Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Algorithms 57 (2005) 49–73

Journal of Algorithms

www.elsevier.com/locate/jalgor

# Optimal non-preemptive semi-online scheduling on two related machines

Leah Epstein [a,*], Lene M. Favrholdt [b]

[a] *School of Computer Science, The Interdisciplinary Center, POB 167, 46150 Herzliya, Israel*
[b] *Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark*

Received 31 October 2002

Available online 28 December 2004

**Abstract**

We consider the following non-preemptive semi-online scheduling problem. Jobs with non-increasing sizes arrive one by one to be scheduled on two uniformly related machines, with the goal of minimizing the makespan. We analyze both the optimal overall competitive ratio, and the optimal competitive ratio as a function of the speed ratio ($q \geqslant 1$) between the two machines. We show that the greedy algorithm LPT has optimal competitive ratio $\frac{1}{4}(1+\sqrt{17}) \approx 1.28$ overall, but does not have optimal competitive ratio for every value of $q$. We determine the intervals of $q$ where LPT is an algorithm of optimal competitive ratio, and design different algorithms of optimal competitive ratio for the intervals where it fails to be the best algorithm. As a result, we give a tight analysis of the competitive ratio for every speed ratio.


*Keywords:* Scheduling; Makespan; Semi-online; Competitive ratio; LPT

## 1. Introduction

*The problem*

In this paper we study non-preemptive semi-online scheduling on two uniformly related machines. In the model of uniformly related machines, each machine has a *speed* and each

* Corresponding author.
*E-mail addresses:* lea@idc.ac.il (L. Epstein), lenem@imada.sdu.dk (L.M. Favrholdt).

0196-6774/$ – see front matter 
doi:10.1016/j.jalgor.2004.11.002

job has a *size* which is the time it takes to complete it on a machine with unit speed. The jobs arrive one by one in order of non-increasing sizes. Each job must be assigned to one of the machines without any knowledge of future jobs (except for a bound on their size that follows from the size of the current job). Since the jobs are known to have non-increasing sizes, the problem cannot be seen as online but semi-online. We study the non-preemptive case, where it is not allowed to split a job in more parts and run the various parts on different machines. The goal is to minimize the *makespan*, i.e., the latest completion time of any job.

The processing time of a job on a given machine is also called the *load* of the job on that machine. The load of a machine is the sum of the loads of the jobs assigned to it. Thus, the makespan is the maximum load of any machine.

Since we study the case of two machines, the important parameter is the speed ratio $q \geqslant 1$ between the two machines. Without loss of generality, we assume that the faster machine has speed 1, and the other machine has speed $\frac{1}{q}$. We denote the faster machine by $M_1$ and the other machine by $M_q$.

*Preliminaries*

The quality of a semi-online algorithm, similarly to online algorithms, is measured by the *competitive ratio* which is the worst case ratio of the cost (the makespan, in this paper) of the semi-online algorithm to the cost of an optimal offline algorithm which knows the whole sequence in advance.

The semi-online algorithm under consideration as well as its makespan is denoted by SONL. Similarly, the optimal offline algorithm as well as its makespan is denoted by OPT. Thus, the competitive ratio of an algorithm SONLis

$$C = \inf\{c \mid \text{SONL} \leqslant c \cdot \text{OPT}, \text{ for any input sequence}\}.$$

For any $c \geqslant C$, SONL is said to be $c$-*competitive*.

The greedy algorithm LPT (Longest Processing Time first) was originally designed by Graham [6] for offline scheduling on identical machines. It sorts the jobs by non-increasing sizes and schedules them one by one on the least loaded machine. This algorithm also works for the semi-online version where the jobs arrive in order of non-increasing sizes. The natural extension for uniformly related machines is as follows:

**Algorithm LPT.** Assign each arriving job $J$ (of size $p$) to the machine that would finish it first. Formally, for each machine $i$ let $L_i$ be its load before the arrival of $J$. The job $J$ is assigned to the fastest machine $i$ for which $L_i + \frac{p}{s_i}$ is minimized.

*Previous work*

All previous study of this problem on non-identical machines involves a study of the LPT algorithm. For two machines, Mireault, Orlin and Vohra [7] give a complete analysis of LPT as a function of the speed ratio. They show that the interval $q \geqslant 1$ is partitioned into nine intervals, and introduce a function which gives the competitive ratio in each interval (they consider the offline problem, so they do not use the term competitive ratio). Some properties of LPT were already shown earlier. Graham [6] shows that the exact approximation ratio of LPT is $\frac{7}{6}$ for two identical machines. Seiden, Sgall and Woeginger