Theoretical Computer Science 702 (2017) 65–76

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Total completion time minimization scheduling on two hierarchical uniform machines

Hao Zhou [a], Yiwei Jiang [b,*], Ping Zhou [c], Min Ji [b], Yun Zhao [d]

[a] *Department of Basic Science, Zhejiang Shuren University, Hangzhou 310015, China*
[b] *School of Management and E-Business, Contemporary Business and Trade Research Center, Zhejiang Gongshang University, Hangzhou 310018, China*
[c] *School of Humanities, Zhejiang Business College, Hangzhou 310053, China*
[d] *Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou 310018, China*

ARTICLE INFO

Article history:
Received 22 January 2017
Received in revised form 9 June 2017
Accepted 21 August 2017
Available online 30 August 2017
Communicated by T. Erlebach

Keywords:
Online scheduling
Hierarchy
Total completion time
Competitive ratio

ABSTRACT

This paper investigates an online hierarchical scheduling problem on two uniform machines to minimize the total completion time of all jobs. Machine M_1 with a speed v_1 has a lower hierarchy and machine M_2 with a speed v_2 has a higher hierarchy. Each job has a unit processing time and a hierarchy. The job with a lower hierarchy can only be processed on the machine M_1 and the job with a higher hierarchy can be processed on any of the two machines. We consider two variants of the problem: $v_1 = 1 \le v_2 = s$ and $v_1 = s \ge v_2 = 1$. For both variants, we provide parameter lower bounds and online algorithms with competitive ratio of $\frac{9-\sqrt{41}}{2}$ and $\frac{\sqrt{33}-3}{2}$ respectively, which are optimal in sense of constant bound.

1. Introduction

In this paper, we study an online hierarchical scheduling problem on two uniform machines M_1 and M_2 with speeds of v_1 and v_2, respectively. Jobs arrive one by one in a sequence of $J_1, J_2, \ldots, J_n$ and each job has to be assigned to a machine before the next job arrives. The job J_j has a unit processing time $p_j = 1$ and a hierarchy g_j associated with it, and a machine M_i is also associated with a hierarchy $g(M_i)$. The job J_j can be processed on the machine M_i only if $g_j \ge g(M_i)$. Without loss of generality, we assume that $g(M_1) = 1$ and $g(M_2) = 2$. Each job J_j, $j = 1, \cdots, n$, has a hierarchy $g_j = 1$ or alternatively $g_j = 2$. Namely, the job with $g_j = 1$ can only be processed on the machine M_1 and the job with $g_j = 2$ can be processed on any one of two machines. Our goal is to minimize the total completion time of all jobs, i.e., $z = \sum_{j=1}^{n} C_j$, where C_j is the completion time of job J_j. We consider two variants according to the speeds of two uniform machines. The first variant, denoted by $\mathcal{P}_1$, considers the case where $v_1 \le v_2$. W. L. O. G., we can assume that $v_1 = 1$ and $v_2 = s(s \ge 1)$. The second variant, denoted by $\mathcal{P}_2$, considers the case where $v_1 = s(s \ge 1)$ and $v_2 = 1$. Using the three-field notation for describing scheduling problem, we denote the two variants of the problem as $Q2(1,s)|p_j = 1, g_i|\sum C_j$ and $Q2(s,1)|p_j = 1, g_i|\sum C_j$, respectively.

* Corresponding author.
E-mail address: ywjiang@zstu.edu.cn (Y. Jiang).

http://dx.doi.org/10.1016/j.tcs.2017.08.016
0304-3975/

For the online scheduling problem, the performance of an online algorithm A is usually measured by its *competitive ratio*, which can be defined as follows.

$$R_A = \inf\{R| \frac{C^A(\mathcal{J})}{C^*(\mathcal{J})} \leq R, \ \forall \mathcal{J}\},$$

where $C^A(\mathcal{J})$ (or in short C^A) denotes the objective value produced by A and $C^*(\mathcal{J})$ (or in short C^*) denotes the optimal objective value in the offline version of the problem for a job instance $\mathcal{J}$. In the online scheduling, it is often possible to show that there exists a *lower bound* to the competitive ratio achievable by any (deterministic) online algorithm. In that case, an online algorithm is called *optimal* if its competitive ratio matches the lower bound.

Research on the online hierarchical scheduling problem abounds. However, most of papers discuss the objective of minimizing the makespan. In Bar-Noy et al. [1] and Crescenzi et al. [3], they consider the non-preemptive version on m identical machines with general hierarchy settings. For the problem on two identical machines, Jiang et al. [8] and Park et al. [10] independently provide an optimal online algorithm. For the general m-machines case with two hierarchies, Jiang [7] show that the lower bound of the problem is at least 2 and present an online algorithm with competitive ratio $\frac{12+4\sqrt{2}}{7} \approx 2.522$. Hereafter, Zhang et al. [15] improve the result to 7/3 and present an optimal algorithm for the special case where all jobs have a unit processing time. In Shabtay and Khari [9,11], they also consider a kind of online scheduling problem with two hierarchies and some special cases. Tan and Zhang [14] consider the problem with multiple hierarchies using linear programming method. In addition, Jiang et al. [8] study the preemptive version in which idle time is not allowed and they presented an optimal algorithm with competitive ratio of 3/2 on two identical machines. In Chassid and Epstein [2], Dósa and Epstein [4], Tan and Zhang [13], Hou and Kang [5,6], the authors study some models on two uniform machines.

Unlike the above online hierarchical scheduling problem dealing with the objective of minimizing makespan, Shabtay and Khari [12] consider the objective of minimizing total completion time for the problem on two machines with unit processing times. They show that the lower bound of this problem is at least 1.1573, and provide an asymptotically optimal online algorithm with competitive ratio of $1.1573 + O(\frac{1}{n})$. Recently, Hu et al. [16] improve the lower bound of 1.1573 to $16/13 \approx 1.2308$ and proposed an optimal online algorithm with competitive ratio of 16/13. Furthermore, they extend this problem to m-machines case where one machine is of hierarchy 1 and the rest $m-1$ machines are of hierarchy 2. They show that the lower bound is at least

$$1+\min\{\frac{1}{m}, \max\{\frac{2}{\lceil\sqrt{2m+4}\rceil + \frac{2m+4}{\lceil\sqrt{2m+4}\rceil} + 3}, \frac{2}{\lfloor\sqrt{2m+4}\rfloor + \frac{2m+4}{\lfloor\sqrt{2m+4}\rfloor} + 3}\}\}.$$

They further propose a greedy algorithm with competitive ratio of $1 + \frac{2(m-1)}{m(\sqrt{4m-3}+1)}$.

In this paper, we discuss the objective of minimizing total completion time for the problem on two uniform machines. For both variants $\mathcal{P}_1$ and $\mathcal{P}_2$, we give the lower bounds and present optimal algorithms with competitive ratios of $\frac{9-\sqrt{41}}{2} \approx 1.298$ and $\frac{\sqrt{33}-3}{2} \approx 1.372$ for all $s \geq 1$, respectively.

The rest of the paper is organized as follows: In Section 2 we provide the preliminaries. In Sections 3 and 4 we discuss the two variants $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Finally, we conclude the paper and suggest future research topics in Section 5.

2. Preliminaries

In this section, we mainly provide the optimal schedules in the offline versions and the lower bounds of two variants $\mathcal{P}_1$ and $\mathcal{P}_2$.

We first discuss the objective value by introducing a function $f(x) = \frac{x(x+1)}{2}$. Suppose that there are x jobs with unit processing times processed on the machine with speed v, we can conclude that the total completion time of all these x jobs is $\frac{f(x)}{v} = \frac{x(x+1)}{2v}$. Thus, let x_i be the number of jobs assigned to machine M_i in a schedule, $i = 1, 2$, the objective value of the schedule for $\mathcal{P}_1$ and $\mathcal{P}_2$ are

$$f(x_1) + \frac{f(x_2)}{s}$$

and

$$\frac{f(x_1)}{s} + f(x_2),$$

respectively.

We next focus on the optimal schedules in two variants $\mathcal{P}_1$ and $\mathcal{P}_2$. Let n_i be the number of jobs with hierarchy i, $i = 1, 2$, thus we have $n_1 + n_2 = n$. Let x_i^* be the number of jobs assigned to machine M_i in an optimal schedule σ^*, $i = 1, 2$. The following two lemmas give the optimal schedules of the variants $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively.