AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances A; Vol. 54; N°1; pp 64-74
Submitted Jan. 2017; Revised March 15, 2017, Accepted April 15, 2016

Parallel-Batch Scheduling with Deterioration and Rejection on a Single Machine

*Cuixia Miao, ** Fanxiao Meng

*School of Mathematical Sciences, Qufu Normal University,
Qufu 273165, China (miaocuixia@126.com)
** School of Mathematical Sciences, Qufu Normal University,
Qufu 273165, China (819027149@qq.com)

Abstract

In this paper, we consider the bounded parallel-batch scheduling with deterioration and rejection on a single machine. A job is either rejected with a certain penalty having to be paid, or accepted and processed in batches on the single machine. The objective is to minimize the maximum completion time of the accepted jobs and the total penalty of the rejected jobs. We analyze the complexity of the problem, present a pseudo-polynomial time algorithm and a fully polynomial-time approximation scheme.

Key words

Batch scheduling, deterioration, rejection, pseudo-polynomial time algorithm, fully polynomial-time approximation scheme (FPTAS)

1. Introduction

Consider the following problem of bounded parallel-batch scheduling with deterioration and rejection. There are n independent non-preemptive deteriorating jobs $J=\{J_1, J_2,\dots, J_n\}=$ to be processed on a single batch machine. The actual processing time of job $J_j(j=1,2\dots,n)$ is $p_j= \alpha_j t$, where $\alpha_j(\geq 0)$ and t denote the deteriorating rate and starting time, respectively. $J_j(j=1,2\dots,n)$ has a rejection penalty e_j and a release date r_j. Without loss of generality, we assume that the jobs' parameters are integral, unless stated otherwise. Each job $J_j(j=1,2\dots,n)$ is either rejected with a rejection penalty e_j having to be paid, or accepted to be processed on the machine in batches. The machine can process up to b jobs simultaneously as a batch, and the processing time of the batch is equal to the longest time of any job in the batch. In the batch scheduling with deterioration, the deteriorating rate of a given batch B is defined as $\alpha(B)=\max\{\alpha_j: J_j \in B\}$. All jobs contained in the same batch start and complete at the same time. Once processing of a batch is initiated, it can not be interrupted and other jobs cannot be introduced into the batch until processing is completed. The objective is to minimize the maximum completion time of the accepted jobs and the total penalty of the rejected jobs. Following Gawiejnowicz (2008), we denote our problem as:

$$1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j ,$$

where "p-batch" means parallel-batch, "rej" and "S" denote rejection and the set of accepted jobs, respectively.

The model described above falls into three categories: parallel-batch scheduling, scheduling with deterioration and scheduling with rejection. The parallel-batch scheduling is motivated by burn-in operations in semiconductor manufacturing, see Lee et al. (1992) for more details of the background. By Brucker et al. (1998), there are 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., $b \geq n$. The extensive survey of different models and results were provided both by Potts and Kovalyov (2000). Afterwards, Yuan et al. (2009) gave some new results for this parallel-batch scheduling.

Scheduling with deteriorating job was first considered by Gupta and Gupta (1988), and Browne and Yechiali (1990). From then on, this scheduling model has been extensively studied. The monograph by Gawiejnowicz (2008) presents this scheduling from different perspectives and covers results and examples. Ji and Cheng (2009) and Liu et al. (2012) gave some new results for this scheduling.

In classical scheduling literatures, all jobs must be processed. In the practical applications, however, this may not be true. Due to the limited resources, the scheduler can have the option to reject some jobs. However, rejected jobs will incur rejection penalties. The scheduling with rejection was first considered by Bartal et al. (2000). They studied both the off-line and the on-line versions of scheduling with rejection on identical parallel machines, the objective is to minimize the maximum completion time of the accepted jobs and the total penalty of the rejected jobs. Subsequently, the scheduling with rejection has many results.

Cheng and Sun (2009) considered the scheduling with linear deteriorating jobs and rejection on a single machine, they gave the proofs of the NP-hardness and presented some pseudo-polynomial time algorithms and FPTASs for some objectives. Li and Yuan (2009) gave some results for the deteriorating job scheduling with rejection on parallel machines.

In this paper, we consider the bounded parallel-batch scheduling with deterioration and rejection on a single machine. The objective is to minimize the maximum completion time of the accepted jobs and the total penalty of the rejected jobs.

The rest of the paper is organized as follows. We present some preliminaries in Section 2. In Section 3, we analyze complexity of the problem, present a pseudo-polynomial time algorithm , and a fully polynomial-time approximation scheme for the case where the jobs have identical release dates is presented. We conclude the paper and suggest some new topics for future research in the last section.

2. Preliminaries

An algorithm is called a$(1+\varepsilon)$-approximation algorithm for a minimization problem if it produces a solution that is at most $1+\varepsilon$ times as big as the optimal value, running in time that is polynomial in the input size. A family of algorithms $\{A_s\}$ for a problem is called a polynomial time approximation scheme (PTAS, for short) if, for every $\varepsilon>0$ $\{A_s\}$ is a $(1+\varepsilon)$-approximation algorithm whose running time is polynomial in the input size. Furthermore, if the running time of every $\{A_s\}$ is bounded by a polynomial in the input size and $1/\varepsilon$, then the family is called a fully polynomial time approximation scheme (FPTAS for short). Algorithms that have running times polynomial in n and the maximum of the elements of the instance are called pseudo-polynomial time algorithms.

We list the following useful algorithm stated in Miao et al. (2011).

Algorithm FBLDR (Fully Batching Longest Deteriorating Rate)

Step 1: Re-index jobs in non-increasing order of

their deteriorating rates such that. $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n$.

Step 2: Form batches by placing jobs J_{jb+1}

through $J_{(j+1)b}$ together in the same batch, for $j = 1, 2, ..., \left\lfloor \frac{n}{b} \right\rfloor$.

Step 3: Schedule the batches in any arbitrary order.

The schedule contains at most $\left\lfloor \frac{n}{b} \right\rfloor + 1$ batches and all batches are full except possibly the last one, where $\left\lfloor \frac{n}{b} \right\rfloor$ denotes the largest integer less than $\frac{n}{b}$.

Lemma 1. (Cheng et al. 2009) Problem $1\left|p_j = \alpha_j t, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$ is NP-hard.

Lemma 2. (Woeginger 1999) For any $0 \le y \le 1$ and for any real $m \ge 1$, $(1+\frac{y}{m})^m \le 1+2y$ holds.

Lemma 3. (Mosheiov 1994) For the single machine scheduling problem $1\left|p_j = \alpha_j t\right|C_{\max}$, if a schedule $\pi = \{J_{[1]}, J_{[2]}, ..., J_{[n]}\}$, the starting time of job $J_{[1]}$ is t_0, then the makespan is $C_{\max} = t_0 \prod_{j=1}^{n} \left(1+\alpha_{[j]}\right)$.

The geometric rounding technique developed by Sengupta (2003) is stated as: For any $\varepsilon' > 0$, and $x \ge 1$, if $\left(1+\varepsilon'\right)^{k-1} < x < \left(1+\varepsilon'\right)^k$, then we define $\lfloor x \rfloor_{\varepsilon'} = \left(1+\varepsilon'\right)^{k-1}, \lceil x \rceil_{\varepsilon'} = \left(1+\varepsilon'\right)^k$. If x is an exact power of $1+\varepsilon'$, then $\lfloor x \rfloor_{\varepsilon'} = \lceil x \rceil_{\varepsilon'} = x$. Note that $\lceil x \rceil_{\varepsilon'} \le \left(1+\varepsilon'\right)^k x$ for any $x \ge 1$.

3. Mainresults

3.1 NP-hardness

From Lemma 1 and Zhang and Miao (2004), we can get the following theorem.

Theorem 1. Problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$ is NP-hard.

3.2 Pseudo-polynomial time algorithm

Theorem 2. For problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$,there exists an optimal schedule in which the accepted jobs are assigned to the machine by Algorithm FBLDR.

Assume that the jobs have been indexed so that $\alpha_1 \ge \alpha_2 \ge ... \ge \alpha_n$

Let $F_j(b_j, E)$ be the optimal value of the objective function satisfying the following conditions for problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$:

(i) The jobs in consideration are $\{J_1, ..., J_j\}$.

(ii)The number of processed jobs in the last batch is b_j. If there is no such batch, we set $b_j = 0$.

(iii)The total rejection penalty of rejected jobs is E.

To get $F_j(b_j, E)$, we distinguish two cases as follows.

Case 1. Job J_j is accepted

In this case, we distinguish two subcases.

Subcase 1.1. $b_j = 1$

In this subcase, job J_j has to start a new batch. The number of processed jobs in the last batch among $\{J_1, ..., J_{j-1}\}$ is 0 or b. Without loss of generality, let it be b. Then the maximum completion time of accepted jobs among $\{J_1, ..., J_{j-1}\}$ is $F_{j-1}(b, E) - E$, which is the starting time of job J_j. Then we have $F_j(b_j, E) = (1+\alpha_j)(F_{j-1}(b, E) - E) + E = (1+\alpha_j)F_{j-1}(b, E) - \alpha_j E$

Subcase 1.2. $b_j > 1$

In this subcase, job J_j can be assigned to the last batch which has existed, and the makespan does not change by inserting job J_j. Therefore,

$$F_j(b_j, E) = F_{j-1}(b_j - 1, E).$$

Case 2. Job J_j is rejected

Since J_j is rejected, the total rejection penalty is $E - e_j$ among $\{J_1, ..., J_{j-1}\}$. Therefore, we have $F_j(b_j, E) = F_{j-1}(b_j, E - e_j) + e_j$.

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

Algorithm DP1

Step 1: (Initialization) $F_1(1,0) = t_0(1+\alpha_j)$ and $F_1(b_1, E) = +\infty$ for $(b_1, E) \neq (1,0)$.

$F_1(0, e_1) = t_0 + e_1$ and $F_1(b_1, E) = +\infty$ for $(b_1, E) \neq (0, e_1)$.

Step 2: (Iteration)

If $b_j = 1$, $F_j(b_j, E) = \min\{(1+\alpha_j)F_{j-1}(b\ , E) - \alpha_j E,\ \ F_{j-1}(b_j, E-e_j) + e_j\}$

If $b_j > 1$, $F_j(b_j, E) = \min\{F_{j-1}(b_j - 1, E),\quad F_{j-1}(b_j, E-e_j) + e_j\}$

Step 3: (Solution) .

Define the optimal value

$$F^* = \min\left\{F_n(b_n, E):\ 0 \le b_n \le b, 0 \le E \le \sum_{j=1}^{n} e_j\right\}$$

Theorem 3. Algorithm DP1 solves problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$ in $O(nb\sum_{j=1}^{n} e_j)$ time.

Proof : The correctness of Algorithm DP1 is guaranteed by the above discussion. Clearly, we have $0 \le b_i \le b$ for $i = 1, 2, ..., n$ and $0 \le E \le \sum_{j=1}^{n} e_j$. Thus, the recursive function has at most $O(nb\sum_{j=1}^{n} e_j)$ states. Each iteration takes a constant time to execute. Hence, the running time of the Algorithm DP1 is $O(nb\sum_{j=1}^{n} e_j)$.

3.3 A Fully polynomial-time approximation scheme

In this subsection, motivated by Cheng and Sun (2009), we design an FPTAS for problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$ by considering the modified deteriorating rates of the scheduled jobs. The definition of the modified deteriorating rates involves the rounding technique stated in Section 2.

For any $\varepsilon > 0$, we define the modified deteriorating rates $\alpha_j' = \lceil 1+\alpha_j \rceil_{\varepsilon'} - 1$, where $\varepsilon' = \frac{\varepsilon}{2\left\lceil \frac{n}{b} \right\rceil}$ and $\left\lceil \frac{n}{b} \right\rceil$ denotes the smallest integer larger than equal to $\frac{n}{b}$. Let L_j denote the exponent of $1+\alpha_j'$, i.e., $1+\alpha_j' = (1+\varepsilon')^{L_j}$, then $L_j = \frac{\log\lceil 1+\alpha_j \rceil_{\varepsilon'}}{\log(1+\varepsilon')} = O\left(\frac{\left\lceil \frac{n}{b} \right\rceil \log(1+\alpha_j)}{\varepsilon} \right)$.

We propose a dynamic programming algorithm for problem $1\left| p_j = \alpha_j t, b < n, rej \right| C_{\max}(S) + \sum_{J_j \in \bar{S}} e_j$ with modified deteriorating rates α_j'.

Assume that the jobs have been indexed so that $\alpha_1' \geq \alpha_2' \geq \ldots \geq \alpha_n'$. We define $\varphi_k = (1+\varepsilon')^k$. Let $F_j(b_j, k)$ be the optimal value of the objective function satisfying the following conditions.

(i) The jobs in consideration are $\{J_1, \ldots, J_j\}$.

(ii)The number of processed jobs in the last batch is b_j. If there is no such batch, we set $b_j = 0$.

(iii)The maximum completion time of the accepted jobs is is $t_0 \varphi_k$.

To get $F_j(b_j, k)$, we distinguish two cases as follows.

Case 1. Job J_j is accepted

In this case, we distinguish two subcases.

Subcase 1.1. $b_j = 1$

In this subcase, job J_j has to start a new batch. The number of processed jobs in the last batch among $\{J_1, \ldots, J_{j-1}\}$ is 0 or b. Without loss of generality, let it be b. Then the maximum completion time of accepted jobs among $\{J_1, \ldots, J_{j-1}\}$ is $t_0 \varphi_{k-L_j}$, Thus, the objective value increased by $t_0 \alpha_j' \varphi_{k-L_j}$ .Therefore, $F_j(b_j, k) = F_{j-1}(b, k-L_j) + t_0 \alpha_j' \varphi_{k-L_j}$.

Subcase 1.2. $b_j > 1$

In this subcase, job J_j can be assigned to the last batch which has existed, and the makespan does not change by inserting job J_j. Thus,,

$$F_j\left(b_j,k\right)=F_{j-1}\left(b_j-1,k\right).$$

Case 2. Job J_j is rejected

Similar to the above discussion, we have $F_j\left(b_j,k\right)=F_{j-1}\left(b_j,k\right)+e_j$.

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

Algorithm DP2

Step 1: For any $\varepsilon>0$, set $\varepsilon'=\dfrac{\varepsilon}{2\left\lceil \frac{n}{b}\right\rceil}$.Given an instance I, we define a new instance I' by rounding α_j such that $\alpha_j{}'=\left\lceil 1+\alpha_j\right\rceil_{\varepsilon'}-1$.

Step 2: Index the jobs so that $\alpha_1{}'\geq\alpha_2{}'\geq\ldots\geq\alpha_n{}'$.

Step 3: (Dynamic Programming)

Step 3.1:(Initialization)

$F_1\left(1,L_1\right)=t_0\varphi_{L_1}$ and $F_1\left(b_1,k\right)=+\infty$ for $\left(b_1,k\right)\neq\left(1,L_1\right)$.

$F_1\left(0,0\right)=t_0+e_1$ and $F_1\left(b_1,k\right)=+\infty$ for $\left(b_1,k\right)\neq\left(0,0\right)$.

Step 3.2: (Iteration)

$$F_j\left(b_j,k\right)=\min\{F_{j-1}\left(b\ ,k-L_j\right)+t_0\alpha_j{}'\varphi_{k-L_j},\ \ F_{j-1}\left(b_j,k\right)+e_j\}$$

for $b_j=1$.

$$F_j\left(b_j,k\right)=\min\{F_{j-1}\left(b_j-1,k\right),\quad F_{j-1}\left(b_j,k\right)+e_j\}$$

for $b_j>1$.

Step 3*:* (Solution) .

Define the optimal value

$$\min\left\{F_n\left(b_n,k\right):\ 0\leq b_n\leq b,0\leq k\leq\sum_{j=1}^{n}L_j\right\}.$$

Let F'^* and F^* be the optimal value for problem $1\left|p_j=\alpha_j t, b<n, rej\right|C_{\max}(S)+\sum_{J_j\in\bar{S}} e_j$ with modified deteriorating rates and problem $1\left|p_j=\alpha_j t, b<n, rej\right|C_{\max}(S)+\sum_{J_j\in\bar{S}} e_j$, respectively.

Theorem 4. For any given $0\le\varepsilon\le 2$, $\frac{F'^*}{F^*}\le 1+\varepsilon$.

Proof : Let S be the set of accepted jobs and T be the set of batches scheduled by algorithm FBLDR from S . We $|T|=\left\lceil\frac{|S|}{b}\right\rceil\le\left\lceil\frac{n}{b}\right\rceil$,The objective function with modified deteriorating rate α_j' is

$$
\begin{aligned}
C'_{\max}(S)+\sum_{J_j\in\bar{S}} e_j &= t_0\prod_{B_i\in T}\left\lceil 1+\alpha\left(B_i\right)\right\rceil_{\varepsilon'}+\sum_{J_j\in\bar{S}} e_j \\
&\le t_0\left(1+\varepsilon'\right)^{\left\lceil\frac{|S|}{b}\right\rceil}\prod_{B_i\in T}\left(1+\alpha\left(B_i\right)\right)+\sum_{J_j\in\bar{S}} e_j \\
&\le t_0\left(1+\varepsilon'\right)^{\left\lceil\frac{n}{b}\right\rceil}\prod_{B_i\in T}\left(1+\alpha\left(B_i\right)\right)+\sum_{J_j\in\bar{S}} e_j .
\end{aligned}
$$

From Lemma 2, we have $\left(1+\varepsilon'\right)^{\left\lceil\frac{n}{b}\right\rceil}=\left(1+\frac{\frac{\varepsilon}{2}}{\left\lceil\frac{n}{b}\right\rceil}\right)^{\left\lceil\frac{n}{b}\right\rceil}\le 1+\varepsilon$ for $0\le\varepsilon\le 2$. Then

$$
C'_{\max}(S)+\sum_{J_j\in\bar{S}} e_j \le\left(1+\varepsilon\right)\left(t_0\prod_{B_i\in T}\left(1+\alpha\left(B_i\right)\right)+\sum_{J_j\in\bar{S}} e_j\right)=\left(1+\varepsilon\right)\left(C_{\max}\left(S\right)+\sum_{J_j\in\bar{S}} e_j\right),
$$

$\forall S\subset J$.

Note that $F'^*\le C'_{\max}(S)+\sum_{J_j\in\bar{S}} e_j$ for $\forall S\subset J$ and

$F'^{*} = \min_{S\subset J}\left\{C_{\max}(S)+\sum_{J_j\in\bar{S}} e_j\right\}$. Thus, $\frac{F'^{*}}{F^{*}} \le 1+\varepsilon$. From the discussion of the algorithm, we know that the time complexity is $O(nb\sum_{j=1}^{n} L_j) = O\left(\frac{nb\left\lceil \frac{n}{b}\right\rceil \sum_{j=1}^{n}\log(1+\alpha_j)}{\varepsilon}\right) = O\left(\frac{nb\left\lceil \frac{n}{b}\right\rceil \log\prod_{j=1}^{n}(1+\alpha_j)}{\varepsilon}\right) = O\left(\frac{n^2 b\left\lceil \frac{n}{b}\right\rceil \log(1+\alpha_{\max})}{\varepsilon}\right)$.

Where $\alpha_{\max} = \max\{\alpha_j : j = 1,2,...,n\}$.This completes the proof.

Conclusion

In this paper, we presented pseudo-polynomial time algorithm and FPTAS for problem $1\left|p_j = \alpha_j t, b < n, rej\right|C_{\max}(S)+\sum_{J_j\in\bar{S}} e_j$.

For future research, it is worth considering other objective such as $\sum_{J_j\in S} C_j + \sum_{J_j\in\bar{S}} e_j$. Another direction for future research is to consider the parallel-machine problems.

References

1. Y.Bartal, S.Leonardi,A. Marchetti-Spaccamela ,L. Stougie, Multiprocessor scheduling with rejection, ,2000, SIAM Journal of Discrete Mathematics, vol.13, pp.64-78.
2. P. Brucker, A. Gladky, H. Hoogeveen, M.Y. Kovalyov, C.N. Potts, T. Tautenhahn, S. van de Velde, Scheduling a batching machine, 1998, Journal of Scheduling, vol.1, pp. 31-54.
3. S.Browne, U.Yechiali, Scheduling deteriorating jobs on a single processor, 1990, Operations Research, vol.38, no.3, pp. 495-498.
4. T.C.E Cheng, Q. Ding, B.M.T. Lin, A concise survey of scheduling with time-dependent processing times, 2004, European Journal of Operational Research, vol.152, pp.1-13.
5. Y.S. Cheng, S.T. Sun, Scheduling linear deterirating jobs with rejection on a single machine, 2009,European Journal of Operational Research, vol.194, pp. 18-27.
6. S.Gawiejnowicz, Time-Dependent Scheduling, 2008, Monographs in Theoretical Computer Science An EATCS Series, Berlin-New York, vol18.
7. G.N.D. Gupta, S.K. Gupta, Single facility scheduling with nonlinear processing times, 1988, Computer and Ind Engineering, vol.14, pp. 387-393.

8. M. Ji, T.C.E. Cheng, Parallel machine scheduling of single linear deteriorating jobs, Theoretical Computer Science, vol.410, pp.3761-3768.
9. C.Y. Lee, R. Uzsoy, L.A. Martin-Vega, Efficient algorithms for scheduling semiconductor burn-in operations, 1992, Operations Research, vol.40, pp.764-775.
10. S.S. Li, J.J. Yuan, Single machine parallel-btach scheduling with deteriorating jobs, 2009, Theoretical Computer Science, vol.410, pp. 830-836.
11. M. Liu, F.F. Zheng, C.B. Chu, J.T. Zhang, An FPTAS for uniform machine scheduling to minimize makespan with linear deterioration, 2012, Journal of Combinatorial Optimization, vol.23, pp. 483-492.
12. G. Mosheiov, Scheduling jobs under simple linear deterioration, 1994, Computers and Operations Research, vol.21, pp.653-659.
13. C.X. Miao, Y.Z. Zhang, Z.G. Cao, Bounded parallel-batch scheduling on single and multi-machines for deteriorating jobs, 2011, Information Processing Letters, vol.111, pp. 798-803.
14. C.N. Potts, M.Y. Kovalyov, Scheduling with batching: A review, 2000, European Journal of Operational Research, vol. 120, pp.228-249.
15. S. Sengupta, Algorithms and approximation schemes for minimum lateness/tardiness scheduling with rejection, 2003, LNCS, vol.2748, pp.79-90.
16. G.J. Woeginger, When does a dynamic programming formulation guarantee the existence of an FPTAS? 1999, Proceeding of 10th ACM-SIAM Symposium on discrete Algorithms, pp.820-829.
17. J.J. Yuan, S.S. Li, R.Y. Fu, A best on-line algorithm for the single machine parallel-batch scheduling with restricted delivery times, 2009, Journal of Combinatorial Optimization, vol.7, pp.206-213.
18. Y.Z. Zhang, C.X. Miao, Copy method and its applications on the batching scheduling problems, 2004, Journal of Qufu Normal University, vol.2, pp. 41-43. (in Chinese)