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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Improved lower bounds for online scheduling to minimize total stretch

Koji M. Kobayashi

National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan

ARTICLE INFO

Article history:
Received 13 December 2016
Received in revised form 1 August 2017
Accepted 27 September 2017
Available online xxxx
Communicated by D.-Z. Du

Keywords:
Scheduling problem
Online problem
Total stretch minimization
Competitive analysis

ABSTRACT

Online scheduling is one of the most basic online problems. We are given a sequence of jobs in an online fashion, and they must be assigned to one of $m(\geq 1)$ equivalent processors. Each job j with the processing time $p(j)$ is given at the release time $r(j)$. An algorithm decides which processor is used to process j and must schedule it for time $p(j)$ after time $r(j)$ on the chosen processor. Many criteria have been proposed to evaluate the performance of scheduling algorithms, and Muthukrishnan et al. (FOCS 1999 and SIAM Journal on Computing 34 (2), 2005) used the *stretch* of a job j in an objective function, which is defined as the ratio of the amount of time which j spends in the system to its processing time $p(j)$. The cost of an algorithm is the total stretch to schedule all the given jobs, and our goal is to minimize it. Muthukrishnan et al. considered the case in which preemption is allowed, that is, any job can be stopped during the process and after a while the process can resume.

In this paper, we show how to construct instances to obtain various lower bounds on the competitive ratio for each m. In the instances, m jobs are given regularly for a sufficiently large time span after a maliciously chosen time. The processing times of the given jobs are taken depending on the remaining processing times of uncompleted jobs at the start time of the "burst" of jobs. We prove that for the instances, the stretch of a job completed before the burst hardly affects the evaluation of a competitive ratio. Further, we provide a job sequence given before the burst for each m. Then, we can improve the previous lower bounds for any deterministic online algorithm for each m using the instances. For example, we obtain lower bounds of 1.228, 1.257 and $21/17 \approx 1.235$ for $m = 1, 2$ and ∞, respectively. Moreover, we obtain the first non-trivial lower bounds for any randomized online algorithm for all m.

1. Introduction

The scheduling problem is one of the most fundamental problems in computer science, and many variants have been extensively studied. The most elementary setting of them has been discussed since the 1960s [13], and the definition is as follows. We are given $m(\geq 1)$ equivalent processors and a list of jobs to be processed. Each job j is specified by two parameters: the *release time* $r(j)$ and the *processing time* $p(j)$. An algorithm must schedule j on a processor for time $p(j)$ after time $r(j)$. The objective of this problem is to minimize the maximum completion time of any job (called the *makespan*), where the *completion time* of a job j denotes the time at which the algorithm completes j.

E-mail address: kobaya@nii.ac.jp.

https://doi.org/10.1016/j.tcs.2017.09.032
0304-3975/

Table 1
Competitive ratios of deterministic algorithms.

m	Upper bound	Previous lower bound	Our lower bound
1	2 [16]	1.194 [14]	1.228
2	9.82 [9]	$7/6 \approx 1.166$ [16]	1.257
3		1	$23/19 \approx 1.210$
4		$12/11 \approx 1.090$ [16]	$21/17 \approx 1.235$
5		1	$25/21 \approx 1.190$
6		$12/11 \approx 1.090$ [16]	$23/19 \approx 1.210$
∞		1, $12/11 \approx 1.090$ [16]	$21/17 \approx 1.235$

The scheduling problem has been studied in an online setting as well. In most online settings, an online algorithm does not know the total number of given jobs in advance, and each job is given over time. Specifically, the existence of a job j and its processing time $p(j)$ are revealed to an online algorithm at time $r(j)$. The performance of an online algorithm is evaluated using *competitive analysis* [8,20]. In the competitive analysis, we compare the cost of the online algorithm with that of an optimal offline algorithm (called OPT throughout this paper), which knows the whole input, namely all the given jobs together with their parameters, in advance and can schedule them at the minimum cost. If for any input, the cost incurred by the online algorithm is at most c times that incurred by OPT, then we say that the *competitive ratio* of the online algorithm is at most c.

The online scheduling problem has been discussed in many settings: whether preemption, which is interrupting processing jobs, is allowed, whether processors are equivalent, and so on. In addition, many kinds of performance measures are proposed to correctly evaluate the performance of an online algorithm ALG. The following values are used in the performance measures: the completion time $c_{ALG}(j)$ of a job j (note that the makespan is the maximum completion time), the *flow time* $f_{ALG}(j) = c_{ALG}(j) - r(j)$ of j, that is, the amount of time which j of ALG spends in the system, the *stretch* $f_{ALG}(j)/p(j)$ of j, that is, the ratio of the flow time of j to the processing time of j, and so on. Muthukrishnan et al. [16] introduced a performance measure using the stretch. Specifically, they analyzed the scheduling problem using the objective function which is to minimize the total stretch (i.e., the average stretch).

Previous results There is some previous work for the case where preemption is allowed, that is, the preemptive case, as follows. Muthukrishnan et al. [16] showed that the competitive ratio of some greedy algorithm Shortest Remaining Processing Time (SRPT) is exactly 2 for $m = 1$. Moreover, they claimed that the competitive ratio of any deterministic online algorithm is at least $7/6 \approx 1.166$ for any even m. (There is some miscalculation and their correct lower bounds on the competitive ratio are $7/6$ for $m = 2$ and $12/11 \approx 1.090$ for any even $m \geq 4$. See Appendix A.) Legrand et al. [14] proved a lower bound of 1.194 for $m = 1$. For any m, Chekuri et al. [9] developed a deterministic online algorithm whose competitive ratio is at most 9.82. The above results are summarized in Table 1.

Our results In this paper, for the preemptive case, we improve all the previous lower bounds for any deterministic online algorithm. Some of them are shown in Table 1 in this section and Table 2 in Sec. 3.3. Moreover, we present the first non-trivial lower bound for any randomized online algorithm for each m. Improving lower bounds is one of the open problems stated in [16].

Specifically, we show some key characteristic of job sequences to minimize the total stretch and present a job sequence satisfying the key characteristic to improve the previous lower bound on the competitive ratio of any deterministic online algorithm for each $m (\geq 1)$. Furthermore, we provide a lower bound of 1.089 for any randomized online algorithm for any $m (\geq 1)$ using a job sequence to show some lower bound for any deterministic online algorithm when $m = 1$.

Related results In the setting discussed in this paper, Muthukrishnan et al. [16] proved that the competitive ratio of SRPT is at most $9 + 2\sqrt{6} \approx 13.899$ for any $m (\geq 2)$. Also for any m, they claimed a lower bound of SRPT is 2.5 (there is an error in the calculation and the correct lower bound is 2). Becchetti et al. [4,5] and Chekuri et al. [9] studied the same setting without migration, that is, the setting in which once a job is assigned to a processor at some time, the job is not allowed to be assigned to any other processor anymore (Of course, preemption is allowed in this setting). The best known upper bound is 17.32 achieved by Chekuri et al. [9]. In addition, when preemption is not allowed, for any m, Becchetti et al. [4,5] showed a lower bound of $\Omega(\Delta)$ for any randomized online algorithm, where Δ is the ratio of the maximum processing time to the minimum one. This result means that preemption is essential for this setting.

Bender et al. [6] first introduced the stretch into an objective function of the scheduling problem. They presented some results with respect to two objective functions: to minimize the maximum stretch and to minimize the maximum flow time. Research on the scheduling problem with the objective function of minimizing the maximum stretch has been ongoingly studied [6,7,14,19,11]. For the multiprocessor case without preemption, Saule et al. [19] showed that the current best upper and lower bounds are $2\Delta + 1$ and $(1 + \frac{\Delta}{m+1})/2$, respectively. For the single processor case, when preemption is not allowed, Dutot et al. [11] developed an optimal online algorithm whose competitive ratio is $1 + \frac{\sqrt{5}-1}{2}\Delta$. When preemption is allowed, the current best upper and lower bounds are $O(\Delta^{1/2})$ [6,7] and $\Delta^{\sqrt{2}-1}/2$ [14], respectively, for this case.

Please cite this article in press as: K.M. Kobayashi, Improved lower bounds for online scheduling to minimize total stretch, Theoret. Comput. Sci. (2017), https://doi.org/10.1016/j.tcs.2017.09.032