Theoretical Computer Science 412 (2011) 5225–5231

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Optimal algorithms for online scheduling on parallel machines to minimize the makespan with a periodic availability constraint

Ming Liu [a,c,*], Feifeng Zheng [c], Chengbin Chu [a,b], Yinfeng Xu [c]

[a] School of Economics & Management, Tongji University, Shanghai, 200092, PR China
[b] Laboratoire Génie Industriel, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Châtenay-Malabry Cedex, France
[c] School of Management, Xi'an Jiaotong University, Xi'an, Shaanxi Province, 710049, PR China

ARTICLE INFO

Article history:
Received 7 June 2010
Received in revised form 12 May 2011
Accepted 16 May 2011
Communicated by D.-Z. Du

Keywords:
Online scheduling
Makespan
Competitive analysis
Parallel machines
Periodic availability constraint

ABSTRACT

In this paper we investigate two online scheduling problems. The first one is online scheduling on m parallel machines with one machine periodically unavailable. The second problem is online scheduling on two uniform parallel machines where one machine is periodically unavailable. The online paradigm is that jobs arrive over list, i.e., when a job presents, we have to irrevocably assign it before the next one is seen. Preemption is not allowed. The objective is to minimize makespan. We suppose that the length of each available period is normalized to 1 and the length of each unavailable period is $\alpha > 0$. For the first problem, we give an optimal algorithm with competitive ratio 2. For the second problem, we assume that the speed of the periodically unavailable machine is normalized to 1, while the speed of the other one is $s > 0$. In the case where $s \geq 1$, we design an algorithm and show that it is optimal with competitive ratio $1 + \frac{1}{s}$. Then we further give some lower bounds on competitive ratio in the case $0 < s < 1$. We also study a special case and prove that *LPT* algorithm proposed in Xu et al. (2009) [7] is optimal with competitive ratio $\frac{3}{2}$.

1. Introduction

Online scheduling becomes more and more concerned because in practice the useful information of the problem instance was not known in advance. In the literature of online scheduling, two online models have been widely researched [2]. The first one assumes that there are no release times and that the jobs arrive over list (or one by one). An online algorithm has to irrevocably schedule the first job in this list before it sees the next job in the list. The other model assumes that jobs arrive over time. Each job is associated with a release time, before which this job cannot be scheduled. At each time when the machine is idle, the algorithm decides which one of the available jobs is scheduled, if any. In this paper, we consider the first online paradigm.

Online algorithm is developed to cope with online (scheduling) problems. For a certain online scheduling problem, we would like to find an optimal (or called best possible) algorithm. This algorithm has the best possible performance. In order to compare the performance of online algorithms, we need a tool to measure the performance of each algorithm. In the literature, competitive analysis [1] is such a tool to serve this purpose. Specifically, for any input job sequence I, let $C_{ON}(I)$ denote the makespan of the schedule produced by the online algorithm $\mathcal{A}_{ON}$ and $C_{OPT}(I)$ denote the makespan of an optimal schedule. We say that $\mathcal{A}_{ON}$ is ρ-competitive if

$$C_{ON}(I) \leq \rho C_{OPT}(I).$$

We also say that ρ is the competitive ratio of $\mathcal{A}_{ON}$.

* Corresponding author at: School of Economics & Management, Tongji University, Shanghai, 200092, PR China. Tel.: +86 013659234252.
E-mail address: minyivg@gmail.com (M. Liu).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2011.05.028

For an online minimization problem, a lower bound means that there exists no online algorithm with a competitive ratio smaller than this bound. If an online algorithm's competitive ratio achieves this lower bound, this algorithm is called optimal and the corresponding lower bound is called tight. For an online problem, a general idea is to first give a lower bound then prove the competitive ratio of an online algorithm. Then we can justify whether or not the proposed algorithm is optimal.

In the classical scheduling problem, it is assumed that machines are available simultaneously at all times. However, this availability assumption may not be true in reality [3]. In the real industry settings, machines may be unavailable because of preventive maintenances, periodical repairs and tool changes.

This paper studies online scheduling with a periodic availability constraint. We are given a sequence of independent jobs which arrive over list. When all information is available at one time before scheduling, the problem is called *offline*. In the offline settings, Lee [5] investigated some parallel machine scheduling problems where at least one machine is always available and each of the other machines has at most one unavailable period. He gave some results of competitive ratios for different objectives, such as minimization of total completion time and makespan. Liao et al. [4] considered a special case of one of the scheduling problems studied in [5]. They partitioned the problem into four subproblems, each of which was optimally solved. In the online settings, Tan and He [6] considered the online scheduling on two identical machines with machine availability constraint to minimize makespan. They assumed that the unavailable periods of two machines do not overlap and proposed an optimal online algorithm with a competitive ratio $\frac{5}{2}$. Xu et al. [7] showed that for the problem of online scheduling on two identical machines where one machine is periodically unavailable with the objective of minimizing makespan, the competitive ratio of *LS* algorithm is 2.

The remainder of this paper is organized as follows. In Section 2, we deal with the problem of online scheduling on m parallel machines with one machine periodically unavailable. In this part, we first give a lower bound on competitive ratio and then present an optimal algorithm. In Section 3, we study the problem of online scheduling on two uniform machines. The speeds of the periodically unavailable machine and the other one are 1 and $s > 0$, respectively. In the case $s \geq 1$, we show a lower bound and design an optimal algorithm. In the case $0 < s < 1$, we prove several lower bounds in different situations. Then in Section 4, we investigate a special case of two parallel machine scheduling problem, i.e., jobs arrive in a non-increasing order of their processing times. We prove that *LPT* is optimal.

2. *m* parallel machine scheduling problem

We are given m parallel machines where one of them is periodically unavailable. Jobs arrive over list, i.e., a sequence of jobs $\sigma = \{J_1, J_2, \dots, J_n\}$ which arrive online have to be scheduled irrevocably on one of the machines at the time of their arrivals. The new job shows up only after the current job is scheduled. At the beginning, the periodically unavailable machine starts with an available period. Preemption is not allowed. The objective is to minimize the makespan. We use $M1$ to denote the machine which is periodically unavailable and $M2, \dots, Mm$ to denote the others and the speed of each machine is 1. Without loss of generality, on machine $M1$, we assume the length of available period and unavailable period are 1 and $\alpha > 0$, respectively. (Since jobs' processing times can be scaled according to the length of available or unavailable period.) We use p_j to denote the processing time of job J_j. Our first problem can be written by $Pm, M1PU|online|C_{max}$, where $M1PU$ means one machine is periodically unavailable.

Let C_{ON} and C_{OPT} denote the makespan of online algorithm and offline optimal algorithm (for short, offline algorithm), respectively.

2.1. Lower bound on competitive ratio

In this subsection, we present a lower bound on competitive ratio for this problem.

Theorem 1. *For problem $Pm, M1PU|online|C_{max}$, there exists no online algorithm with competitive ratio less than* 2.

Proof. Let ϵ be a sufficiently small positive number. We first give a set of m jobs $J_1, \dots, J_m$ with a common processing time ϵ.

Case 1. One machine processes at lease two jobs.

In this case, $C_{ON} \geq 2\epsilon$. In an optimal schedule, each machine processes exactly one job, i.e., $C_{OPT} = \epsilon$. Therefore, $\frac{C_{ON}}{C_{OPT}} \geq 2$.

Case 2. Each machine processes one job.

We further give a second set of m jobs $J_{m+1}, \dots, J_{2m}$ with a common processing time 1. Therefore, $C_{ON} \geq \min\{\epsilon+2, 2+\alpha\}$. In an optimal schedule, each machine only processes one job of the second set and one of machines $M2, \dots, Mm$ schedules two jobs of the first set. Therefore, $C_{OPT} = 1 + 2\epsilon$. It follows that

$$\frac{C_{ON}}{C_{OPT}} \geq \frac{\min\{\epsilon + 2, 2 + \alpha\}}{1 + 2\epsilon} \to 2, \quad \epsilon \to 0.$$

This completes the proof. □

Corollary 1. *For problem $P2, M1PU|online|C_{max}$, LS algorithm proposed in [7] is optimal with competitive ratio 2.*