Theoretical Computer Science 616 (2016) 94–99

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Scheduling with job rejection and nonsimultaneous machine available time on unrelated parallel machines

Dakui Jiang [a,*], Jiayin Tan [b]

[a] College of Management and Economics, Tianjin University, Tianjin 300072, China
[b] School of Management Science and Engineering, Nanjing Audit University, Nanjing, 211815, China

ARTICLE INFO

Article history:
Received 12 June 2015
Received in revised form 28 October 2015
Accepted 15 December 2015
Available online 22 December 2015
Communicated by D.-Z. Du

Keywords:
Scheduling
Unrelated machine
Rejection
Machine availability constraint
Heuristic

ABSTRACT

An unrelated-parallel-machine scheduling model with job rejection option was studied. In this model, all jobs are available at time zero but some machines are not available at time zero. Each job is either accepted and processed once without interruption, or is rejected at a penalty cost. To minimize the makespan of all accepted jobs plus the total cost of rejecting and processing jobs, we provide a heuristic with worst-case ratio bound of 2.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Scheduling problems have been extensively studied in the literature under the assumption that all jobs have to be processed. However, in realty, the manufacturers always reject part of jobs and process the remaining ones so as to maximize the total revenue [12]. Hence, machine scheduling with rejection (MSR) has attracted considerable attention from scheduling researchers as well as operation managers in the past few decades. In traditional MSR models, machines are assumed to be available at time zero of the planning horizon. However, in many practical cases, some machines are actually not available at time zero due to some reasons such as the machines are in maintenance or occupied by other tasks. In this paper, we deal with an unrelated-parallel-machine MSR model with nonsimultaneous machine available time. In this proposed model, job rejection is allowed and some machines are not available at time zero. When a job is rejected, a job-dependent penalty will occur. The manufacturer needs to balance the makespan of completing all accepted jobs and the total cost of rejecting and processing jobs.

MSR was first introduced by Bartal et al. [1]. They considered the model with identical-parallel-machine configuration, the objective of which is to minimize the sum of the makespan of all accepted jobs and total penalty of the rejected jobs. Hoogeveen et al. [4] investigated a special case that job preemption is allowed. After that, Lu et al. [8,9], and Zhang et al. [13] studied the models with release dates on a single machine. In these models, all machines are available at time zero but some jobs are not available at time zero. However, all of the MSR models mentioned above have the potential assumption that all machines are available at time zero, which differ from our model.

* Corresponding author.
E-mail address: jiangdakui@tju.edu.cn (D. Jiang).

http://dx.doi.org/10.1016/j.tcs.2015.12.020
0304-3975/© 2015 Elsevier B.V. All rights reserved.

Among the literature on machine scheduling models, Lee [6] was the first to study the model with nonsimultaneous machine available time. He assumed that some machines may not be available at time zero of the planning horizon. Two approximation algorithms were proposed to solve the problem with m identical parallel machines, the worst-case ratio of which are $3/2 - 1/2m$ and $4/3$ respectively. Kellerer [5] and Chang and Hwang [2] improved the bound to $5/4$ and $9/2 + 2^{-k}$ respectively, where k is the number of the major iterations in the algorithm. As is known to all, machines are unrelated which is the general case of parallel machine environments. However, according to the survey provided by Ma et al. [10], little attention is paid to unrelated-parallel-machine configuration in the literature on scheduling with nonsimultaneous machine available time.

In this paper, we consider an MSR model on unrelated parallel machines with nonsimultaneous machine available time, which is represented by P in the remainder of this paper. We now describe problem P formally as follows. Given a set of n independent jobs, $N = \{1, 2, \dots, n\}$, and a set of m unrelated parallel machines, $M = \{1, 2, \dots, m\}$. All jobs are available at time 0, but machine i is available at time a_i, for $i \in M$. Without loss of generality, we assume that $a_1 \geq a_2 \geq \cdots \geq a_m$. Each job $j \in N$ is either accepted and then processed once by one of the machines without interruption, or is rejected at w_j units of rejection penalty cost. It takes p_{ij} units of processing time and c_{ij} units of production cost for machine i to process job j, for $i \in M$ and $j \in N$. The problem is to determine which jobs to accept and how to schedule them so as to minimize the makespan of processing all accepted jobs plus the total cost of rejecting and processing jobs.

Problem P is NP-hard in the strong sense, since the special case with $a_i = 0$, $c_{ij} = 0$, and $w_j = +\infty$, for $i \in M$ and $j \in N$, is equivalent to the classical unrelated parallel machine makespan minimization scheduling problem, which is known to be NP-hard in the strong sense [11]. Therefore, we will design a heuristic for problem P and analyze its worst-case performance in the following section.

2. Heuristic algorithm

In this section, we present a heuristic for problem P. The heuristic generalizes the two phase approach proposed by Potts [11], which schedules the majority of jobs by solving a linear program in the first phase and then applies an exact algorithm to schedule the remaining jobs in the second phase. Potts [11] adopted this approach to solve the unrelated parallel machine makespan minimization scheduling problem. In that problem, all machines are available at time zero of the planning horizon, which make it easy to design the first phase by constructing only one linear program. However, problem P is more general, which allows that some machines are not available at time zero of the planning horizon. Thus, it is possible to have machines that will not process any jobs to be assigned in current planning horizon [7]. For example, if a_i is a very large number, then machine i may not process any job and hence is inactive in the current horizon. Therefore, problem P is more complex and design the first phase of the heuristic will not be easy.

In the following, we first introduce the first phase of the heuristic. To obtain an optimal solution, we need to find a subset $Q \subseteq M$ that not only all accepted jobs are scheduled at the machines in Q but also at least one job should be scheduled at each machine in Q. Then, we formulate problem P as follows:

$$P : Z = \min\left\{\sum_{j\in N} w_j, \min\left\{Z(Q) \middle| Q \subseteq M, Q \neq \emptyset\right\}\right\}, \tag{1}$$

where

$$P(Q) : Z(Q) = \min C_{max} + \sum_{j\in N} w_j x_j + \sum_{i\in Q}\sum_{j\in N} c_{ij} y_{ij} \tag{2}$$

$$\text{s.t. } C_{max} \geq a_i + \sum_{j\in N} p_{ij} y_{ij} \quad \forall i \in Q, \tag{3}$$

$$\sum_{j\in N} y_{ij} \geq 1 \quad \forall i \in Q, \tag{4}$$

$$x_j + \sum_{i\in Q} y_{ij} = 1 \quad \forall j \in N, \tag{5}$$

$$x_j \in \{0, 1\}, y_{ij} \in \{0, 1\} \quad \forall i \in Q, \forall j \in N. \tag{6}$$

In $P(Q)$, binary variable $x_j = 1$ if job j is rejected, and 0 otherwise; binary variable $y_{ij} = 1$ if job j is assigned to machine i, and 0 otherwise; and variable C_{max} is the makespan of all accepted jobs, which is defined by constraints (3). Constraints (4) imply that at least one job is scheduled at each machine in Q. Constraints (5) ensure that each job is either rejected or processed on one of the machines in Q.

In the following, we construct a relaxation model of P, denoted as LP.

$$LP : Z_{LP} = \min\left\{\sum_{j\in N} w_j, \min\left\{Z_{LP}(Q) \middle| Q \subseteq M, Q \neq \emptyset\right\}\right\},$$

Download English Version:

https://daneshyari.com/en/article/437493

Download Persian Version:

https://daneshyari.com/article/437493

Daneshyari.com