Research Article

Parallel Batch Scheduling of Deteriorating Jobs with Release Dates and Rejection

Juan Zou^{1,2} and Cuixia Miao¹

¹ School of Mathematical Sciences, Qufu Normal University, Qufu, Shandong 273165, China

² School of Management Sciences, Qufu Normal University, Rizhao, Shandong 276826, China

Correspondence should be addressed to Juan Zou; zoujuanjn@163.com

Received 12 March 2014; Accepted 1 July 2014; Published 21 July 2014

Academic Editor: Bahman Naderi

Copyright © 2014 J. Zou and C. Miao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider the unbounded parallel batch scheduling with deterioration, release dates, and rejection. Each job is either accepted and processed on a single batching machine, or rejected by paying penalties. The processing time of a job is a simple linear increasing function of its starting time. The objective is to minimize the sum of the makespan of the accepted jobs and the total penalty of the rejected jobs. First, we show that the problem is NP-hard in the ordinary sense. Then, we present two pseudopolynomial time algorithms and a fully polynomial-time approximation scheme to solve this problem. Furthermore, we provide an optimal $O(n \log n)$ time algorithm for the case where jobs have identical release dates.

1. Introduction

The parallel batch scheduling problem has been extensively studied in the literature over the last decade. The fundamental model of the parallel batch scheduling problem was first introduced by Lee et al. [1] with the restriction that the number of jobs in each batch is bounded by a number b . This bounded model is motivated by the burn-in operations in semiconductor manufacturing. Brucker et al. [2] provided an extensive discussion of the unbounded version of the parallel batch scheduling problem. This unbounded model can be applied, for example, to situations where the batch contents are heated in a sufficiently large kiln. Cheng et al. [3], Deng et al. [4, 5], and Liu et al. [6, 7] presented new complexity results and approximation algorithms for the parallel batch scheduling problem subject to release dates. In addition, more recent developments of this topic can be found in [8].

In classical scheduling problems, it is assumed that the processing times of jobs are fixed parameters. In practice, however, we often encounter situations in which processing time increases over time. Examples can be found in financial management, steel production, firefighting, and so forth, where any delay in processing a job may increase its completion time. Such problems are generally known as

scheduling with deterioration effects. Browne and Yechiali [9] first considered scheduling problem with linear deterioration. They provided the optimal solution for the single machine when the objective is to minimize makespan. In addition, they solved a special case when the objective function is to minimize the total weighted completion time. Mosheiov [10] considered single machine scheduling problems with simple linear deteriorating jobs. He showed that most commonly scheduling objectives, such as the makespan, the total flow time, the sum of weighted completion times, the maximum lateness, the total lateness, the maximum tardiness, and the number of tardy jobs, remain solvable in polynomial time. Alidaee and Womer [11] presented an extensive survey of different deteriorating models and scheduling problems. Cheng et al. [12] provided an updated survey of scheduling problems with time-dependent processing times. Li et al. [13] considered the problem of scheduling of deteriorating jobs with release dates on a single batching machine. Miao et al. [14] studied the scheduling problems with deteriorating jobs and release dates on a single (batching) machine to minimize the maximum lateness. Zou et al. [15] considered several uniform parallel machine scheduling problems with linear deteriorating jobs.

On the other hand, it is always assumed in traditional scheduling research that all the jobs have to be processed on some machines. In real applications, however, things may be more complicated. For example, due to limited resources, the scheduler may reject some jobs and thereby incur job-dependent penalties for each rejected job. Bartal et al. [16] first introduced the notion of rejection. They considered both the offline and online versions of scheduling problem and studied the multiprocessor scheduling problem to minimize the sum of the makespan of the accepted jobs and the total rejection penalty of the rejected jobs. Subsequently, the machine scheduling problem with rejection has received increasing research attention. Hoogeveen et al. [17] considered the offline multiprocessor scheduling problem with rejection where preemption is allowed. They provided a pseudopolynomial time algorithm for each problem. Lu et al. [18] considered the unbounded parallel batch machine scheduling problem with release dates and rejection. Engels et al. [19] studied the single machine scheduling with rejection to minimize the sum of the weighted completion times of the scheduled jobs and the total penalty of the rejected jobs. Shabtay et al. [20] recently presented an updated survey on offline scheduling with rejection. In addition, there has been some work on scheduling deteriorating jobs with rejection. For example, Cheng and Sun [21] considered the single machine scheduling problems with linear deteriorating jobs and rejection. Li and Yuan [22] studied several parallel machine scheduling problems in which the processing time of a job is a linear increasing function of its starting time and jobs can be rejected by paying penalties. To the best of our knowledge, no work has been done on the problem of scheduling deteriorating jobs with rejection on a single batching machine. Literature [20] on scheduling with rejection focuses on the following four models:

- P1: to minimize the total integrated cost $F_1 + F_2$;
- P2: to minimize F_1 subject to $F_2 \leq \bar{E}$;
- P3: to minimize F_2 subject to $F_1 \leq K$;
- P4: to identify a Pareto-optimal solution for each Pareto-optimal point,

where F_1 is the original scheduling objective function and F_2 is total rejection cost. We discuss the first model with deterioration and release dates.

In this paper, we study parallel batch scheduling of deteriorating jobs with release dates and rejection. The paper is organized as follows. In Section 2, we formulate the problem under consideration and introduce some notations. In Section 3, we first show that the problem is binary NP-hard and provide two pseudopolynomial time algorithms. We also propose a fully polynomial-time approximation scheme for the problem and discuss one special case. In Section 4, we conclude the paper and suggest some topics for future research.

2. Problem Description and Notation

The problem considered in this paper can be formally described as follows. There are an independent job set

$\mathcal{J} = \{J_1, \dots, J_n\}$ and a single batching machine. The machine can process up to b job simultaneously as a batch, where $b \geq n$. The jobs processed in the same batch have the same starting time and completion time. The processing time of a batch is equal to the largest processing time of the jobs in the batch. Each job J_j ($j = 1, \dots, n$) has a release date r_j , a deteriorating rate $b_j > 0$, and a rejection penalty e_j . Each job J_j ($j = 1, \dots, n$) is either rejected with a rejection penalty e_j having to be paid or accepted and processed on the parallel batch machine. Denote by R and $\bar{R} = \mathcal{J} \setminus R$ the set of rejected jobs and the set of accepted jobs, respectively. The actual processing time of job J_j is given by $p_j = b_j t$, where $t \geq r_j$ is the starting time of J_j . We assume $\min\{r_j \mid j = 1, \dots, n\} \geq t_0 > 0$. For a given batch B , we denote its deteriorating rate and release date by $b(B)$ and $r(B)$, respectively. Then, $b(B) = \max\{b_j : J_j \in B\}$ and $r(B) = \max\{r_j : J_j \in B\}$. Denote by C_j the completion time of the accepted job J_j . We study the problem of minimizing the sum of the makespan of the accepted jobs (i.e., $C_{\max}(\bar{R}) = \max\{C_j : J_j \in \bar{R}\}$) and the total penalty of the rejected jobs. Using the three-field notation of Graham et al. [23], we denote the problem by $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$.

3. Main Results

3.1. NP-Hardness Proof. We first give a lemma, which will be used in what follows.

Lemma 1 (see [10]). *For the problem $1|r_j = t_0, p_j = b_j t|C_{\max}$, the maximum completion time of the jobs is equal to $C_{\max} = t_0 \prod_{J_j \in \mathcal{J}} (1 + b_j)$ and it does not depend on the processing order of the jobs.*

Theorem 2. *The problem $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$ is binary NP-hard.*

Proof. The decision version of the scheduling problem is clearly in NP. We use a reduction from the Equal Products Problem, which is NP-hard (see [24]).

An instance I of the Equal Products Problem is formulated as follows. Given a set $Q = \{1, 2, \dots, m\}$ and positive integers a_i for each $i \in Q$ such that $\prod_{i \in Q} a_i = A^2$ for a certain positive integer A , does there exist a subset $S \subset Q$ such that $\prod_{i \in S} a_i = A$?

For any given instance I of the Equal Products Problem, we construct a corresponding instance II of our problem as follows.

- (i) There are $n = 2m + 1$ jobs.
- (ii) For $i = 1$, there exist job J_1 with $(r_1, b_1, e_1) = (1, A^{2m} - 1, 2A^{m^2+m+1} + A)$ and job J_2 with $(r_2, b_2, e_2) = (1, A^{2m} a_1 - 1, \ln a_1)$.
- (iii) For each $2 \leq i \leq m$, there exist job J_{2i-1} with $(r_{2i-1}, b_{2i-1}, e_{2i-1}) = (A^{\sum_{k=0}^{i-2} (2m-2k)}, A^{2m-2(i-1)} - 1, 2A^{m^2+m+1} + A)$ and job J_{2i} with $(r_{2i}, b_{2i}, e_{2i}) = (A^{\sum_{k=0}^{i-2} (2m-2k)}, A^{2m-2(i-1)} a_i - 1, \ln a_i)$.

(iv) For $i = 2m + 1$, there exists only a job J_{2m+1} with $(r_{2m+1}, b_{2m+1}, e_{2m+1}) = (A^{m^2+m+1}, 1, 2A^{m^2+m+1} + A)$.

(v) The threshold value is given by $Y = 2A^{m^2+m+1} + \ln A$.

The decision asks whether there is a schedule such that $C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j \leq Y$. $\square$

It is clear that the reduction can be done in polynomial time, and it is easy to verify that $b_i > 0$ for $i = 1, \dots, m+1$. Assume first that the Equal Products Problem has a solution S such that $\prod_{i \in S} a_i = A$. Then, we construct a schedule in the following way. If $i \in S$, we assign the jobs J_{2i-1} and J_{2i} as a batch B_i . If $i \notin S$, we assign the job J_{2i-1} as a batch B_i . We assign the job J_{2m+1} as the last batch B_{m+1} . The jobs are processed on the batch machine in the order of $B_1, \dots, B_{m+1}$. All other jobs are rejected. It is easy to verify that

$$\begin{aligned} C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j &= \max \left\{ A^{m^2+m} \prod_{i \in S} a_i, r_{m+1} \right\} (1 + b_{m+1}) + \sum_{i \notin S} \ln a_i \\ &= 2A^{m^2+m+1} + \ln A = Y. \end{aligned} \quad (1)$$

Conversely, suppose that there is a schedule π with $C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j \leq Y$. We need to show that there is a subset S for the Equal Products Problem with $\prod_{i \in S} a_i = A$. We have the following claims.

Claim 1. All jobs J_{2i-1} for each $1 \leq i \leq m+1$ are accepted and $C_{\max}(\bar{R}) \geq 2A^{m^2+m+1}$.

Proof. If there exists some job J_{2i-1} ($1 \leq i \leq m+1$) being rejected, then we have $C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j \geq e_{2i-1} = 2A^{m^2+m+1} + A > Y$. It is a contradiction. Since job J_{2m+1} is accepted, we have $C_{\max}(\bar{R}) \geq r_{2m+1}(1 + b_{2m+1}) \geq 2A^{m^2+m+1}$.

From Claim 1, we can obtain that only the jobs $\{J_{2i} : 1 \leq i \leq m\}$ can be rejected. $\square$

Claim 2. Only job J_{2m+1} is processed in the last batch.

Proof. If there exists a job J_i ($1 \leq i \leq 2m$) with J_{2m+1} processed in the same batch B , then we have

$$\begin{aligned} C_{\max} &= r_{2m+1}(1 + b(B)) \\ &= r_{2m+1}(1 + b_i) \geq A^{m^2+m+1}(1 + A^2 - 1) \\ &= A^{m^2+m+3} > Y. \end{aligned} \quad (2)$$

Then, we denote by B_{m+1} the last batch containing only job J_{2m+1} . $\square$

Claim 3. Each batch contains either a single job J_{2i-1} or two jobs $\{J_{2i-1}, J_{2i}\}$ for $1 \leq i \leq m$.

Proof. For job J_{2m-1} , if there exists a job J_i ($1 \leq i \leq 2m-2$) with J_{2m-1} which are processed in the same batch B , then we have $C_{\max} = r_{2m-1}(1 + b(B)) = r_{2m-1}(1 + b_i) \geq A^{m^2+m-2}(1 + A^4 - 1) = A^{m^2+m+1} \cdot A > Y$.

Thus, we denote by B_m the new batch containing job J_{2m-1} .

For job J_{2m-3} , if there exists a job J_i ($1 \leq i \leq 2m-4$) with J_{2m-3} which are processed in the same batch B_{m-1} , then we have the completion time of the batch as

$$\begin{aligned} C(B_{m-1}) &\geq r_{2m-3}(1 + b(B_{m-1})) \\ &= r_{2m-3}(1 + b_i) \geq A^{m^2+m-6}(1 + A^6 - 1) \\ &= A^{m^2+m}. \end{aligned} \quad (3)$$

The completion time of batch B_m is

$$\begin{aligned} C(B_m) &= \max\{C(B_{m-1}), r_{2m-1}\}(1 + b(B_m)) \\ &\geq A^{m^2+m+2} > Y. \end{aligned} \quad (4)$$

Therefore, batch B_{m-1} can only contain either job J_{2m-3} or jobs $\{J_{2m-3}, J_{2m-4}\}$. Repeat this deduction, we have the correctness of Claim 3. Let B_i be the batch containing job J_{2i-1} for $1 \leq i \leq m$.

By Claims 2 and 3, we have batch $B_i = \{J_{2i-1}, J_{2i}\}$ if J_{2i} is accepted and batch $B_i = \{J_{2i-1}\}$ if J_{2i} is rejected. Let $S = \{i : J_{2i} \text{ is rejected for } 1 \leq i \leq m\}$. Next, we are ready to show that S is a solution of the Equal Products Problem.

Since $C_{\max}(\bar{R}) \geq 2A^{m^2+m+1}$ and $C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j \leq Y = 2A^{m^2+m+1} + \ln A$, we have $\sum_{j \in S} \ln a_j = \sum_{J_j \in R} e_{2j} \leq \ln A$. If $\prod_{j \in S} a_j = X < A$, we have

$$\begin{aligned} C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j &= \max\{r(B_{m+1}), C(B_m)\}(1 + b_{m+1}) + \sum_{j \in S} \ln a_j \\ &= 2 \max \left\{ A^{m^2+m+1}, A^{m^2+m} \cdot \prod_{i \in \bar{S}} a_i \right\} \\ &\quad + \ln X > 2A^{m^2+m+1} \cdot \frac{A}{X} + \ln X > Y. \end{aligned} \quad (5)$$

It is a contradiction. Hence, we have $\prod_{j \in S} a_j = A$ and S is a solution of the Equal Products Problem. This completes the proof. $\square$

3.2. Dynamic Programming Algorithms

Lemma 3. For problem 1|p-batch, $r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$, there exists an optimal schedule such that the accepted jobs are processed in a nonincreasing order of deteriorating rates.

Proof. Otherwise, there exist two accepted jobs J_i and J_j such that J_j is processed later than J_i , but $b_i < b_j$. Since $b \geq n$, there is no limit on the number of jobs in a batch and we can put job J_i into the batch containing job J_j . This does not delay the completion time of any batch. A finite number of repetitions of this procedure yields an optimal schedule of the required form. $\square$

Based on Lemma 3, we only consider the schedules in which the accepted jobs are processed in a nonincreasing order of deterioration rates. Assume that jobs are indexed such that $b_1 \geq \dots \geq b_n$.

Let $F(j, k, E)$ denote the optimal objective function value of the following scheduling problem:

- (i) the jobs in consideration are $J_1, \dots, J_j$;
- (ii) J_k is the job with the minimum index in the last batch;
- (iii) the total penalty of rejected jobs is E .

In any such schedule, there are two possible cases: either J_j is rejected or J_j is accepted and processed on the batching machine. We distinguish two cases in the following discussion.

Case 1. Job J_j is rejected. Since J_j is rejected, in the corresponding optimal schedule for $J_1, \dots, J_{j-1}$, J_k is still the job with the minimum index in the last batch and the total rejection penalty among $J_1, \dots, J_{j-1}$ is $E - e_j$. Therefore, we have $F(j, k, E) = F(j-1, k, E - e_j) + e_j$.

Case 2. Job J_j is accepted. In this case, we consider this problem in the following two cases.

Case 2.1. If $k = j$, then job J_j starts a new batch. The starting time of job J_j is $\max\{F(j-1, k', E) - E, r_j\}$. Therefore, we have $F(j, k, E) = \min_{0 \leq k' \leq j-1} \{\max\{F(j-1, k', E) - E, r_j\}(1 + b_j) + E\}$.

Case 2.2. If $k < j$, then job J_j can be assigned to the last batch containing job J_k . Therefore, $F(j, k, E) = \max\{F(j-1, k, E) - E, r_j\}(1 + b_k) + E$.

Combining the above two cases, we design the following dynamic programming algorithm.

Algorithm DP1. Consider the following steps.

Step 1 (initialization). Consider $F(1, 1, 0) = r_1(1 + b_1)$ and $F(1, k, E) = \infty$ for $(k, E) \neq (1, 0)$.

Consider $F(1, 0, e_1) = t_0 + e_1$ and $F(1, k, E) = \infty$ for $(k, E) \neq (0, e_1)$.

Step 2 (recursion). If $k = j$,

$$F(j, k, E) = \min \left\{ \min_{0 \leq k' \leq j-1} \left\{ \max \left\{ F(j-1, k', E) - E, r_j \right\} \times (1 + b_j) + E \right\}, F(j-1, k, E - e_j) + e_j \right\}. \quad (6)$$

If $k < j$,

$$F(j, k, E) = \min \left\{ \max \left\{ \frac{F(j-1, k, E) - E}{1 + b_k}, r_j \right\} (1 + b_k) + E, F(j-1, k, E - e_j) + e_j \right\}. \quad (7)$$

Step 3 (optimal solution). The optimal value is given by $\min\{F(n, k, E) : 0 \leq k \leq n, 0 \leq E \leq \sum_{j=1}^n e_j\}$.

Theorem 4. Algorithm DP1 solves the problem 1|p-batch, $r_j, p_j = b_j t, rej, b \geq n | C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$ in $O(n^2 \sum_{j=1}^n e_j)$ time.

Proof. The correctness of Algorithm DP1 is guaranteed by the above discussion. The recursive function has at most $O(n^2 \sum_{j=1}^n e_j)$ states; each iteration takes a constant time to execute. Hence, the running time of Algorithm DP1 is $O(n^2 \sum_{j=1}^n e_j)$. $\square$

Next, we give another dynamic programming Algorithm DP2. Let $f(j, k, E)$ denote the minimum value of the maximum completion time satisfying the following conditions.

- (i) The jobs in consideration are $J_1, \dots, J_j$.
- (ii) J_k is the job with the minimum index in the last batch.
- (iii) The total penalty of rejected jobs is E .

In any such schedule, there are two possible cases: either J_j is rejected or J_j is accepted and processed on the batching machine.

Case 1. If job J_j is rejected, we have $f(j, k, E) = f(j-1, k, E - e_j)$.

Case 2. Job J_j is accepted.

Case 2.1. If $k = j$, then job J_j starts a new batch; the starting time of job J_j is $\max\{f(j-1, k', E), r_j\}$. Therefore, we have $f(j, k, E) = \min_{0 \leq k' \leq j-1} \{\max\{f(j-1, k', E), r_j\}(1 + b_j)\}$.

Case 2.2. If $k < j$, then job J_j can be assigned to the last batch containing job J_k . Therefore, $f(j, k, E) = \max\{f(j-1, k, E)/(1 + b_k), r_j\}(1 + b_k)$.

Combining the above two cases, we design the dynamic programming algorithm as follows.

Algorithm DP2. Consider the following steps.

Step 1 (initialization). Consider $f(1, 1, 0) = r_1(1 + b_1)$ and $f(1, k, E) = \infty$ for any $(k, E) \neq (1, 0)$.

Consider $f(1, 0, e_1) = t_0$ and $f(1, k, E) = \infty$ for any $(k, E) \neq (0, e_1)$.

Step 2 (recursion). If $k = j$,

$$f(j, k, E) = \min \left\{ \min_{0 \leq k' \leq j-1} \left\{ \max \{f(j-1, k', E), r_j\} (1 + b_j) \right\}, f(j-1, k, E - e_j) \right\}. \quad (8)$$

If $k < j$,

$$f(j, k, E) = \min \left\{ \max \left\{ \frac{f(j-1, k, E)}{1 + b_k}, r_j \right\} (1 + b_k), f(j-1, k, E - e_j) \right\}. \quad (9)$$

Step 3 (optimal solution). The optimal value is determined by $\min\{f(n, k, E) + E : 0 \leq k \leq n, 0 \leq E \leq \sum_{j=1}^n e_j\}$.

Theorem 5. Algorithm DP2 solves the problem $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{j \in R} e_j$ in $O(n^2 \sum_{j=1}^n e_j)$ time.

3.3. An FPTAS. Algorithm $\mathcal{A}$ is a ρ -approximation ($\rho \geq 1$) algorithm for a minimization problem if it produces a solution that is at most ρ times than the optimal one (ρ is also referred to as the worst-case ratio). A family of algorithms $\{\mathcal{A}_\epsilon : \epsilon > 0\}$ is called a fully polynomial-time approximation scheme (FPTAS), if, for each $\epsilon > 0$, the algorithm $\mathcal{A}_\epsilon$ is a $(1 + \epsilon)$ -approximation algorithm running in polynomial time in the input size and $1/\epsilon$. In the sequel, we assume $0 < \epsilon \leq 1$.

For any $\epsilon' > 0$ and $x \geq 1$, if $(1 + \epsilon')^{k-1} < x < (1 + \epsilon')^k$, then we define $\lceil x \rceil_{\epsilon'} = (1 + \epsilon')^k$, $\lfloor x \rfloor_{\epsilon'} = (1 + \epsilon')^{k-1}$. If x is an exact power of $(1 + \epsilon')$, then $\lceil x \rceil_{\epsilon'} = \lfloor x \rfloor_{\epsilon'} = x$. Note that $\lceil x \rceil_{\epsilon'} \leq (1 + \epsilon')x$ for any $x \geq 1$.

For any $\epsilon > 0$, let $\epsilon' = \epsilon/2n$ and let $R = \{J_{i_1}, J_{i_2}, \dots, J_{i_m}\}$ be the set of the rejected jobs, where $i_1 < i_2 < \dots < i_m$. The inflated rejection penalty of the set R with respect to any $\epsilon' > 0$ is defined as

$$\phi_{\epsilon'}(R) = \begin{cases} \lceil e_{i_m} + \phi_{\epsilon'}(R - J_{i_m}) \rceil_{\epsilon'}, & \text{if } m \geq 1, \\ 0, & \text{if } R = \emptyset. \end{cases} \quad (10)$$

Lemma 6 (see [25]). For any set R of jobs and any $\epsilon' > 0$, $\phi_{\epsilon'}(R) \leq (1 + \epsilon')^{|R|} \sum_{j \in R} e_j$.

Lemma 7 (see [19]). For any $0 < x \leq 1$ and for any integer $n \geq 1$, $(1 + (x/n))^n \leq 1 + 2x$ holds.

Based on Lemmas 6 and 7, we have the following theorem.

Theorem 8. For any $0 < \epsilon \leq 2$, the optimal objective function value for $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \phi_{\epsilon'}(R)$

is at most $(1 + \epsilon)$ times the optimal objective function value for $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{j \in R} e_j$.

Assume that the jobs have been indexed such that $b_1 \geq \dots \geq b_n$ and define

$$\tau_l = \begin{cases} (1 + \epsilon')^l, & \text{if } l \neq -1, \\ 0, & \text{if } l = -1. \end{cases} \quad (11)$$

Let $f(j, k, l)$ denote the minimum value of the maximum completion time satisfying the following conditions.

- (i) The jobs in consideration are $J_1, \dots, J_j$.
- (ii) In the last batch, J_k is the job with the minimum index.
- (iii) The inflated rejection penalty of the rejected jobs is τ_l .

We design a dynamic programming algorithm as follows.

Algorithm $\mathcal{A}_\epsilon$. Consider the following steps.

Step 1 (initialization). Consider

$$f(1, k, l) = \begin{cases} t_0, & \text{if } \tau_1 = \lceil e_1 \rceil_{\epsilon'}, \\ r_1 (1 + b_1), & \text{if } l = -1, \\ +\infty, & \text{otherwise.} \end{cases} \quad (12)$$

Step 2 (recursion). If $k = j$, then

$$f(j, k, l) = \min \left\{ \min_{0 \leq k' \leq j-1} \left\{ \max \{f(j-1, k', l), r_j\} (1 + b_j) \right\}, \min_{-1 \leq l' \leq l} f(j-1, k, l') \right\}. \quad (13)$$

If $k < j$, then

$$f(j, k, l) = \min \left\{ \max \left\{ \frac{f(j-1, k, l)}{1 + b_k}, r_j \right\} (1 + b_k), \min_{-1 \leq l' \leq l} f(j-1, k, l') \right\}, \quad (14)$$

where $\tilde{l}$ is given by $(1 + \epsilon')^{\tilde{l}} = \lfloor (1 + \epsilon')^l - e_j \rfloor_{\epsilon'}$.

Step 3 (output). The optimal value is determined by $\min\{f(n, k, l) + \tau_l : 0 \leq k \leq n, -1 \leq l \leq L\}$, where L is the smallest integer such that $(1 + \epsilon')^L \geq (1 + \epsilon')^n \sum_{j=1}^n e_j$.

Theorem 9. There exists an FPTAS for problem $1|p\text{-batch}, r_j, p_j = b_j t, rej, b \geq n|C_{\max}(\bar{R}) + \sum_{j \in R} e_j$ which runs in $O((n^4/\epsilon^2) \log^2 \sum_{j=1}^n e_j)$ time.

Proof. The correctness of Algorithm $\mathcal{A}_\epsilon$ is guaranteed by the above discussion. We need to compute exactly $O(n^2(L + 2))$ values $f(j, k, l)$ and computation of each value takes $O(L)$ time, where $L = O((n/\epsilon) \log \sum_{j=1}^n e_j)$. Hence, the running time of Algorithm $\mathcal{A}_\epsilon$ is $O(n^2 L^2) = O((n^4/\epsilon^2) \log^2 \sum_{j=1}^n e_j)$. $\square$

3.4. An Optimal Algorithm of One Special Case. The special case where all jobs have the same release dates is considered in this subsection, that is, the problem 1| p -batch, $r_j = t_0$, $p_j = b_j t$, rej , $b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$.

We first reindex the jobs in nonincreasing order of their deteriorating rates such that $b_1 \geq \dots \geq b_n$.

Lemma 10. *For problem 1| p -batch, $r_j = t_0$, $p_j = b_j t$, rej , $b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$, in any optimal schedule, if job J_j ($1 \leq j \leq n$) is accepted, then the jobs set $\{J_i : i \geq j\}$ is accepted too.*

Proof. If job J_j is accepted, since $b \geq n$, there is no limit on the number of jobs in a batch; the jobs $\{J_{j+1}, \dots, J_n\}$ can be accepted and assigned to the current last batch. Obviously, this does not increase the value of the objective function. This completes the proof. $\square$

Theorem 11. *For problem 1| p -batch, $r_j = t_0$, $p_j = b_j t$, rej , $b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$, the optimal function value is $\min\{t_0(1 + b_1), \sum_{j=1}^n e_j, \min_{2 \leq j \leq n} \{t_0(1 + b_j) + \sum_{i=1}^{j-1} e_i\}\}$; the optimal solution can be found in $O(n \log n)$ time.*

4. Conclusion

In this paper, we focus on the unbounded parallel batch scheduling of deteriorating jobs with release date and rejection. The objective is to minimize the makespan of the accepted jobs plus the total penalty of rejected jobs. We show that it is binary NP-hard; we provide two pseudopolynomial time algorithms and also propose a fully polynomial-time approximation scheme for the problem 1| p -batch, r_j , $p_j = b_j t$, rej , $b \geq n|C_{\max}(\bar{R}) + \sum_{J_j \in R} e_j$. Furthermore, we show that this problem can be solvable in $O(n \log n)$ time when jobs have identical release dates. For future research, it would be interesting to focus on scheduling deteriorating jobs with other objectives. Analysis of the other models of scheduling with rejection is also an interesting research direction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (11201259), the Special Funds of the National Natural Science Foundation of China (61340045),

and the Doctoral Fund of the Ministry of Education (20123705120001, 20123705110003).

References

- [1] C. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Operations Research*, vol. 40, no. 4, pp. 764–775, 1992.
- [2] P. Brucker, A. Gladky, H. Hoogeveen et al., "Scheduling a batching machine," *Journal of Scheduling*, vol. 1, no. 1, pp. 31–54, 1998.
- [3] T. C. E. Cheng, Z.-H. Liu, and W.-C. Yu, "Scheduling jobs with release dates and deadlines on a batch processing machine," *IIE Transactions*, vol. 33, no. 8, pp. 685–690, 2001.
- [4] X. Deng, C. K. Poon, and Y. Zhang, "Approximation algorithms in batch processing," in *Algorithms and Computation (Chennai, 1999)*, vol. 1741 of *Lecture Notes on Computer Science*, pp. 153–162, Springer, Berlin, Germany, 1999.
- [5] X. Deng and Y. Z. Zhang, "Minimizing mean response time for batch processing systems," in *Computing and Combinatorics*, vol. 1627 of *Lecture Notes on Computer Science*, pp. 231–240, 1999.
- [6] Z. Liu and W. Yu, "Scheduling one batch processor subject to job release dates," *Discrete Applied Mathematics*, vol. 105, no. 1–3, pp. 129–136, 2000.
- [7] Z. H. Liu, J. J. Yuan, and T. C. E. Cheng, "On scheduling an unbounded batch machine," *Operations Research Letters*, vol. 31, no. 1, pp. 42–48, 2003.
- [8] P. Brucker and S. Knust, *Complexity Results for Scheduling Problems*, 2009, <http://www.informatik.uni-osnabrueck.de/knust/class/>.
- [9] S. Browne and U. Yechiali, "Scheduling deteriorating jobs on a single processor," *Operations Research*, vol. 38, no. 3, pp. 495–498, 1990.
- [10] G. Mosheiov, "Scheduling jobs under simple linear deterioration," *Computers & Operations Research*, vol. 21, no. 6, pp. 653–659, 1994.
- [11] B. Alidaee and N. K. Womer, "Scheduling with time dependent processing times: review and extensions," *Journal of the Operational Research Society*, vol. 50, no. 7, pp. 711–720, 1999.
- [12] T. C. E. Cheng, Q. Ding, and B. M. T. Lin, "A concise survey of scheduling with time-dependent processing times," *European Journal of Operational Research*, vol. 152, no. 1, pp. 1–13, 2004.
- [13] S. Li, C. T. Ng, T. C. E. Cheng, and J. Yuan, "Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan," *European Journal of Operational Research*, vol. 210, no. 3, pp. 482–488, 2011.
- [14] C. X. Miao, Y. Z. Zhang, and C. L. Wu, "Scheduling of deteriorating jobs with release dates to minimize the maximum lateness," *Theoretical Computer Science*, vol. 462, pp. 80–87, 2012.
- [15] J. Zou, Y. Z. Zhang, and C. X. Miao, "Uniform parallel-machine scheduling with time dependent processing times," *Journal of Operations Research Society of China*, vol. 1, pp. 239–252, 2013.
- [16] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, and L. Stougie, "Multiprocessor scheduling with rejection," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 1, pp. 64–78, 2000.
- [17] H. Hoogeveen, M. Skutella, and G. J. Woeginger, "Preemptive scheduling with rejection," *Mathematical Programming. A Publication of the Mathematical Programming Society*, vol. 94, no. 2–3, Ser. B, pp. 361–374, 2003.

- [18] L. Lu, L. Zhang, and J. Yuan, "The unbounded parallel batch machine scheduling with release dates and rejection to minimize makespan," *Theoretical Computer Science*, vol. 396, no. 1–3, pp. 283–289, 2008.
- [19] D. W. Engels, D. R. Karger, S. G. Kolliopoulos, S. Sengupta, R. N. Uma, and J. Wein, "Techniques for scheduling with rejection," *Journal of Algorithms*, vol. 49, no. 1, pp. 175–191, 2003.
- [20] D. Shabtay, N. Gaspar, and M. Kaspi, "A survey on offline scheduling with rejection," *Journal of Scheduling*, vol. 16, no. 1, pp. 3–28, 2013.
- [21] Y. Cheng and S. Sun, "Scheduling linear deteriorating jobs with rejection on a single machine," *European Journal of Operational Research*, vol. 194, no. 1, pp. 18–27, 2009.
- [22] S. Li and J. Yuan, "Parallel-machine scheduling with deteriorating jobs and rejection," *Theoretical Computer Science*, vol. 411, no. 40–42, pp. 3642–3650, 2010.
- [23] 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.
- [24] S. Gawiejnowicz, *Time-Dependent Scheduling*, Monographs in Theoretical Computer Science. An EATCS Series, Springer, Berlin, Germany, 2008.
- [25] S. Sengupta, "Algorithms and approximation schemes for minimum lateness/tardiness scheduling with rejection," in *Algorithms and Data Structures*, vol. 2748 of *Lecture Notes in Computer Science*, pp. 79–90, 2003.