Information Processing Letters 114 (2014) 239–246

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A k-server problem with parallel requests and unit distances

R. Hildenbrandt

Department of Mathematics, Ilmenau Technical University, PF 10 06 65, 98684 Ilmenau, Germany

ARTICLE INFO

Article history:
Received 7 November 2011
Received in revised form 16 December 2013
Accepted 16 December 2013
Available online 8 January 2014
Communicated by Ł. Kowalik

Keywords:
Analysis of algorithms
Competitive analysis
Server problems
Combinatorial problems

ABSTRACT

In this paper we consider k-server problems with parallel requests where several servers can also be located on one point. We will distinguish the surplus-situation where the request can be completely fulfilled by means of the k servers and the scarcity-situation where the request cannot be completely met. We use the method of the potential function by Bartal and Grove [2] in order to prove that a corresponding Harmonic algorithm is competitive for the more general k-server problem in the case of unit distances. For this purpose we partition the set of points in relation to the online and offline servers' positions and then use detailed considerations related to sets of certain partitions.

1. Introduction

In this paper a generalized k-server problem with parallel requests where several servers can also be located on one point is discussed. The investigation of the generalized k-server problem was initiated by an operations research problem which consists of optimal conversions of machines or moulds (see [4], [5] or [7]). It is sensible in the case of parallel requests to distinguish the surplus-situation where the request can be completely fulfilled by means of the k servers and the scarcity-situation where the request cannot be completely met.

The k-server problem was introduced by Manasse, McGeoch and Sleator [9]. Meanwhile it is the most studied problem in the area of competitive online problems. Historical notes on k-server problems can be found in the book by A. Borodin and R. El-Yaniv [3] (Sections 10.9 and 11.7)[1] or also in the paper by Y. Bartal and E. Grove [2]. There the two important results are the competitiveness of the deterministic work-function algorithm (see E. Koutsoupias and C. Papadimitriou [8]) and of the randomized Harmonic k-server algorithm against an adaptive online adversary (see Y. Bartal and E. Grove [2]).

The work-function algorithm is an inefficient algorithm (with a good competitive ratio). In contrast the Harmonic k-server algorithm is memoryless and time-efficient. For this reason we first want to focus on a corresponding Harmonic k-server algorithm for the generalized k-server problem.

If one tries to generalize the proof by Y. Bartal and E. Grove [2] several sub-chains with different lengths must be considered and one will see that the computation of the weights $f(j)$ is not possible. In this paper we consider the generalized k-server problem in the case of unit distances. The (usual) k-server problem with unit distances is known as the paging problem and the Harmonic k-server algorithm as RAND algorithm (see [3], Chapters 3 and 4; see also [11]). Raghavan and Snir [10] have shown that the RAND algorithm is k-competitive against an adaptive online adversary. Although a lot more feasible requests can occur in the case of the generalized k-server problem we will show that the corresponding Harmonic k-server algorithm is $\max\{k, R(k) - k + 1\}$-competitive (where $R(k)$ is a bound of the requests related to the scarcity-situation, see Theorem 1) and k-competitive (just as RAND), if only the surplus-situation is allowed.

Parallel requests imply that the proof is more difficult as in the case of the RAND algorithm. We will apply a potential function in the proof and then partition the set of

[1] For basic knowledge of (usual) k-server problems see also [3], Chapters 10 and 11.

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2013.12.011

points in relation to the online and offline servers' positions. Finally detailed considerations related to sets of certain partitions will be used.

2. The formulation of the model

Let $k \geqslant 1$ be an integer, and $M = (M, d)$ be a finite metric space where M is a set of points with $|M| = N$. An algorithm controls k mobile servers, which are located on points of M. Several servers can be located on one point. The algorithm is presented with a sequence $\sigma = r^1, r^2, \ldots, r^n$ of requests where a request r is defined as an N-ary vector of integers with $r_i \in \{0, 1, \ldots, k\}$ ("parallel requests"). The request means that r_i server is needed on point i $(i = 1, 2, \ldots, N)$. We say a request r is served if $\left\{ \begin{smallmatrix} \text{at least} \\ \text{at most} \end{smallmatrix} \right\}$ r_i servers lie on i $(i = 1, 2, \ldots, N)$ in case $\left\{ \begin{smallmatrix} C[r,k] \\ C[k,r] \end{smallmatrix} \right\}$. $C[r, k]$ denotes the case $\sum_{i=1}^{N} r_i \leqslant k$ (surplus-situation, the request can be completely fulfilled) and $C[k, r]$ denotes the case $\sum_{i=1}^{N} r_i \geqslant k$ (scarcity-situation, the request cannot be completely met, however it should be met as much as possible). By moving servers, the algorithm must serve the requests $r^1, r^2, \ldots, r^n$ sequentially. For any request sequence σ and any generalized k-server algorithm $ALG_{p(arallel)}$, $ALG_p(\sigma)$ is defined as the total distance (measured by the metric d) moved by the ALG_p's servers in servicing σ.

In this paper we will show that the corresponding Harmonic k-server algorithm attains a competitive ratio of $\max\{k, R(k) - k + 1\}$ (see Theorem 1) against an adaptive online adversary in the case of unit distances (for the definitions of competitive ratio and adaptive online adversary see [2] or [3], Sections 4.1 and 7.1). Analogous to [3], p. 152, working with lazy algorithms ALG_p is sufficient. For that reason we define the set of feasible servers' positions with respect to s and r in the following way

$$\hat{A}_{N;k}(s, r) = \left\{ s' \in P_N(k) \middle| \begin{array}{l} r_i \leqslant s'_i \leqslant \max\{s_i, r_i\},\ i = 1, \ldots, N,\ \text{in } C[r,k] \\ \min\{s_i, r_i\} \leqslant s'_i \leqslant r_i,\ i = 1, \ldots, N,\ \text{in } C[k,r] \end{array} \right\} \tag{1}$$

where

$$P_N(k) := \left\{ s \in \mathbb{Z}_+^n \middle| \sum_{i=1}^{N} s_i = k \right\}. \tag{2}$$

The metric d implies that $(P_N(k), \hat{d})$ is also a finite metric space where $\hat{d}$ are the optimal values of the classical transportation problems with availabilities s and requirements s' of $P_N(k)$: $\sum_{i=1}^{N} \sum_{j=N}^{N} d(i, j) x_{ij} \to \min$ subject to $\sum_{j=1}^{N} x_{ij} = s_i$ $\forall i$, $\sum_{i=1}^{N} x_{ij} = s'_j$ $\forall j$, $x \in \mathbb{Z}_+^n \times \mathbb{Z}_+^n$ (see [6], Lemma 3.6).

The corresponding $HARMONIC_p$ k-server algorithm operates as follows: Serve a (not completely covered) request r with randomly chosen servers so that for the (new) servers' positions $s' \in \hat{A}_{N;k}(s, r)$ is valid with respect to the previous servers' positions s and the request r. More precisely, $HARMONIC_p$ leads to $s' \in \hat{A}_{N;k}(s, r)$ with probability

$$\frac{\frac{1}{\hat{d}(s,s')}}{\sum_{s'':\, s'' \in \hat{A}_{N;k}(s,r)} \frac{1}{\hat{d}(s,s'')}}. \tag{3}$$

3. The competitiveness of $HARMONIC_p$ in case of unit distances

Unit distances mean that $d(i, j) = 1$ $\forall i \neq j$. Thus, $\hat{d}(s, s') = \sum_{i=1}^{N} \frac{1}{2} |s_i - s'_i|$ for $s, s' \in P_N(k)$ follows and (1) yields

$$\hat{d}(s, s') = \begin{cases} \sum_{i: r_i^t > s_i} (r_i^t - s_i) & \text{in } C[r, k] \\ \sum_{i: r_i^t < s_i} (s_i - r_i^t) & \text{in } C[k, r] \end{cases}$$

for every $s' \in \hat{A}_{N;k}(s, r)$. Then $s' \in \hat{A}_{N;k}(s, r)$ is chosen randomly and uniformly with probability $\frac{1}{|\hat{A}_{N;k}(s,r)|}$ among all elements of $\hat{A}_{N;k}(s, r)$ by $HARMONIC_p$.

Next we give an example that an additional assumption (as $\sum_{i \in M} r_i^t \leqslant R(k)$ in the following theorem) in the case $C[k, r^t]$ is necessary in order to prove the competitiveness. Let $k = 1$ and $\sum_{i \in M} r_i^t$ not bounded in the case $C[k, r^t]$.

The adversary moves his server to another point if and only if the servers of the adversary and of the algorithm are located on the same point.

The adversary produces the request sequence with $r^t = (1, \ldots, 1, 0, 1, \ldots, 1)$ in step t where $r_{i_0}^t = 0$ for this point i_0 on which the server of the algorithm is located. Then the cost of the algorithm is equal to 1 in every step.

The $HARMONIC_p$ algorithm moves its server to a point $i \neq i_0$ with the probability $\frac{1}{N-1}$. This also means that then the servers of the adversary and of the algorithm are located on the same point with the probability $\frac{1}{N-1}$.

Hence

$$E\big[cost(HARMONIC_p\ algorithm)\big] = (N - 1) E\big[cost(adversary)\big]$$

follows in relation to the expected costs and no $C(k)$ (independent of N) exists such that the $HARMONIC_p$ k-server algorithm is $C(k)$-competitive.

Theorem 1. *The $HARMONIC_p$ k-server algorithm attains a competitive ratio of $C(k) = \max\{k, R(k) - k + 1\}$ against an adaptive online adversary in case of unit distances if $\sum_{i \in M} r_i^t \leqslant R(k)$ $\forall t$ for given $R(k) > k$.*[2]

Proof. We will use a potential function (see [2]) to prove the statement. As described in [2], the potential function is a function of the current locations of the online and offline servers. Intuitively, Φ is an upper bound on the expected amount of work the algorithm can be forced to do if the offline servers do not move. In case of unit distances it is sufficient to use the following simple potential function

[2] This condition is important for case $C[k, r^t]$. (According to the introduced model $\sum_{i \in M} r_i^t \leqslant k$ is true in case $C[r^t, k]$.) See also the above mentioned example.