Computers & Industrial Engineering 60 (2011) 542–549

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Online deadline scheduling with preemption penalties ☆

Feifeng Zheng [a,b], Yinfeng Xu [a,b], Chung Keung Poon [c,*], E. Zhang [d], Xiaoping Wu [e]

[a] School of Management, Xi'an JiaoTong University, Xi'an 710049, China
[b] The State Key Lab for Manufacturing Systems Engineering, Xi'an 710049, China
[c] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China
[d] Shanghai University of Finance and Economics, Shanghai 200433, China
[e] Xi'an University of Posts and Telecommunications, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 3 May 2010
Received in revised form 9 December 2010
Accepted 17 December 2010
Available online 25 December 2010

Keywords:
Deadline scheduling
Online scheduling algorithms
Competitive analysis
Preemption penalty

ABSTRACT

This paper presents a study of the problem of online deadline scheduling under the preemption penalty model of Zheng, Xu, and Zhang (2007). In that model, each preemption incurs a penalty of ρ times the weight of the preempted job, where $\rho \geqslant 0$ is the preemption penalty parameter. The objective is to maximise the total weight of jobs completed on time minus the total penalty.

When the scheduler knows the ratio of longest to shortest job length, Δ, we show that the WAL algorithm of Zheng et al. (2007) is $((1+\rho)\Delta + o(\Delta))$-competitive for sufficiently large Δ. This improves the bound shown in Zheng et al. (2007). When the scheduler only knows that $\Delta \geqslant (k(1+\rho))^3$ for some $k > 1$, we propose a $((k(1+\rho)\Delta/(k-1)) + o(\Delta))$-competitive algorithm.

When $\rho = 0$, we give an optimal, $O(\Delta/\log \Delta)$-competitive algorithm that, unlike previous algorithms, does not require knowledge of Δ. This settles an open problem mentioned in Ting (2008).

1. Introduction

Online deadline scheduling has recently received much attention due to its many applications in the manufacturing industry, real time systems and networks; see for example (Baruah et al., 1991; Baruah et al., 1992; Hoogeveen, Potts, & Woeginger, 2000; Goldman, Parwatikar, & Suri, 2000; Lee, 2003; Pruhs, Torng, & Sgall, 2004, chap. 15). In a typical scenario in manufacturing, jobs of various weights arrive over time, and the scheduler has to arrange the processing so that they can be finished before their respective deadlines. When a job is completed before its deadline, an amount of profit proportional to the weight of the job is obtained. When the system is overloaded, there are more jobs than the machine can handle. In this case, the challenge is to decide which jobs to process and which to give up to maximise the total profit.

In this paper, we consider preemptive scheduling on a single machine and follow the standard *preemption-restart model* (see Shmoys, Wein, & Williamson, 1995). That is, preemption is allowed, but the aborted job has to be started again from the beginning to obtain its profit.

1.1. Preemption penalty

In some applications, starting a job represents a commitment to serve the corresponding client. Aborting the job will then likely cause a certain degree of discontent in the affected client. Therefore, we are motivated to study scheduling when there is a penalty for preemption. Zheng et al. (2007) were the first to investigate this problem and introduced the following model of preemption penalties. The system consists of a single machine to process jobs that arrive online. Each job J has four attributes, $a(J)$, $p(J)$, $w(J)$ and $d(J)$, representing its arrival time, processing time (i.e., job length), profit and deadline respectively. The scheduler does not know any of these job parameters until the job arrives, i.e., at time $a(J)$. When it arrives, all its attributes become known. The scheduler gains a profit of $w(J)$ if it completes job J by its deadline $d(J)$. On the other hand, there will be a penalty of $\rho\, w(J)$ if he starts J but aborts it before its completion. Here ρ is called the *preemption penalty parameter* and is a non-negative real number. (When $\rho = 0$, the model reduces to the classic one in which there is no penalty.) Note that if a job is preempted multiple times, a penalty will be imposed for each preemption. Our goal is to maximise the total weight of the completed jobs minus the total penalties caused by preemptions.

To measure the performance of an online algorithm $\mathscr{A}$, competitive ratio analysis (refer to Borodin & El-yaniv (1998)) is often used. Denote the schedules produced by $\mathscr{A}$ on an input I as $\Gamma_{\mathscr{A}}(I)$, and the schedule produced by an optimal offline algorithm OPT as $\Gamma^*(I)$. Let $|\Gamma_{\mathscr{A}}(I)|$ and $|\Gamma^*(I)|$ be the total profit of completed jobs in $\Gamma_{\mathscr{A}}(I)$ and $\Gamma^*(I)$ respectively. Denote the total preemption

☆ This manuscript was handled by area editor Maged M. Dessouky.

* Corresponding author. Tel.: +852 2788 7157; fax: +852 2788 8614.
E-mail addresses: zhengff@mail.xjtu.edu.cn (F. Zheng), yfxu@mail.xjtu.edu.cn (Y. Xu), ckpoon@cs.cityu.edu.hk (C.K. Poon), eezhang@163.com (E. Zhang), wuxiaoping@stu.xjtu.edu.cn (X. Wu).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.12.011

penalty received by $\mathscr{A}$ in $\Gamma_{\mathscr{A}}(I)$ as $|P_{\Gamma_{\mathscr{A}}}(I)|$. OPT is an offline optimal algorithm and never aborts jobs, implying that there will be no preemption penalties. Therefore, the competitive ratio of $\mathscr{A}$ is defined as $r_{\mathscr{A}} = \sup_I \frac{|\Gamma^*(I)|}{|\Gamma_{\mathscr{A}}(I)|-|P_{\Gamma_{\mathscr{A}}}(I)|}$. If $\rho = 0$, then $r_{\mathscr{A}} = \sup_I \frac{|\Gamma^*(I)|}{|\Gamma_{\mathscr{A}}(I)|}$.

1.2. Previous results

Zheng et al. (2007) presented the WAL algorithm that makes use of knowledge of Δ (the ratio of the length between the longest and shortest jobs) and proved their algorithm to be $(3\Delta + o(\Delta))$-competitive for $\Delta > 9$ and $\rho = 1$. They also gave a $(1.366\Delta + 0.366)$ lower bound.

Fung (2008) considered the general case for $\rho > 0$, and proved a lower bound of $((1 + 1/\rho)^{1/\lceil\Delta\rceil} - 1)^{-1} + 1$. When $\rho = 1$, the lower bound is approximately $\Delta/\ln 2 \approx 1.443\Delta$ for large Δ, improving the previous bound of $(1.366\Delta + 0.366)$. Fung (2008) also pointed out that in fact WAL has a competitive ratio of $(2 + \rho)\Delta + o(\Delta)$ for constant ρ and sufficiently large Δ. When $\rho = 1$, the ratio is the same as that proved in Zheng et al. (2007).

Some previous works for the online broadcast scheduling problem are also relevant to the special case of $\rho = 0$ in our problem. Specifically, when translated to our terminology, Fung, Chin, and Poon (2005) proposed the Another Completes Earlier (ACE) algorithm and proved that it is $(\Delta + 2\sqrt{\Delta} + 2)$-competitive. This was improved by Ting (2008) to $O(\Delta/\log \Delta)$, matching the lower bound of $\Omega(\Delta/\log \Delta)$ by Zheng et al. (2006). Note that both Fung et al.'s and Ting's algorithms require knowledge of Δ. Whether an equally competitive algorithm exists that does not assume knowledge of Δ was left as an open problem in Ting (2008). The best previous algorithm without knowledge of Δ is $(4\Delta + 1)$-competitive, obtained by a simple extension of the greedy algorithm, GD, proposed by Kim and Chwa (2003).

1.3. Our contributions

Throughout this paper, we assume that $1 \leqslant p(J) \leqslant \Delta$. For the problem when $\rho > 0$, we give a tighter analysis of WAL and show that it is $((1 + \rho)\Delta + o(\Delta))$-competitive for a large enough Δ. This implies that WAL is $(2\Delta + o(\Delta))$-competitive when $\rho = 1$, improving the previous bound of $(3\Delta + o(\Delta))$.

Note that WAL knows the exact value of Δ beforehand. We also present a modification of WAL so that it only needs to know a lower bound on Δ, i.e., $\Delta \geqslant k^3(1 + \rho)^3$ for some real number $k > 1$. We prove that the modified algorithm is $\left(\frac{k(1+\rho)}{k-1}\Delta + o(\Delta)\right)$-competitive for large enough Δ. As k increases, the ratio approaches $((1 + \rho)\Delta + o(\Delta))$ from above. Note that this result is useful in situations, where the manufacturer may foresee the information of some future jobs at the beginning via certain business techniques so that he has a lower bound on Δ.

For the problem when $\rho = 0$, we present an optimal ($O(\Delta/\log \Delta)$-competitive) algorithm, which does not require knowledge of Δ at all. This answers an open question in Ting (2008).

1.4. Related work

One can view the non-preemptive problem as a special case of our problem, where $\rho = \infty$. Lipton and Tomkins (1994) studied the scenario when the job deadlines are always tight (interval scheduling) and the job lengths are chosen from a finite set of real numbers instead of an arbitrary number within $[1, \Delta]$. One of their results is a two-competitive non-preemptive algorithm in the case, where $p(J)$ is either 1 or Δ and the profit is proportional to the job length (i.e., $w(J) = p(J)$). Goldwasser (2003) extended the work of Lipton and Tomkins (1994) and investigated the case, where each job has a slack time equal to $k \geqslant 0$ times of its length, i.e., $d(J) - a(J) = (k + 1)p(J)$. They proved a matching upper and lower bound of $\left(2 + \frac{\lceil\Delta\rceil - 1}{\Delta}\right)$ when $\frac{1}{\Delta} \leqslant k < 1$, and a matching bound of $\left(1 + \frac{\lceil\Delta\rceil}{\Delta}\right)$ when $1 \leqslant k < \Delta$.

Preemption penalty has also been studied with other objectives, such as minimising flow time or total completion time. There, the penalty is either modelled as set-up costs or as a requirement to redo some portion of the preempted jobs (see e.g., Julian, Magazine, & Hall (1997), Liu & Cheng (2002), Liu & Cheng (2004) and the references therein).

1.5. Organisation of Paper

The rest of the paper is organised as follows. In Section 2, we give our analysis of WAL and present a modified algorithm that only assumes knowledge of a lower bound on Δ. In Section 3, we give an algorithm for the case, where $\rho = 0$ and does not require any knowledge of Δ. Finally, Section 4 concludes the paper.

2. Scheduling with preemption penalties

2.1. The case with knowledge of Δ

In this section, we will give a tighter analysis of the algorithm WAL (Weight-and-Length) proposed in Zheng et al. (2007). We first state the WAL Algorithm below.

The WAL Algorithm. The algorithm is triggered when either a job is completed or a new job arrives. When WAL completes a job, it will start to process the job with the largest profit among those that have arrived but not yet been satisfied. If a job R arrives while WAL is processing J, WAL will abort J to start R if and only if one of the following two conditions is satisfied:

C1: $w(R) \geqslant \beta w(J)$,
C2: $\alpha w(J) \leqslant w(R) < \beta w(J)$ and $p(R) < p(J)/\sqrt{\Delta}$,

where α is some constant (to be determined later) such that $1 < \alpha < \beta$ and $\beta = \Delta^{1/3}$. (Note that we could have chosen $\beta = \Delta^{\gamma}$ for any positive $\gamma < 1/2$. We fix β at $\Delta^{1/3}$ to avoid introducing more symbols in our analysis.)

To analyse the competitive ratio of WAL, we define the notion of a *preempting chain* (called a *subschedule* in Zheng et al., 2007) as follows. A *preempting chain* in the schedule produced by WAL is a sequence of jobs $\sigma = (J_1, \ldots, J_m)$ such that J_1 is preceded by an idle period or a completed job, J_i is preempted by J_{i+1} for all $1 \leqslant i \leqslant m - 1$ and J_m is a completed job.

Because OPT is an optimal offline algorithm, we can assume without loss of generality that it never aborts a job that it starts. By construction of WAL, every job scheduled by OPT must start between the start and completion of some preempting chain unless it has already been finished by WAL earlier. (Otherwise, if there is some job J started by OPT at time t outside any preempting chain and J has not been finished by WAL earlier, then J is available at time t while WAL is idle, a contradiction.) Thus, to prove an upper bound on the competitive ratio, it suffices to compute the maximum ratio, r, of the profit of OPT to that of WAL on an arbitrary preempting chain. Then the competitive ratio is at most $r + 1$.

The main idea of Zheng et al. (2007) is to continually modify a preempting chain until it possesses certain desirable properties. Moreover, the ratio of the optimal profit to that obtained by WAL can only increase by the sequence of changes. More precisely, suppose J_{i+1} preempts J_i by condition C2 in σ. Then we change the weight and processing time of J_i and J_{i+1} so that J_i preempts J_{i-1} by condition C2 and J_{i+1} preempts J_i by condition C1. We achieve this by decreasing $w(J_i)$ to $w(J_{i+1})/\beta$ and swapping $p(J_i)$ and $p(J_{i+1})$. We repeat this change until no more such change is possible. Using this approach, Zheng et al. (2007) proved that the competitive ratio is at most $3\Delta + o(\Delta)$.