COMPETITIVE DEADLINE SCHEDULING VIA ADDITIONAL OR FASTER PROCESSORS*

CHIU-YUEN KOO, TAK-WAH LAM†, TSUEN-WAN NGAN, AND KAR-KEUNG TO
Department of Computer Science, University of Hong Kong, Hong Kong

ABSTRACT

This paper studies on-line scheduling in a single-processor system that allows preemption. The aim is to maximize the total value of jobs completed by their deadlines. It is known that if the on-line scheduler is given a processor faster (say, two times faster) than the off-line scheduler, then there exists an on-line algorithm called Sacker that can achieve an O(1) competitive ratio. In this paper, we show that using additional unit-speed processors instead of a faster processor is a possible but not cost effective way to achieve an O(1) competitive ratio. Specifically, we find that Θ(log k) unit-speed processors are required, where k is the importance ratio. Another contribution of this paper is an improved analysis of the competitiveness of Sacker; this new analysis enables us to show that Sacker, when extended to multi-processor systems, can still guarantee an O(1) competitive ratio.

KEY WORDS: on-line algorithms; extra-resource analysis; competitiveness; deadline scheduling

1. INTRODUCTION

This paper studies the following on-line deadline scheduling problem in a single-processor system. Jobs are released in an unpredictable fashion. The processing time and deadline of a job are known only when the job is released. Preemption is allowed at no cost. In general, a system may be overloaded, that is, there is no schedule meeting the deadline of every job released. A scheduler aims to maximize the total value of jobs that can be completed by their deadlines, where the value of a job is another parameter given upon the release of the job and reflects the importance of the job. We define the value density of a job to be its value divided by the processing time. The importance ratio of a system is the ratio of the largest possible value density to the smallest possible density. See, Stankovic et al. (1998) for more discussion on deadline scheduling.

We analyze the performance of on-line algorithms with respect to their competitive ratios (see, e.g., Borodin and El-Yaniv (1998) and Sgall (1998) for general background). In this paper, we say that an on-line algorithm A is c-competitive if, for any job sequence, A guarantees to obtain at least a factor 1/c of the total value obtained by any off-line algorithm. For the firm-deadline scheduling problem, the early work of Dertouzos (1974) showed that the Earliest

---

† Correspondence to: Tak-Wah Lam, Department of Computer Science, University of Hong Kong, Hong Kong. E-mail: twlam@CIS.hku.hk

Contract/grant sponsor: Hong Kong RGC; Contract/grant number: HKU-7024/01E
Deadline First (EDF) strategy is 1-competitive for underloaded systems. That is, whenever there exists a schedule in which every job meets the deadline, EDF will give such a schedule. However, without the underloaded assumption, no algorithm can be 1-competitive; indeed, Baruah et al. (1991, 1992) gave a lower bound of $(1 + \sqrt{k})^3$ on the competitive ratio, where $k$ is the importance ratio. This means that even if all jobs have uniform job density (i.e., $k=1$), the best algorithm that we can expect is 4-competitive. Koren and Shasha (1995) showed that this lower bound is tight by giving a $(1 + \sqrt{k})^3$-competitive algorithm. Note that when $k$ is large, such a performance guarantee is not satisfactory.

A plausible approach to obtaining a better performance guarantee without making assumptions about future inputs is to allow the on-line scheduler to have more resources than the off-line algorithm (e.g., Phillips et al., 1997; Berman and Coulston, 1998; Edmonds, 1999; Lam and To, 1999; Brehob, Tornq, and Uthaisombut, 2000; Kalyanasundaram and Pruhs, 2000). Specifically, we compare the on-line scheduler using a faster processor or more than one unit-speed processor against the optimal off-line scheduler using a unit-speed processor. Intuitively, additional resources are needed to compensate for the on-line scheduler’s lack of future information. The key question is whether a moderate amount of additional resources can provide small competitive ratios. For the firm-deadline scheduling problem, the pioneering work of Kalyanasundaram and Pruhs (2000) showed that the competitive ratio can be improved to a constant independent of the importance ratio $k$ when the on-line scheduler is given a moderately faster processor; precisely, they gave an algorithm called Slacker which, if given a $(1 + 2\delta)$-speed processor for any $\delta > 0$, is $(1 + \delta^{-1})(1 + \delta^{1/2})(1 + \delta^{1/2} + \delta^{-1})$-competitive. (A $s$-speed processor is $s$ times faster than a unit-speed processor.) For example, with a 2-speed processor, Slacker is 32-competitive. Furthermore, Lam and To (2001) showed that EDF (supplemented with admission control) can be 1-competitive if a $4\lceil \log k \rceil$-speed processor is used.

An alternative to increasing processor speed is by using extra unit-speed processors. To ease our discussion, we say that an algorithm $A$ is $s$-speed (resp. $p$-processor) $c$-competitive if, for any job sequence, $A$ using a $s$-speed processor (resp. $p$ unit-speed processors) guarantees to obtain at least a factor $1/c$ of the total value obtained by any off-line algorithm using a unit-speed processor. A schedule using $p > 1$ unit-speed processors implies one using a speed-$p$ processor; however, the converse is not necessarily true as jobs may be sequential in nature. For the firm-deadline scheduling problem stated above, it is not known how to make use of additional unit-speed processors to improve the competitive ratio from $(1 + \sqrt{k})^2$ to $O(1)$. Nevertheless, there are two related results when value densities are not a concern. First, Baruah (1998) considered jobs with uniform value density (i.e., $k=1$) and gave a $p$-processor $pl(p-1)$-competitive algorithm. (Without extra resource, the algorithm in Koren and Shasha (1995) is 4-competitive for jobs with uniform value density.) Second, if every job has the same value, or equivalently, the concern is to maximize the number of job completions, Kalyanasundaram and Pruhs (1998) gave an algorithm that is 2-processor $O(1)$-competitive.

In this paper, we revisit the problem of maximizing the total value for jobs with nonuniform value densities. We observe that the algorithm Safe-Risky given by Koren (1993) is indeed 2-processor $k$-competitive. This implies an improvement over Baruah’s result (1998), specifically, for $k=1$, Safe-Risky is 2-processor 1-competitive. More interestingly, Safe-Risky can be easily adapted to be $2\lceil \log k \rceil$-processor 2-competitive. This processor bound is indeed asymptotically tight, as we prove that no on-line algorithm is $p$-processor $O(1)$-competitive unless $p = \Omega(\log k)$. Intuitively, increasing the computational power of the on-line scheduler with extra unit-speed processors is not an effective way of attaining a constant competitive ratio.