that, just prior to t_0, none of these jobs in S were available by our choice of time t_0. Thus, $r(S) = t_0$. Furthermore, we have that $\alpha(S) = \frac{C_j}{t_0} = \frac{C_j}{r(S)}$ since only jobs in S are processed throughout interval $[t_0, C_j)$. Thus, $C_j = r(S)\alpha(S)$. Since $d(S) < 0$, we can apply Claim 2.2.2 to get that

$$L^*_{\max} \ge r(S)\alpha(S) - d(S) \ge r(S)\alpha(S) = C_j. \tag{1}$$

On the other hand, by applying Claim 2.2.2 with $S = \{J_j\}$,

$$L^*_{\max} \geq r_j(1+\alpha_j) - d_j \geq -d_j. \quad (2)$$

Adding the above two inequalities (1) and (2), we have that

$$2L^*_{\max} \geq C_j - d_j = L_{\max}(\pi),$$

i.e.,

$$\frac{L_{\max}(\pi)}{L^*_{\max}} \leq 2.$$

This completes the proof. □

2.3. An optimal algorithm of one special case

The special case where all jobs have agreeable release dates, due dates and deteriorating rates is considered in this subsection. We assume that the jobs are indexed in such a way that $r_1 \leq r_2 \leq \cdots \leq r_n$, $d_1 \leq d_2 \leq \cdots \leq d_n$ and $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$. Note that the above order obeys the EDD rule.

Theorem 2.3.1. *The EDD rule provides an optimal schedule for problem $1|r_j, p_j = b_j t|L_{\max}$ with agreeable release dates, due dates and deteriorating rates.*

Proof. Let π be any optimal schedule for problem $1|r_j, p_j = \alpha_j t|L_{\max}$ with agreeable release dates, due dates and deteriorating rates. Suppose that there are two adjoint jobs J_j and J_{j+1} such that job J_{j+1} is processed before job J_j but $d_{j+1} \geq d_j$ in schedule π. We have that $r_{j+1} \geq r_j$ and $\alpha_{j+1} \geq \alpha_j$ since the release dates, due dates and deteriorating rates are assumed to be agreeable; this implies that there is no idle time between jobs J_j and J_{j+1}. For convenience, let t_0 be the starting time of job J_{j+1} in schedule π. Then, the lateness values of jobs J_{j+1} and J_j are $L_{j+1} = t_0(1+\alpha_{j+1}) - d_{j+1}$ and $L_j = t_0(1+\alpha_{j+1})(1+\alpha_j) - d_j$, respectively.

We get a new schedule π' by swapping jobs J_j and J_{j+1}; meanwhile, we have $L'_j = t'_0(1+\alpha_{j+1}) - d_j$ and $L'_{j+1} = t'_0(1+\alpha_{j+1})(1+\alpha_j) - d_{j+1}$, respectively, where t'_0 denotes the starting time of job J_j in schedule π'. It may be that $t'_0 \leq t_0$ since $r_{j+1} \geq r_j$. Note that $\alpha_{j+1} \geq \alpha_j$ and $d_{j+1} \geq d_j$. Therefore, we have $L'_j \leq L_j$, $L'_{j+1} \leq L_j$ and $L_{j+1} \leq L_j$, and the lateness of other jobs does not increase. From this, we have that the swap does not increase the value of the maximum lateness. A finite number of repetitions of this procedure yields an optimal schedule of the required form. This completes the proof. □

2.4. EDD rule in the general case

In this subsection, we show that the EDD rule is a bad algorithm for problem $1|r_j, p_j = \alpha_j t|L_{\max}$. Let $L^{EDD}_{\max}$ and $L^*_{\max}$ be the maximum lateness using the EDD rule and an optimal algorithm for problem $1|r_j, p_j = \alpha_j t|L_{\max}$, respectively.

Theorem 2.4.1. $\frac{L^{EDD}_{\max}}{L^*_{\max}}$ *can be arbitrarily large.*

Proof. Consider the problem with the following instance.
$(r_1, d_1, \alpha_1,) = (n^2+1, 1, n-1)$.
$(r_2, d_2, \alpha_2,) = (n, 2, n)$.

Using the EDD rule, we may have that job J_1 followed by job J_2 is processed from time $t = r_1 = n^2 + 1$ with maximum lateness $L^{EDD}_{\max} = \max\{L_1, L_2\} = \max\{n^3 + n - 1, n^4 + n^2 - 2\} = n^4 + n^2 - 2$. However, in the optimal solution, job J_2 should be processed from time $t = r_2 = n$ and job J_1 should be processed from time $t = n^2 + 1$; the optimal value is $L^*_{\max} = \max\{L_1, L_2\} = \max\{n^3 + n - 1, n^2 - 2\} = n^3 + n - 1$. It can be seen that $\frac{L^{EDD}_{\max}}{L^*_{\max}} = \frac{n^4+n^2-2}{n^3+n-1}$ can be arbitrarily large as n approaches infinity. This completes the proof. □

3. Problem $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$

In this section, we discuss unbounded parallel-batch scheduling.

3.1. NP-hardness proof

Theorem 3.1.1. *The problem $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$ is NP-hard.*

Proof. We use a reduction from the Subset-Product Problem. Let I' be an instance of the Subset-Product Problem. And we can assume that $x_j \geq 2$ for every $j \in N$. For the given instance I', we construct a corresponding instance II' of problem $1|p-batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$ as follows.

We define $T_h = 2A\prod_{i=h+1}^{m} T_i$ and $T = 2(A\prod_{h=1}^{m} T_h)^2$, and assume that the parameter $T_i > 2$ $(i = 1, \ldots, m)$. The number of jobs is $n = 3m + 1$ and there are four types (i), (ii), (iii), (iv) of job, as follows.

(i) Let J_{3m+1} be a job with $r_{3m+1} = A\prod_{h=1}^{m} T_h$, $\alpha_{3m+1} = 1$ and $d_{3m+1} = 2A\prod_{h=1}^{m} T_h$.

(ii) The parameters of job J_1 are $r_1 = 1$, $\alpha_1 = T_1 - 1$ and $d_1 = \frac{T}{T_1}$. For $2 \leq j \leq m$, $r_j = \prod_{h=1}^{j-1} T_h$, $\alpha_j = T_j - 1$ and $d_j = \frac{T}{\prod_{h=1}^{j} T_h}$.

(iii) The parameters of job J_{m+1} are $r_{m+1} = T_1$, $\alpha_{m+1} = T_m - 1$ and $d_{m+1} = T$. For $2 \leq j \leq m$, $r_{m+j} = \prod_{h=1}^{j} T_h$, $\alpha_{m+j} = T_j - 1$ and $d_{m+j} = \frac{T}{\prod_{h=1}^{j-1} T_h}$.

(iv) The parameters of job J_{2m+1} are $r_{2m+1} = 1$, $\alpha_{2m+1} = T_1 x_1 - 1$ and $d_{2m+1} = T$. For $2 \leq j \leq m$, $r_{2m+j} = \prod_{h=1}^{j-1} T_h$, $\alpha_{2m+j} = T_j x_j - 1$ and $d_{2m+j} = \frac{T}{\prod_{h=1}^{j-1} T_h}$.

The threshold value is defined by $G' = 0$.

It is clear that the reduction can be done in polynomial time, and it is easy to verify that the deteriorating rate $\alpha_j > 1$ for $j = 1, \ldots, 3m$ and $\alpha_{3m+1} = 1$.

We will prove that instance I' has a solution if and only if instance II' has a schedule π with $L_{\max}(\pi) \leq G'$.

To prove the necessity, suppose that the Subset-Product Problem has a solution. Without loss of generality, we assume that $N_1 = \{1, \ldots, k\}$; then $N \backslash N_1 = \{k+1, \ldots, m\}$ and $\prod_{j \in N_1} x_j = \prod_{j=1}^{k} x_j = A = \prod_{j=k+1}^{m} x_j$.

Now, we construct the following schedule π:

$$\begin{array}{llllll} J_1 & \cdots J_k & J_{k+1} \cdots J_m J_{3m+1} & J_{2m} \cdots J_{m+k+1} & J_{m+k} \cdots J_{m+1} \\ J_{2m+1} & \cdots J_{2m+k} & & J_{3m} \cdots J_{2m+k+1} & \end{array},$$

where the two jobs in the same column are processed as a batch.

Obviously, there are $2m + 1$ batches in schedule π.

From the construction of instance II', we have that schedule π is a feasible schedule with the following results.

$C(B_1) = C_1 = C_{2m+1} = T_1 x_1$. Therefore, $C_1 < \frac{T}{T_1} = d_1$ and $C_{2m+1} < T = d_{2m+1}$ since $T = 2(A\prod_{h=1}^{m} T_h)^2$. That is, $L_1 < 0$ and $L_{2m+1} < 0$. Similarly, we have $L_j < 0$ for $j = 1, \ldots, m, 2m+1, \ldots, 2m+k$. Meanwhile, we have $C(B_m) = A\prod_{h=1}^{m} T_h = r_{3m+1}$.

$$C(B_{m+1}) = C_{3m+1} = C(B_m)(1 + \alpha_{3m+1}) = 2A\prod_{h=1}^{m} T_h = d_{3m+1}.$$

Therefore, $L_{3m+1} = 0$. Similarly, we have $L_j < 0$ for $j = 2m, \ldots m+k+2, 3m, \ldots, 2m+k+2$, and $L_j = 0$ for $j = m+k+1, \ldots m+1, 2m+k+1$.

From the above discussion, we have that the lateness of all jobs is less than or equal to zero; that is, $L_{\max} = \max_{1 \leq j \leq 3m+1}\{L_j\} = 0 = G'$.

To prove the sufficiency, suppose that there is a schedule π that satisfies $L_{\max}(\pi) \leq G' = 0$. Then, the following conditions hold.

- Job J_{3m+1} must be scheduled only in the interval $[r_{3m+1}, d_{3m+1}]$ alone in schedule π. Otherwise, the schedule will not be feasible with $L_{3m+1} > 0$ since $\alpha_{3m+1} = 1$ and $\alpha_j > 1$ for $1 \leq j \leq 3m$. This is a contradiction.
- For $j = 1, \ldots m$, job J_j must be scheduled before job J_{3m+1}. Otherwise,

$$\begin{aligned} \frac{d_{3m+1}(1+\alpha_j)}{d_j} &= \frac{2AT_j\prod_{h=1}^{m} T_h}{\frac{T}{\prod_{h=1}^{m} T_h}} \\ &= \frac{2A \cdot 2AT_{j+1}\cdots T_m\left(\prod_{h=1}^{m} T_h\right)^2}{T} \\ &> \frac{4\left(A\prod_{h=1}^{m} T_h\right)^2}{T} = 2 > 1, \end{aligned}$$

a contradiction.

- For $j = 1, \ldots m$, job J_j must start before r_{j+1}. Otherwise,

$$C_j \geq r_{j+1}(1+\alpha_j) = T_j\prod_{h=1}^{j} T_h = 2A\prod_{h=1}^{m} T_h > r_{3m+1},$$

a contradiction.

• For $j = 1, \ldots m$, job J_{m+j} must be scheduled after job J_{3m+1}. Otherwise,

$$\begin{aligned} C_{m+j} \geq r_{m+j}(1+\alpha_{m+j}) &= T_j \prod_{h=1}^{j} T_h \\ &= 2A \prod_{h=1}^{m} T_h > r_{3m+1}, \end{aligned}$$

a contradiction.

• For $j = 1, \ldots m$, job J_{m+j} must be completed after d_{m+j+1}. Otherwise,

$$\begin{aligned} \frac{d_{m+j+1}}{1+\alpha_{m+j}} &= \frac{T}{T_j \prod_{h=1}^{j} T_h} = \frac{T}{2A \prod_{h=1}^{m} T_h} \\ &= A \prod_{h=1}^{m} T_h < d_{3m+1}, \end{aligned}$$

a contradiction.

• For each $j = 1, \ldots m$, if job J_{2m+j} is scheduled before job J_{3m+1}, then it must be in the same batch as job J_j since $r_j = r_{2m+j}$ and $r_j(1+\alpha_j) = T_j T_j x_j \prod_{h=1}^{j-1} T_h = 2Ax_j \prod_{h=1}^{m} T_h > r_{3m+1}$; if job J_{2m+j} is scheduled after job J_{3m+1}, then it must be in the same batch as job J_{m+j} since $d_{m+j} = d_{2m+j}$ and

$$\begin{aligned} \frac{d_{m+j}}{(1+\alpha_{m+j})(1+\alpha_{2m+j})} &= \frac{T}{T_j T_j x_j \prod_{h=1}^{j-1} T_h} \\ &= \frac{T}{2Ax_j \prod_{h=1}^{m} T_h} \\ &= \frac{A \prod_{h=1}^{m} T_h}{x_j} < d_{3m+1}. \end{aligned}$$

Denote by J_{N_1} the set of jobs corresponding to elements of set $N_1 \subseteq N$, and jobs in J_{N_1} are scheduled before job J_{3m+1}. We have

$$\prod_{j \in N_1} (1+\alpha_j) \prod_{j \in N_1} (1+\alpha_j) = \prod_{j=1}^{m} T_j \prod_{j \in N_1} x_j \leq r_{3m+1} = A \prod_{j=1}^{m} T_j;$$

then, $\prod_{j \in N_1} x_j \leq A$. On the other hand,

$$\begin{aligned} \prod_{j \in N \setminus N_1} (1+\alpha_{2m+j}) \prod_{j \in N \setminus N_1} (1+\alpha_{m+j}) &= \prod_{j=1}^{m} T_j \prod_{j \in N \setminus N_1} x_j \\ &\leq \frac{T}{d_{3m+1}} \\ &= \frac{2\left(A \prod_{j=1}^{m} T_j\right)^2}{2A \prod_{j=1}^{m} T_j} \\ &= A \prod_{j=1}^{m} T_j; \end{aligned}$$

then, $\prod_{j \in N \setminus N_1} x_j \leq A$.

Note that $\prod_{j\in N_1} x_j \prod_{j\in N\setminus N_1} x_j = A^2$. Thus, $\prod_{j\in N_1} x_j = A$, i.e., instance I' of the Subset-Product Problem has a solution. This completes the proof. □

3.2. Agreeable due dates and release dates

In this subsection, the due dates and release dates of jobs are assumed to be agreeable. Thus, the jobs can be re-indexed such that $r_1 \le r_2 \le \cdots \le r_n$ and $d_1 \le d_2 \le \cdots \le d_n$.

Theorem 3.2.1. *For problem $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$ with agreeable due dates and release dates, there exists an optimal batch sequence $BS = \{B_1, B_2, \ldots, B_k\}$ such that, for every two batches B_{k_1} and B_{k_2} with $k_1 < k_2$, $\max\{r_i : J_i \in B_{k_1}\} < \min\{r_j : J_j \in B_{k_2}\}$.*

Proof. Let $\pi = \{BS, ST\}$ be an optimal schedule for problem $1|B, r_j, p_j = \alpha_j t|L_{\max}$ with agreeable due dates and release dates, where $BS = \{B_1, B_2, \ldots, B_k\}$ and $ST = \{S(B_1), S(B_2), \ldots, S(B_k)\}$. Suppose that there are two batches B_{k_1} and B_{k_2} with $k_1 < k_2$ such that $\max\{r_i : J_i \in B_{k_1}\} \ge \min\{r_j : J_j \in B_{k_2}\} = r_{j0}$. Then, there exists job $J_i \in B_{k_1}$ such that $r_i \ge r_{j0}$, and then $d_i \ge d_{j_0}$ since the case has agreeable due dates and release dates. Let $\alpha(B_{k_1})$ and $\alpha(B_{k_2})$ denote the deteriorating rates of batches B_{k_1} and B_{k_2}, respectively.
Case 1. $\alpha(B_{k_1}) \ge \alpha(B_{k_2})$.

We get a new schedule π' by moving job J_{j_0} from batch B_{k_2} to batch B_{k_1}. Note that the deteriorating rate of batch B_{k_1} does not change and the deteriorating rate of batch B_{k_2} does not increase. Therefore, we have that $L'_{j_0} = S(B_{k_1})(1+b(B_{k_1})) - d_{j_0} \le S(B_{k_2})(1+\alpha(B_{k_2})) - d_{j_0} = L_{j_0}$ and the lateness of other jobs does not increase.
Case 2. $\alpha(B_{k_1}) < \alpha(B_{k_2})$.

We get a new schedule π' by moving job J_i from batch B_{k_1} to batch B_{k_2}. Note that the deteriorating rate of batch B_{k_2} does not change and the deteriorating rate of batch B_{k_1} does not increase. Let $S'(B_{k_2})$ be the starting time of batch B_{k_2} in schedule π'; then $S'(B_{k_2}) \le S(B_{k_2})$. Therefore, we have that $L_i = S(B_{k_1})(1+\alpha(B_{k_1})) - d_i < S(B_{k_2})(1+\alpha(B_{k_2})) - d_j = L_{j_0}$ since $d_i \ge d_{j_0}$, and $L'_i = S'(B_{k_2})(1+\alpha(B_{k_2})) - d_i \le L'_{j_0} = S'(B_{k_2})(1+\alpha(B_{k_2})) - d_{j_0} \le L_{j_0}$. The lateness of other jobs does not increase. Clearly, schedule π' is still an optimal batch sequence. Continuing this procedure, we eventually obtain an optimal batch sequence with the required form. This completes the proof. □

4. Conclusion

In this paper, we consider the scheduling of simple deteriorating jobs with release dates on a single machine to minimize the maximum lateness. We prove that the problem $1|r_j, p_j = \alpha_j t|L_{\max}$ is NP-hard, give a 2-approximation algorithm for the case where all jobs have negative due dates, and discuss the EDD rule. And we prove that the batch-scheduling problem $1|p - batch, r_j, p_j = \alpha_j t, b = \infty|L_{\max}$ is NP-hard, and present one property of the optimal schedule for one special case.

For future research, it is worth considering other objectives. Furthermore, one may continue our research with the same objective, but focus on the polynomial time approximation scheme (PTAS) or fully polynomial time approximation scheme (FPTAS) for our two NP-hard problems.

Acknowledgments

We thank two anonymous reviewers for their helpful and detailed comments on an earlier version of our paper. This work was supported by the National Natural Science Foundation of China (11071142, 11201259), the Natural Science Foundation of Shandong Province (ZR2011AL017, ZR2010AM034).

References

[1] P. Brucker, A. Gladky, H. Hoogeveen, M.Y. Kovalyov, C.N. Potts, T. Tautenhahn, S.L. van de Velde, Scheduling a batching, Journal of Scheduling 1 (1998) 31–54.
[2] A. Bachman, A. Janiak, Scheduling Jobs with Special Type of Start Time Dependent Processing Time. Report no. 34/97, Institute of Engineering Cybernetics, Wroclaw University of Technology, Wroclaw, 1997.
[3] A. Bachman, A. Janiak, Minimizing maximum lateness under linear deterioration, European Journal of Operational Research 126 (2000) 557–566.
[4] Z.L. Chen, Parallel machine scheduling with time-dependent processing times, Discrete Applied Mathematics 70 (1996) 81–93.
[5] T.C.E. Cheng, Q. Ding, Single machine scheduling with deadlines and increasing rates of processing times, Acta Informatica 36 (9–10) (2000) 673–692.
[6] T.C.E. Cheng, Q. Ding, Single machine scheduling with step-dependent processing times, European Journal of Operational Research 134 (2001) 623–630.
[7] T.C.E. Cheng, Z.H. Liu, W.C. Yu, Scheduling jobs with release dates and deadline on a batch processing machine, IIE Transactions 33 (2001) 685–690.
[8] M.R. Garey, D.S. Johnsonm, Computers and intractability: a guide to the theory of NP-Completeness, Freeman, New York, 1979.
[9] R.L. Graham, Lawle, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann Disc Math 5 (1979) 287–326.
[10] Y.S. Hsu, B.M.T. Lin, Minimization of maximum lateness under linear deterioration, Omeg 31 (6) (2003) 459–469.
[11] M. Ji, T.C.E. Cheng, Parallel-machine scheduling with simple linear deterioration to minimize total completion time, European Journal of Operational Research 188 (2008) 342–347.
[12] M. Ji, T.C.E. Cheng, Parallel machine scheduling of single linear deteriorating jobs, Theoretical Computer Science 410 (2009) 3761–3768.

[13] A.S. Kunnathur, S.K. Gupta, Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problems, European Journal of Operational Research 47 (1990) 56–64.
[14] M.Y. Kovalyov, W. Kubiak, A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs, Journal of Heuristics 3 (4) (1998) 287–297.
[15] C.Y. Lee, R. Uzsoy, L.A. Martin-Vega, Efficient algorithms for scheduling semiconductor burn-in operations, Operations Research 40 (1992) 764–775.
[16] Z.H. Liu, W.C. Yu, Scheduling one batch processor subject to job release dates, Discrete Applied Mathematics 70 (2000) 81–93.
[17] L.F. Lu, T.C.E. Cheng, J.J. Yuan, L.Q. Zhang, Bounded single-machine parallel-batch scheduling with release dates and rejection, Computer and Operations Research 36 (2009) 2748–2751.
[18] S.S. Li, C.T. Ng, T.C.E. Cheng, J.J. Yuan, Parallel-batch scheduling of deteriorating jobs with release date to minimize the makespan, European Journal of Operational Research 210 (2011) 482–488.
[19] G. Mosheiov, Scheduling jobs under simple linear deterioration, Computers and Operations Research 21 (1994) 653–659.
[20] G. Mosheiov, Multi-machine scheduling with linear deterioration, INFOR 36 (1998) 205–214.
[21] C.X. Miao, Z.Y. Zhang, Z.G. Cao, Parallel-batch scheduling of deteriorating jobs with release date to minimize the makespan, Information Processing Letters 111 (2011) 798–803.
[22] C.N. Potts, M.Y. Kovalyov, Scheduling with batching: a review, European Journal of Operational Research 120 (2000) 228–249.
[23] X.L. Qi, S.G. Zhou, J.J. Yuan, Single machine parallel-batch scheduling with deteriorating jobs, Theoretical Computer Science 410 (2009) 830–836.
[24] P.S. Sundararaghavan, A.S. Kunnathur, Single machine scheduling with start time-dependent processing times: some solvable cases, European Journal of Operational Research 78 (1994) 393–403.
[25] Webster, K.R. Baker, Scheduling groups of jobs on a single machine, Operations Research 43 (1995) 692–7039.
[26] Y.Z. Zhang, Z.G. Cao, Parallel batch scheduling: a survey, Advances of mathematics 37 (2008) 392–408.