European Journal of Operational Research 241 (2015) 653–661

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

An improved heuristic for parallel machine scheduling with rejection

Jinwen Ou[a], Xueling Zhong[b,*], Guoqing Wang[a]

[a] Department of Administrative Management, Jinan University, Guangzhou 510632, People's Republic of China
[b] Department of Computer, Guangdong University of Finance, Guangzhou 510520, People's Republic of China

ARTICLE INFO

Article history:
Received 22 September 2013
Accepted 7 September 2014
Available online 18 October 2014

Keywords:
Scheduling
Rejection
Heuristic
Worst-case bound

ABSTRACT

In this paper we study a classical parallel machine scheduling model with m machines and n jobs, where each job is either accepted and then processed by one of the machines, or rejected and then a rejection penalty is paid. The objective is to minimize the completion time of the last accepted job plus the total penalty of all rejected jobs. The scheduling problem is known to be NP-hard in the strong sense. We find some new optimal properties and develop an $O(n\log n + n/\epsilon)$ heuristic to solve the problem with a worst-case bound of $1.5 + \epsilon$, where $\epsilon > 0$ can be any small given constant. This improves upon the worst-case bound $2 - \frac{1}{m}$ of the heuristic presented by Bartal *et al.* (Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., & Stougie, L. (2000). Multiprocessor scheduling with rejection. *SIAM Journal on Discrete Mathematics, 13*, 64–78) in the scheduling literature.

1. Introduction

Machine scheduling with rejection (also named "order acceptance and scheduling") has attracted considerable attention from scheduling researchers as well as production managers who practice it in the past a few decades. One important application of scheduling with rejection arises in make-to-order production systems with limited production capacity or tight delivery requirements, where order rejection and scheduling decisions have to be made simultaneously (Cesaret, Oguz, & Salman, 2012). In practice, due to production capacity constraints and short delivery due dates, a typical manufacturer may have to reject some orders which have long processing times but contribute relatively small profits. Another important application of scheduling with rejection occurs in scheduling when outsourcing is an alternative option (Shabtay, Gaspar, & Kaspi, 2013). Scheduling with rejection combines outsourcing and production scheduling decisions together.

Machine scheduling with rejection (MSR) has been studied extensively in the scheduling literature. Most of the MSR models are considered in the single-machine environment (e.g., Zhang, Lu, & Yuan, 2009; Lu, Cheng, Yuan, & Zhang, 2009; Talla Nobibon & Leus, 2011; Oguz, Salman, & Yalcin, 2010; Lee & Sung, 2008; Slotnick & Morton, 1996; Slotnick & Morton, 2007; Zhong, Ou, & Wang, 2014, among others), or study on-line scheduling (e.g., Bartal, Leonardi, & Marchetti-Spaccamela, 2000; Seiden, 2001; Hoogeveen, Skutella, & Skutella, 2003; Epstein, Noga, & Woeginger, 2002, among others). MSR models with industrial applications are studied by Cheng and Sun (2009), Guerrero and Kern (1988), Cesaret et al. (2012), and Rom and Slotnick (2009), among others. Excellent literature review articles on MSR are provided by Slotnick (2011) and Shabtay et al. (2013) recently.

In this paper, we study a basic off-line parallel-machine MSR model. Bartal et al. (2000) are the first to introduce parallel machine scheduling models with rejection, where each job is either accepted and then processed by one of the machines, or rejected and then a rejection penalty is paid. The objective function is to minimize the makespan of all accepted jobs (i.e., the completion time of the last accepted job) plus the total rejection penalty of all rejected jobs. They study both the on-line model and the off-line model. For the on-line model, they present an on-line algorithm with the best-possible competitive ratio of 2.618. For the off-line model, they develop an $O(n\log n)$ heuristic with a worst-case bound of $2 - \frac{1}{m}$ (here n is the number of jobs, while m is the number of machines), and present a fully polynomial-time approximation scheme (FPTAS) for the case when m is fixed, and a polynomial-time approximation scheme (PTAS) for the case when m is arbitrary. The running time complexity of their FPTAS is $O((n/\epsilon)^m)$, while the running time complexity of their PTAS is $O((n^3/\epsilon)^{9/\epsilon^2})$, where $\epsilon > 0$ can be any small given constant. Note that such a PTAS has a running time complexity of $O(n^{27})$ when $\epsilon = 1$. Thus, it is interesting to develop efficient heuristics with a better performance ratio to solve the off-line model, especially for the general case when m is arbitrary. Surprisingly, in terms of developing fast heuristics, the worst-case bound of $2 - \frac{1}{m}$ for the general off-line

* Corresponding author. Tel.: +86 20 3820 2796.
E-mail addresses: toujinwen@jnu.edu.cn (J. Ou), zhongxuel@hotmail.com (X. Zhong), tgqwang@jnu.edu.cn (G. Wang).

http://dx.doi.org/10.1016/j.ejor.2014.09.028

model in Bartal et al. (2000) is the best known result in the scheduling literature (see the very recent review papers Slotnick (2011); Shabtay et al. (2013)). In this paper, we study the off-line model in Bartal et al. (2000) for the general case when m is arbitrary, and present an $O(n\log n + n/\epsilon)$ heuristic with a worst-case bound of $1.5+\epsilon$.

The scheduling problem we study can be described formally as follows: We are given a set of m identical parallel machines $M = \{M_1, M_2, \ldots, M_m\}$ and a set of $n > m$ jobs $J = \{J_1, J_2, \ldots, J_n\}$. Associated with each job J_j is a processing time $p_j > 0$ and a rejection penalty $w_j > 0$. Job J_j is either rejected and then a rejection penalty w_j is paid, or accepted and then processed by one of the m machines. Job preemption is not allowed during job processing, and each machine is available for processing jobs at time zero. Each machine can process at most one job at a time. Denote $A \subseteq J$ as the set of jobs to be accepted, and $R = J \setminus A$ as the set of jobs to be rejected. Denote C_j as the completion time of job J_j on a machine if $J_j \in A$. The objective is to determine A and a feasible schedule for the jobs in A on the m machines, so as to minimize objective function $Z = \max_{J_j \in A}\{C_j\} + \sum_{J_j\in R} w_j$, i.e., the makespan of all accepted jobs plus the total rejection penalty of all rejected jobs. We denote $P|rej|C_{\max} + \sum w_j$ as the scheduling problem.

Note that the classical parallel machine scheduling problem $P||C_{\max}$ is a special case of problem $P|rej|C_{\max} + \sum w_j$ when $w_j = +\infty$ for $j = 1, 2, \ldots, n$. Problem $P||C_{\max}$ is known to be NP-hard in the strong sense (Garey & Johnson, 1979). Therefore, $P|rej|C_{\max} + \sum w_j$ is also NP-hard in the strong sense. It is also well-known that problem $P||C_{\max}$ can be solved by some simple heuristics with constant worst-case bounds. For examples, the *list schedule* has a worst-case bound of $2 - \frac{1}{m}$, and the *longest-processing-time-first* (LPT) rule has a worst-case bound of $\frac{4}{3} - \frac{1}{3m}$ (see Pinedo, 2012 for more details and more heuristics). However, when job rejection is allowed, it becomes much more complicated to design efficient heuristics with low running time complexity and good worst-case bounds. In this paper we focus on developing an efficient heuristic to solve problem $P|rej|C_{\max} + \sum w_j$ with near-linear time complexity.

2. Heuristic

We will present a heuristic **H** to solve problem $P|rej|C_{\max} + \sum w_j$ in this section. Heuristic **H** is shown to have a worst-case bound of $1.5+\epsilon$ and running time complexity of $O(n\log n + n/\epsilon)$.

Denote σ^* as the optimal solution, A^* as the set of all accepted jobs in σ^*, $R^* = J \setminus A^*$ as the set of all rejected jobs in σ^*, $C^*_{\max}$ as the completion time of the last accepted jobs in A^* (i.e., the makespan of all accepted jobs) in σ^*, and $Z^* = C^*_{\max} + \sum_{J_j\in R^*} w_j$ as the objective function value of σ^*. Let $\mathbf{H}_0$ be the $O(n\log n)$ heuristic presented by Bartal et al. (2000), and let Z_0 be the value of the solution generated by $\mathbf{H}_0$. In our heuristic **H**, we first apply $\mathbf{H}_0$ to generate solution value Z_0. By Theorem 5.1 on page 73 in Bartal et al. (2000), we have

$$Z^* \le Z_0 \le \left(2 - \frac{1}{m}\right) Z^*. \tag{1}$$

According to Bartal et al. (2000), we also have the following property.

Lemma 1. *If $C^*_{max} \le \frac{1}{2}Z^*$, then $Z_0 \le 1.5Z^*$.*

Proof. By the proof of Theorem 5.1 on page 73 in Bartal et al. (2000), we have

$$Z_0 \le Z^* + \left(1 - \frac{1}{m}\right) C^*_{\max}.$$

Thus, if $C^*_{\max} \le \frac{1}{2}Z^*$, then

$$Z_0 \le Z^* + \left(1 - \frac{1}{m}\right)\cdot C^*_{\max} \le Z^* + \left(1 - \frac{1}{m}\right)\cdot \frac{1}{2}\cdot Z^* \le 1.5Z^*. \qquad \square$$

By Lemma 1, to make sure heuristic **H** having a worst-case bound of $1.5+\epsilon$, we only need to consider the situation when

$$C^*_{\max} > \frac{1}{2}Z^* > \frac{1}{4}Z_0. \tag{2}$$

In the remaining of this section, we only consider the situation when inequality (2) is satisfied.

Let ϵ be any small constant such that $\frac{1}{\epsilon}$ is a positive integer, and let $\epsilon' = \frac{\epsilon}{3}$. For $t = 0, 1, \ldots, \frac{1}{\epsilon'}$, we define

$$\hat{C}_t = t\cdot(\epsilon'\cdot Z_0).$$

Clearly, by (1) and (2), there exists some integer $\kappa \in \{1, 2, \ldots, \frac{1}{\epsilon'}\}$ such that

$$\hat{C}_{\kappa-1} = (\kappa - 1)\cdot(\epsilon'\cdot Z_0) < C^*_{\max} \le \kappa\cdot(\epsilon'\cdot Z_0) = \hat{C}_\kappa. \tag{3}$$

By (1) and (3), $\hat{C}_\kappa$ is an upper bound of $C^*_{\max}$ such that

$$\begin{aligned} C^*_{\max} \le \hat{C}_\kappa &= (\kappa-1)\cdot\epsilon'\cdot Z_0 + \epsilon'\cdot Z_0 < C^*_{\max} + \epsilon'\cdot(2Z^*) \\ &= C^*_{\max} + \frac{2}{3}\cdot\epsilon\cdot Z^*. \end{aligned} \tag{4}$$

Note that the exact value of κ is unknown. In heuristic **H** we need to consider all of the $\frac{1}{\epsilon'}$ values $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_{\frac{1}{\epsilon'}}$.

We now introduce the major idea flow of heuristic **H**. In **H** we need to determine the $\frac{1}{\epsilon'}$ values $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_{\frac{1}{\epsilon'}}$. Provided $\hat{C}_t$ ($t = 1, 2, \ldots, \frac{1}{\epsilon'}$), we will determine value $\tilde{Z}_t$, where $\tilde{Z}_t$ is an upper bound of the objective function value of some feasible solution that can be constructed based on the value of $\hat{C}_t$. As what we will show it later, each value $\tilde{Z}_t$ can be determined in $O(n)$ time, and the value of $\tilde{Z}_\kappa$ is bounded by $(1.5+\epsilon)Z^*$ (remember that $\hat{C}_\kappa$ is an upper bound of $C^*_{\max}$ satisfying (4), but we do not know the exact value of κ, although it exists). We then find out the index $\xi \in \{1, 2, \ldots, \frac{1}{\epsilon'}\}$ such that

$$\tilde{Z}_\xi = \min\left\{\tilde{Z}_t \mid t = 1, 2, \ldots, \frac{1}{\epsilon'}\right\}.$$

We have

$$\tilde{Z}_\xi \le \tilde{Z}_\kappa \le (1.5+\epsilon)Z^*.$$

Finally, provided ξ and $\hat{C}_\xi$, a feasible solution with a solution value no more than $\tilde{Z}_\xi$ is generated by **H**. Thus, the generated solution value by **H** is bounded by $\tilde{Z}_\xi \le (1.5+\epsilon)Z^*$. Here we would like to point out that it is unnecessary to know what is the exact value of κ as long as we are able to determine $\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_{\frac{1}{\epsilon'}}$ and the value of ξ.

It is challenging in **H** to determine value $\tilde{Z}_t$ for each $t = 1, 2, \ldots, \frac{1}{\epsilon'}$ in $O(n)$ time. The major idea of how to determine $\tilde{Z}_t$ is as follows. Provided the value of $\hat{C}_t$, we will find out a job set $A'_t \subseteq J$ such that all jobs in A'_t can be scheduled on the m machines with a makespan no more than $1.5\hat{C}_t$ (the total processing time of jobs in A'_t is approximately equal to $m\cdot\hat{C}_t$). To reduce total rejection penalty, we give priority to those jobs with a high value of w_j/p_j or w_j when selecting jobs to A'_t. The value of $\tilde{Z}_t$ is given to be

$$\tilde{Z}_t = 1.5\hat{C}_t + \sum_{J_j\in J\setminus A'_t} w_j.$$

Clearly, provided job set A'_t, there exists some feasible solution $\hat{\sigma}_t$ in which all jobs in A'_t are accepted while all jobs in $J\setminus A'_t$ are rejected, and the corresponding solution value of $\hat{\sigma}_t$ is no more than $\tilde{Z}_t$. As what we will show it later, when $\hat{C}_t = \hat{C}_\kappa$, the generated job set A'_κ satisfies

$$\sum_{J_j\in J\setminus A'_\kappa} w_j \le 1.5\sum_{J_j\in R^*} w_j.$$

This indicates that there exists a feasible solution $\hat{\sigma}_\kappa$ in which all jobs in A'_κ are accepted and scheduled on the machines with a makespan