Online Matching On a Line

Bernhard Fuchs [1]

Zentrum für Angewandte Informatik Köln
Universität zu Köln, Weyertal 80, D-50931 Köln

Winfried Hochstättler

Department of Mathematics, BTU Cottbus
Postfach 10 13 44, D-03013 Cottbus

Walter Kern

Department of Applied Mathematics
University of Twente, P.O.Box 217, NL-7500 AE Enschede

Abstract

We prove a lower bound $\rho \geq 9.001$ for the competitive ratio of the so-called online matching problem on a line. As a consequence, the online matching problem is revealed to be strictly more difficult than the "cow problem".

Key words: online algorithms, competitive analysis, matching

1 Introduction

We consider a special class of online server problems, where a number of servers, located on the real line, is to serve a sequence of requests $r_1, r_2, \ldots, r_k \in \mathbb{R}$. In contrast to classical server problems (cf, e.g. [3]), however, each server can serve at most one request. So the optimal offline solution is the min cost matching of the requests into the set of server positions s_i. The problem is

Email addresses: `bfuchs@zpr.uni-koeln.de` (Bernhard Fuchs), `hochstaettler@math.tu-cottbus.de` (Winfried Hochstättler), `w.kern@math.utwente.nl` (Walter Kern).

[1] supported by Deutsche Forschungsgemeinschaft, Graduiertenkolleg Scientific Computing, GRK 192/5-02

therefore also known as the *online matching problem on a line* [5]. As an application, consider a ski rental with different ski lengths $s_1, s_2, \dots$ at its disposal to meet requested lengths $r_1, r_2, \dots$ of entering clients.

For notational convenience, we consider a "universal" instance with infinitely many servers, one at each integer $s \in \mathbb{Z}$. One may equally well consider finite versions with servers at positions $s_1, \dots, s_n \in \mathbb{R}$, requests $r_1, \dots r_k \in \mathbb{R}$ and $k \leq n$ (or even $k = n$). These are, however, easily seen to be of approximately the same difficulty: Any ρ-competitive algorithm for one version with $\rho < \bar{\rho}$ implies a $\tilde{\rho}$-competitive version for the other with $\tilde{\rho} < \bar{\rho}$.

An online matching algorithm is *ρ-competitive* if, after serving $r_1, \dots, r_t$ ($t \in \mathbb{N}$), the current length L of the online matching constructed so far is at most ρ times the current optimal matching cost. It is a challenging open question to prove or disprove the existence of ρ-competitive online algorithms with finite competitive ratio ρ.

The basic difficulty for an online algorithm is to decide which server to use for matching a new request r. There are essentially two choices: Either the server s_- that is closest to r from left or the server s_+ that is closest to r from right (among those servers that are currently still unmatched). Indeed, serving r from a server at $s < s_-$ can be interpreted as moving s to s_- and serving r from s_-.

Assume, e.g., the first $2m_0$ requests are at $r = 0, \pm 1, \pm 2, \dots, \pm(m_0 - 1), 0$. The first $2m_0 - 1$ requests will then be served from the servers at these positions, whereas the last request $r = 0$ will be served, say, from $s = -m_0$. Assume the following requests $r_{2m_0+1}, r_{2m_0+2}, \dots$ are then exactly at the positions where a server has just been moved off to serve the previous request. So $r_{2m_0+1} = -m_0$ etc. In order to stay ρ-competitive, the online algorithm may first serve a number of requests from left, but must eventually *switch* to serving some request $r = i \leq -m_0$ from right, i.e., from $s = m_0$. (Indeed, $|i| \leq \rho/2m_0$). It may then continue to serve a number of requests from right, but eventually it will have to switch again, serving some request $r = j \geq m_0$ from left etc. Thus the online algorithm basically must behave like the famous cow searching for a bridge to cross the river ([1], [4]). We therefore refer to the request sequence constructed as above as a *cow sequence* with parameter m_0, started at $r = 0$.

This analogy yields a lower bound of $\rho \geq 9$ for the competitive ratio of any online algorithm for matching on a line (even for cow sequences), cf. [1] or section 2. The main purpose of our paper is to slightly improve this bound to $\rho \geq 9.001$. Since a 9-competitive algorithm for the "cow problem" is known, our result proves the online matching problem to be strictly more difficult than the cow problem. In Section 4 we analyze online algorithms based on so-called *work functions* and show that they have infinite competitive ratio.

2 Cow Sequences

Consider an online algorithm for the matching problem on a line and assume it has already served requests $r_1, \ldots, r_k \in \mathbb{Z}$. We denote by L the (length of) the matching constructed so far and refer to it as the *current travel length.* M^* denotes the (length of) the current optimal matching from $R = \{r_1, \ldots, r_t\}$ into $\mathbb{Z}$. In addition, we introduce the *current matching* M: Assume that the online algorithm has served the currently known set of requests $R = \{r_1, \ldots, r_t\}$ from servers $S = \{s_1, \ldots, s_t\}$. Then M is the (length of) the optimal matching from S to R. We stress that, in general, this is different from both L and M^*.

We define a *potential function* on the sequence of current matchings to analyze the behaviour of a ρ-competitive algorithm for the matching problem and provide a new proof for the lower bound $\rho \geq 9$ on cow sequences.

3 More Cows

To disprove the existence of a $(9+\varepsilon)$-competitive algorithm for online matching turns out to be a somewhat more complicated task. The basic idea is to run two (or more) cow sequences, enforcing additional complications when two matchings belonging to different cows are merged. First we force two cows of appropriate size to merge, and afterwards analyze the resulting combined potential. This analysis yields that a 9.001-competitive algorithm for matching on a line cannot exist.

References

[1] R. Baeza-Yates, J. Culberson and G. Rawlins, Searching in the plane. *Information and Computation* **106**, 1993, 234–252.

[2] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis.* Cambride University Press, 1998.

[3] E. Koutsoupias and C. Papadimitriou, On the k-server conjecture. *Journal of the ACM* **42**(5), 1995, 971–983.

[4] C. Papadimitriou and M. Yannakakis, Shortest paths without a map. *Theoretical Computer Science* **84**, 1991, 127–150.

[5] K. Pruhs, Personal Communication (2000).