Doctopic: Algorithms, automata, complexity and games ARTICLE IN PRESS TCS:10912

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Semi-online scheduling with bounded job sizes on two uniform machines

Qian Cao [a,*], Zhaohui Liu [b]

[a] College of Economics and Management, Shanghai University of Electric Power, Shanghai, China
[b] Department of Mathematics, East China University of Science and Technology, Shanghai, China

ARTICLE INFO

Article history:
Received 3 June 2016
Received in revised form 7 August 2016
Accepted 24 August 2016
Available online xxxx
Communicated by D.-Z. Du

Keywords:
Scheduling
Uniform machines
Semi-online
Bounded job sizes

ABSTRACT

In this paper, we investigate a semi-online scheduling problem on two uniform machines with the speed ratio s. It is assumed that all jobs have their processing times between p and tp ($p > 0$, $t \geq 1$). The objective is to minimize the makespan. We give the competitive ratio of LS algorithm which is a piecewise function on $t \geq 1$ and $s \geq 1$. It shows that LS is an optimal algorithm for most regions on s and t. We further present two optimal algorithms. The algorithm H_1 with competitive ratio of s is optimal for $1.325 \leq s \leq \frac{1+\sqrt{5}}{2}$ and $s < t \leq \frac{s^2-1}{1+s-s^2}$. The algorithm H_2 with competitive ratio of s is optimal for $1.206 \leq s \leq 1.5$ and $s \leq t \leq \min\{2s-1, \frac{2(s^2-1)}{1+s-s^2}\}$, and it is also optimal for $1 \leq s \leq \frac{1+\sqrt{17}}{4}$ and $\max\{2s-1, \frac{-s+\sqrt{9s^2+8s}}{2s}\} \leq t \leq \frac{2}{s}$ with competitive ratio of $\frac{1+t}{2}$.

1. Introduction

In this paper, we consider the following semi-online scheduling problem with bounded job sizes on two uniform parallel machines. To state the problem, we are given a list $L = (J_1, J_2, \cdots, J_n)$ of n jobs that are to be assigned one by one to two uniform parallel machines M_1 and M_2. Each job J_j is associated with a size p_j, but we only know the size of the first unassigned job as well as the fact that all jobs have their sizes between p and tp ($p > 0, t \geq 1$). It is possible that no jobs with sizes p and tp come up. Without loss of generality, we assume $p = 1$. The speed of M_i is s_i ($i = 1, 2$), i.e., the processing time of job J_j is $\frac{p_j}{s_i}$ if it is assigned to M_i. We assume that $1 = s_1 \leq s_2 = s$. The objective is to find a schedule that minimizes the makespan, i.e., the maximum completion time of M_1 and M_2.

Compared with the traditional online scheduling model in which the scheduler only knows the information of the current job, a semi-online model allows the scheduler to have more information, such as the lower and upper bounds on the sizes of unscheduled jobs in our problem. However, the performance of a semi-online algorithm is still measured by its competitive ratio with respect to the optimal offline algorithm. Let $C^{\mathcal{H}}_{\max}$ denote the makespan of the schedule produced by a semi-online algorithm $\mathcal{H}$ for the discussed problem, and $C^*_{\max}$ be the optimal makespan of its offline version. Then, the competitive ratio of algorithm $\mathcal{H}$ is defined as

$$r_{\mathcal{H}} = \inf_r \{r \geq 1 | C^{\mathcal{H}}_{\max} \leq r C^*_{\max}\}.$$

* Corresponding author.
E-mail address: qcao@shiep.edu.cn (Q. Cao).

http://dx.doi.org/10.1016/j.tcs.2016.08.022
0304-3975/© 2016 Elsevier B.V. All rights reserved.

Table 1
Lower bounds.

s	t	Lower bounds
$1 \le s \le \frac{1+\sqrt{5}}{2}$	$t \ge \frac{1+s}{s\alpha^2}$ ($\alpha = \frac{1+s-s^2}{s^2}$)	$\frac{2s+1}{s+1}$
	$s \le t \le \frac{1+s}{s\alpha^2}$	s
$s \ge \frac{N+\sqrt{N^2+4N}}{2}$	$t \ge \frac{s}{N}$	$\frac{s+1}{s}$
	$\max\{\frac{1}{s-N}, \frac{s-1}{N}\} \le t \le \frac{s}{N}$	$\frac{1+Nt}{s}$
$N \le s \le N+1$	$1 \le t \le \min\{\frac{1}{s-N}, \frac{s}{N}\}$	t
–	$\frac{2s(N+1)-2N-1}{2N+1-2sN} \le t \le \frac{2}{s}$	$\frac{1+t}{2}$
–	$\frac{sN-N+s}{N+1-sN} \le t \le \min\{\frac{2N+1}{2Ns}, \frac{2s(N+1)-2N-1}{2N+1-2sN}\}$	$\frac{s(Nt+N+1)}{2N+1}$

We call c a lower bound of the problem if it has no determined semi-online algorithm with the competitive ratio less than c. Accordingly, algorithm $\mathcal{H}$ is called optimal or the best possible if its competitive ratio reaches some lower bound.

For the online counterpart of our problem, there is a simple algorithm that assigns the current job to the machine on which it will have the earliest completion time. This algorithm is called list scheduling (LS, for short) algorithm. Cho and Sahni [1] showed that the competitive ratio of LS algorithm is $\frac{1+\sqrt{5}}{2}$. Epstein et al. [2] obtained the parameterized competitive ratio

$$r_{LS} = \begin{cases} \frac{2s+1}{s+1}, & s \le \frac{1+\sqrt{5}}{2}, \\ \frac{s+1}{s}, & s \ge \frac{1+\sqrt{5}}{2}, \end{cases}$$

and proved that LS is the best possible online algorithm for any $s \ge 1$.

Various semi-online scheduling problems with the makespan objective on two uniform machines have been studied in the literature. If the jobs in L are known to arrive in nonincreasing order of their sizes, Gonzalez et al. [3] proved that LS algorithm (i.e., LPT (Longest Processing Time first) rule for the offline version) has a competitive ratio of $\frac{1+\sqrt{17}}{4}$. Further, Mireault, Orlin and Vohra [4] analyzed the performance of LPT rule as a function of s, and Epstein and Favrholdt [5] designed the optimal algorithm for all s concerning the semi-online version. If the maximum size of the jobs is known in advance, Cao and Liu [6] presented the optimal algorithm for $1 \le s \le \sqrt{2}$, $1.559 \le s \le 2$ and $s \ge \frac{3+\sqrt{17}}{2}$. Epstein [7] and Ng et al. [8] studied the problem with known optimal makespan, and gave the optimal algorithm for $\frac{1+\sqrt{65}}{8} \le s \le \frac{1+\sqrt{21}}{4}$ and $s \ge \sqrt{3}$. Ng et al. [8] also studied the problem with known total size of the jobs in L, and presented the optimal algorithm for $\frac{1+\sqrt{65}}{8} \le s \le 1.391$ and $s \ge \sqrt{3}$. Epstein and Ye [9] investigated the problem in which the last job in L is marked and has the maximum size, and presented the optimal algorithm for $1 \le s \le 1.465$ and $s \ge 1+\sqrt{3}$. Du [10] investigated the preemptive version of our problem, and characterized the optimal competitive ratio as a function of both s and t.

Except Du [10]'s result, the semi-online scheduling problem with bounded job sizes was discussed mainly in the identical parallel machine environment, where all machines have the same speed. Achugbue and Chin [11] first analyzed the performance of LS algorithm for jobs with bounded sizes on identical parallel machines. For two identical parallel machines, He and Zhang [12] showed that LS algorithm is optimal and has the competitive ratio $\min\{\frac{1+t}{2}, \frac{3}{2}\}$. For $m \ge 3$ identical parallel machines, He [13] showed that LS algorithm with the competitive ratio $1 + \frac{(m-1)(t-1)}{m}$ is optimal for $1 \le t \le \frac{m}{m-1}$. For three identical parallel machines, He and Dosa [14] proved that LS algorithm is optimal for the cases with $1 \le t \le \frac{3}{2}$, or $\sqrt{3} \le t \le 2$ or $t \ge 6$, and proposed a different optimal algorithm for $2 \le t \le \frac{5}{2}$.

In this paper, we first prove the lower bounds for the semi-online scheduling problem $Q2|1 \le p_j \le t|C_{\max}$ in Section 2. Let $N \in \{1, 2, 3, \cdots\}$ and $\alpha = \frac{1+s-s^2}{s^2}$, the lower bounds given in this paper are shown in Table 1.

In Section 3, we investigate the LS algorithm. First, we obtain a competitive ratio of $\min\{\frac{2s+1}{s+1}, \frac{s+1}{s}, t\}$, then we get that LS algorithm is optimal for the cases with $1 \le s \le \frac{1+\sqrt{5}}{2}$ and $t \ge \frac{1+s}{s\alpha^2}$, or $s \ge \frac{N+\sqrt{N^2+4N}}{2}$ and $t \ge \frac{s}{N}$, or $N \le s \le N+1$ and $1 \le t \le \min\{\frac{1}{s-N}, \frac{s}{N}\}$. Also, we prove that $C_{\max}^{LS} \le \max\{\frac{1+t}{2}, \frac{s(Nt+N+1)}{2N+1}\}C_{\max}^*$ for $t \ge 1$, then we get that LS algorithm is optimal for the cases with $\frac{2s(N+1)-2N-1}{2N+1-2sN} \le t \le \frac{2}{s}$, or

$$\frac{sN-N+s}{N+1-sN} \le t \le \min\{\frac{2N+1}{2Ns}, \frac{2s(N+1)-2N-1}{2N+1-2sN}\}.$$

Furthermore, we prove that $C_{\max}^{LS} \le \frac{1+Nt}{s}C_{\max}^*$ for $s \le N+1$ and $t \ge \frac{3(s^2-1)}{3(s+1)N-s}$, and we get that LS algorithm is optimal for $\frac{N+\sqrt{N^2+4N}}{2} \le s \le N+1$ and $\max\{\frac{3(s^2-1)}{3(s+1)N-s}, \frac{1}{s-N}\} \le t \le \frac{s}{N}$.

Please cite this article in press as: Q. Cao, Z. Liu, Semi-online scheduling with bounded job sizes on two uniform machines, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.08.022