Dipartimento di Informatica e Sistemistica “Antonio Ruberti”
Università degli Studi di Roma “La Sapienza”

Lazy on–line algorithms for metrical service system

G. Ausiello, V. Bonifaci, L. Laura

Technical Report n. 7
2004

Aracne

I *Technical Reports* del Dipartimento di Informatica e Sistemistica "Antonio Ruberti" svolgono la funzione di divulgare tempestivamente, in forma definitiva o provvisoria, i risultati di ricerche scientifiche originali. Per ciascuna pubblicazione vengono soddisfatti gli obblighi previsti dall'art. 1 del D.L.L. 31.8.1945, n. 660 e successive modifiche.
Copie della presente pubblicazione possono essere richieste alla Redazione.

Dipartimento di Informatica e sistemistica "Antonio Ruberti"
Università degli studi di Roma "La Sapienza"

Via Eudossiana, 18 - 00184 Roma
Via Buonarroti, 12 - 00185 Roma
Via Salaria, 113 - 00198 Roma

www.dis.uniroma1.it

www.aracne–editrice.it
info@aracne–editrice.it

00173 Roma
via Raffaele Garofalo, 133 A/B
(06) 72672222 – (06) 93781065
telefax 72672233

ISBN 88–7999–693–2

I edizione: marzo 2004

Finito di stampare nel mese di marzo del 2004
dalla tipografia « Grafica Editrice Romana S.r.l. » di Roma
per conto della « Aracne editrice S.r.l. » di Roma
Printed in Italy

Lazy on-line algorithms for metrical service systems

G. Ausiello* V. Bonifaci* L. Laura*

Abstract

In this paper we consider the problem of efficiently serving a sequence of requests in an on-line fashion, where every request is a subset of a metric space and serving a request means moving the unique server to a point of the subset. Problems of this kind are called metrical service systems (MSS) and are a generalization of problems such as paging, the k-server problem and the so-called CNN problem.

In this context, we give a general definition of lazy algorithm for MSS and show that lazy algorithms are as powerful as general algorithms; in particular, the optimal algorithm is lazy without loss of generality. We also analyze two special cases of MSS, related to the CNN problem, and give an upper bound on their competitive ratio using the laziness of OPT.

Finally, we present the results of an application of an algorithm described by Chrobak and Larmore to decide the c-competitiveness of an MSS to some MSS on finite spaces, including particular cases of the CNN problem.

Keywords: on-line algorithms, metrical service systems, lazy algorithms, CNN problem.

1 Introduction

In the classical approach to optimization problems, it is assumed that all data are present on input before the computation begins. This kind of optimization, called *off-line*, is not always fit to model real problems. In fact, in many applications, such as scheduling or routing, the input of the problem is communicated in successive steps, and partial output has to be computed without knowledge of data that will arrive later. This kind of optimization is called *on-line*.

*Dip. di Informatica e Sistemistica Università di Roma "La Sapienza" Via Salaria 113 00198 Roma Italy. E-mail: {ausiello,bonifaci,laura}@dis.uniroma1.it. Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 ALCOM-FT "Algorithms and Complexity in Future Technologies", and by the Italian research project ALINWEB: "Algoritmica per Internet e per il Web", MIUR – Programmi di Ricerca di Rilevante Interesse Nazionale.

While classical algorithm analysis focuses on the worst case time (or space) complexity, on-line algorithm analysis focuses on the worst case quality of the computed solution. In fact, for an on-line algorithm it is normally impossible to find the optimal solution. The standard way to measure the quality of an on-line algorithm is to consider the worst case ratio between the cost incurred by the on-line algorithm on a sequence of requests and the optimal cost incurred by an algorithm that knows the sequence in advance. Therefore, an algorithm is said to be c-competitive if for every input the cost it pays is at most c times the cost incurred by the optimal off-line algorithm.

This paper considers some problems belonging to the class of *metrical service systems* (MSS). A metrical service system is defined over a metric space in which a mobile server moves to visit some regions of the space. The points to visit are defined by a sequence of requests, each request giving a subset of the space that the server will necessarily move in. Depending on the space and on the requested subsets, different metrical service systems will arise. Among the applications of MSS we want to mention weighted paging, k-headed disk scheduling and some robot exploration problems.

In this paper, we introduce the concept of *lazy* algorithm for a MSS extending the definition of lazy algorithm for the k-server and paging problems and we show that the fact that every algorithm can be turned into a lazy algorithm with at least the same performance holds also for this extended case. Using the consequence that every optimal algorithm is lazy without loss of generality, we prove upper bounds on the competitive ratio of two variations of the CNN problem. One of these upper bounds was already proved in [YI01] but we believe that the laziness of OPT leads to a simplified proof. We also prove a matching lower bound on the competitive ratio of this problem. Finally, we show the results of an application of an algorithm by Chrobak and Larmore that decides the c-competitiveness of an MSS to some small finite instances of the metrical service systems considered in the paper.

2 Metrical service systems and related problems

A metrical service system is a pair $\mathcal{S} = (M, T)$ where M is a metric space and T is a set of allowed requests or tasks. Every request $r \in T$ is a subset of M. The input is given by the initial server position x_0 and by a sequence of requests $\sigma = r_1 r_2 \dots r_n$. The server is initially in x_0 and has to serve request r_t at time t, moving to a point $x_t \in r_t$. The cost incurred by the server is the distance it travels.

As an example of MSS, consider the *k-point request problem* or k-PRP, considered in [FFK+91]. In this problem every request is a finite subset of the metric space having cardinality at most k. The server has to choose for every request in which of the k points it will move and then pay a cost