Computers & Industrial Engineering 74 (2014) 88–91

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Short Communication

Parallel machines scheduling with simple linear job deterioration and non-simultaneous machine available times ☆

Xiao-Yuan Wang [a,b,*], Zhili Zhou [a], Ping Ji [c], Ji-Bo Wang [b,c,d]

[a] School of Management, Xi'an Jiaotong University, Xi'an 710049, China
[b] School of Science, Shenyang Aerospace University, Shenyang 110136, China
[c] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[d] State Key Laboratory for Manufacturing Systems Engineering (Xi'an Jiaotong University), Xi'an 710053, China

ARTICLE INFO

Article history:
Received 23 July 2013
Received in revised form 5 April 2014
Accepted 7 May 2014
Available online 14 May 2014

Keywords:
Scheduling
Parallel machines
Deteriorating jobs
Worst-case analysis
Machine availability constraint

ABSTRACT

In this note, we deal with the parallel-machine scheduling of deteriorating jobs where the actual processing of a job is a simple linear function of the job's starting time. The jobs are available at time zero but the machines may not be available simultaneously at time zero. For the problem to minimize the logarithm of makespan, we present heuristic algorithms and analyze their worst-case error bounds.

1. Introduction

For most traditional scheduling problems, the job processing times are fixed constant. This assumption, however, may not be valid for the modelling of realistic industrial processes. So there is a growing interest in scheduling problems with deteriorating job processing times (deteriorating jobs). Job deterioration occurs, e.g., in scheduling maintenance jobs, steel industrial or cleaning assignments, where any delay (waiting) in processing causes an increase of the processing times of executed jobs. Extensive surveys of scheduling models and problems with time dependent processing times (deteriorating jobs) can be found in Alidaee and Womer (1999) and Cheng, Ding, and Lin (2004). Recently, Gawiejnowicz (2008) provides details on scheduling problems with time-dependent processing times (deteriorating jobs).

Kononov (1997) and Mosheiov (1998) proved that the parallel-machine makespan minimization problem with simple linear job deterioration is NP-hard even for $m = 2$, where m is the number of machines. Chen (1996) and Kononov (1997) proved that the parallel-machine total completion time minimization problem with simple linear job deterioration is NP-hard even for $m = 2$. Jeng and Lin (2007) considered a parallel-machine total completion minimization scheduling problem with deteriorating job processing times. They derived a lower bound for the general m-machine case. Kuo and Yang (2008) studied a parallel-machine scheduling problem with deteriorating job processing times. They showed that the total completion time minimization problem and the total load on all the machines minimization problem is polynomially solvable, respectively. Ji and Cheng (2008) considered a parallel-machine total completion time minimization scheduling problem with simple linear deterioration. When m is fixed, they give a fully polynomial-time approximation scheme (FPTAS) for the case with m identical machines. Ji and Cheng (2009) considered scheduling problems on parallel machines with simple linear deterioration. They showed that the makespan minimization, the total machine load minimization, and the total completion time minimization problems are strongly NP-hard (NP-hard in the ordinary sense) when the number of machines is arbitrarily (when the number of machines is fixed). Cheng, Wang, and He (2009) proved that the machine minimum completion time maximization problem with simple linear job deterioration is NP-hard even for $m = 2$. Mazdeh, Zaerpour, Zareei, and Hajinezhad (2010) considered bicriterion parallel-machine scheduling with job deterioration to minimize job tardiness and machine deteriorating cost. They proposed a heuristic algorithm to locate the optimal or near-optimal solutions. Li and Yuan (2010) considered several scheduling problems with deteriorating jobs and rejection. Zhang and Luo (2013) considered the parallel-machine scheduling problems with

☆ This manuscript was processed by Area Editor Subhash C. Sarin.
* Corresponding author at: School of Science, Shenyang Aerospace University, Shenyang 110136, China. Tel./fax: +86 24 89723548.
E-mail address: wxy5099@126.com (X.-Y. Wang).

http://dx.doi.org/10.1016/j.cie.2014.05.003
0360-8352/

deteriorating jobs, rejection and the first machine has a fixed non-availability interval.

In the manufacturing industry, due to the preventive maintenance and adjustment, machines should be prepared before they can start processing, and the time used for preparing these machines is different, this phenomenon is known as the "machine available time" in the literatures (Lee, 1991; Shen, Wang, & Wang, 2013; Sun & Huang, 2010). In this paper we consider parallel-machine scheduling of deteriorating jobs where the machines may not be available simultaneously at time $a > 0$, and the objectives are to minimize the logarithm of makespan, i.e., the logarithm of the maximum completion time. In scheduling of deteriorating jobs, the completion times of the jobs that are processed late may be excessively long. So it makes sense to adopt performance measures that are logarithmic functions of job completion times. To the best of our knowledge, Cheng et al. (2009) first considered parallel-machine scheduling with deteriorating jobs to minimize the logarithm of makespan and to maximize the logarithm of minimum completion time.

The remainder of this note is organized as follows: we formulate the model in Section 2. In Section 3 we study the problem to minimize the logarithm of makespan. In the last section we conclude the paper and suggest some topics for future research.

2. Problem description

The problem under investigation can be described as follows: There are n independent jobs $N = \{J_1, J_2, \dots, J_n\}$ to be processed on m parallel identical machines $\{M_1, M_2, \dots, M_m\}$. As in Lee (1991) and Shen et al. (2013), let $a_i > 0$ denote the earliest time that machine M_i can start processing jobs. Without loss of generality, we assume that the machines are indexed in nondecreasing order of their ready times, i.e., $0 < a_1 \leqslant a_2 \leqslant \cdots \leqslant a_m$. The machines can only handle one job at a time and preemption is not allowed. We also assume that all the jobs are ready at time zero. A job cannot be processed on two or more machines at the same time. Let p_j be the actual processing time of job J_j, as in Mosheiov (1994), we consider the following model:

$$p_j = b_j t, \tag{1}$$

where $b_j > 0$ is the deterioration rate of J_j and t is the starting time of J_j. Let n_i be the number of jobs processed on machine M_i, so $n_1 + n_2 + \cdots + n_m = n$; A^i be the job set processed on machine $M_i, i = 1, 2, \dots, m$; $\log C_{\max}^{(i)} = \log\left[a_i \prod_{l=1}^{n_i}\left(1 + b_l^{(i)}\right)\right] = \log a_i + \sum_{l=1}^{n_i} \log(1 + b_l)$ be the logarithm of the makespan of machine M_i; $\log C_{\max} = \max_{1 \leqslant i \leqslant m}\left\{\log C_{\max}^{(i)}\right\}$ be the logarithm of makespan (where $\log x \equiv \log_2 x$). Our objective is to determine an optimal schedule to minimize the logarithm of makespan. Following the standard three-field notation used in scheduling problem (Graham, Lawler, Lenstra, & Rinnooy Kan, 1979), we denote our problem as $Pm, a_i | p_j = b_j t | \log C_{\max}$.

3. The logarithm of makespan minimization problem

In this section we consider the problem $Pm, a_i | p_j = b_j t | \log C_{\max}$. For the case $a_i = a$, Kononov (1997) and Mosheiov (1998) prove that the problem $Pm | p_j = b_j t | C_{\max}$ is NP-hard. Note that the classical problem $Pm | p_j = d_j | C_{\max}$ is NP-hard. Letting $a_i = 1$ and $\log(1 + b_j) = d_j$, we deduce that the problem $Pm, a_i | p_j = b_j t | \log C_{\max}$ is NP-hard, too.

We first apply the list scheduling (LS) algorithm to the problem $Pm, a_i | p_j = b_j t | \log C_{\max}$, i.e., whenever a machine is freed, the job with the lowest index among an arbitrary list of jobs is assigned to that machine. Let $\log C_{\max}(\mathrm{LS})$ denote the logarithm of makespan obtained by applying the LS algorithm to the problem $Pm, a_i | p_j = b_j t | \log C_{\max}$ and $\log C_{\max}(\mathrm{OPT})$ denote the logarithm of makespan obtained by the optimal schedule. We have the following result:

Theorem 1. $\frac{\log C_{\max}(\mathrm{LS})}{\log C_{\max}(\mathrm{OPT})} \leqslant 2 - \frac{1}{m}$ *and the bound is tight.*

Proof. We assume that the job J_k is the last completed job. At the time when some machine begins processing job J_k, all other machines are either still busy or just free, we have

$$\begin{aligned}
&\log C_{\max}(\mathrm{LS}) \\
&= \log \min\left\{a_1 \prod_{l \in A^1 \setminus J_k} (1 + b_l), a_2 \prod_{l \in A^2 \setminus J_k} (1 + b_l), \dots, a_m \prod_{l \in A^m \setminus J_k} (1 + b_l)\right\} \\
&\quad + \log(1 + b_k) \\
&= \min\left\{\log a_1 + \sum_{l \in A^1 \setminus J_k} \log(1 + b_l), \log a_2 + \sum_{l \in A^2 \setminus J_k} \log(1 + b_l), \dots,\right. \\
&\qquad \left.\log a_m + \sum_{l \in A^m \setminus J_k} \log(1 + b_l)\right\} + \log(1 + b_k) \\
&\leqslant \frac{1}{m}\left[\log a_1 + \sum_{l \in A^1 \setminus J_k} \log(1 + b_l) + \log a_2 + \sum_{l \in A^2 \setminus J_k} \log(1 + b_l) + \cdots +\right. \\
&\qquad \left.\log a_m + \sum_{l \in A^m \setminus J_k} \log(1 + b_l)\right] + \log(1 + b_k) \\
&= \frac{1}{m}\left[\sum_{i=1}^{m} \log a_i + \sum_{l \in N \setminus J_k} \log(1 + b_l)\right] + \log(1 + b_k) \\
&= \frac{1}{m}\left(\sum_{i=1}^{m} \log a_i + \sum_{l \in N} \log(1 + b_l)\right) + \left(1 - \frac{1}{m}\right) \log(1 + b_k).
\end{aligned}$$

Since

$$\log C_{\max}(\mathrm{OPT}) \geqslant \frac{1}{m}\left(\sum_{i=1}^{m} \log a_i + \sum_{l \in N} \log(1 + b_l)\right)$$

and

$$\log C_{\max}(\mathrm{OPT}) \geqslant \log(1 + b_k),$$

we have

$$\begin{aligned}
\log C_{\max}(\mathrm{LS}) &\leqslant \log C_{\max}(\mathrm{OPT}) + \left(1 - \frac{1}{m}\right) \log C_{\max}(\mathrm{OPT}) \\
&= \left(2 - \frac{1}{m}\right) \log C_{\max}(\mathrm{OPT}),
\end{aligned}$$

i.e.,

$$\frac{\log C_{\max}(\mathrm{LS})}{\log C_{\max}(\mathrm{OPT})} \leqslant 2 - \frac{1}{m}.$$

We show that the bound $2 - \frac{1}{m}$ is tight. Consider an m-machine and $m^2 - m + 1$-job scheduling problem:

$a_i = 1,\ i = 1, 2, \dots, m,$
$b_j = p - 1,\ j = 1, 2, \dots, m^2 - m,$
$b_j = p^m - 1,\ j = m^2 - m + 1$, where $p > 1$.

The assignment of jobs in the optimal and LS schedules is shown in Table 1. Obviously, $\log C_{\max}(\mathrm{LS}) = (2m - 1)\log p$ and $\log C_{\max}(\mathrm{OPT}) = m \log p$. So $\frac{\log C_{\max}(\mathrm{LS})}{\log C_{\max}(\mathrm{OPT})} = 2 - \frac{1}{m}$. □

For $Pm, a_i | p_j = b_j t | \log C_{\max}$ problem, a better worst-case bound can be obtained by applying the largest deterioration rate (LDR)