DIMACS Technical Report 99-59
November 1999

Note on the work function algorithm

by

Béla Csaba[1,2]
Dept. of Computer Science
Rutgers University

[1]Affiliated Graduate Student Member
[2]Research partially supported by a DIMACS fellowship

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Telcordia Technologies (formerly Bellcore) and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

We prove that the work function algorithm is $(n-1)$-competitive for the k-server problem, where n is the number of points in the metric space. This gives improved upper bounds when $k+3 \leq n \leq 2k-1$; in particular, it shows that the work function algorithm is optimal for $k = n-1$.

1 Introduction

We give a short introduction to the deterministic k-server problem ([ST], [MMS]). There is a metric space $\mathcal{M}$ with a distance function $d(.,.)$ on it. Let us denote $|\mathcal{M}|$ by n. There are k $(1 < k < n)$ mobile servers initially residing on the pointset I, no two on the same point. Repeatedly requests are generated by an off-line adversary, and we have to satisfy them in an on-line fashion. Satisfying a request is putting a server on the requested point of the metric space. When moving a server, a cost occurs which is the distance of the previous and the present position of the server which moved. While the adversary has the advantage that it satisfies the request sequence at the end thus it can satisfy them optimally paying the least, an on-line algorithm pays after every request. In competitivity analysis we compare the on-line cost with the optimal (off-line) cost, we are looking for an algorithm with the best competitive ratio. In [MMS] it is proved that no on-line algorithm can have competitive ratio less than k. They also gave optimal on-line algorithm for two special cases: when $k = 2$ and when $k = n - 1$. They conjectured that in every case there is an optimal, k- competitive on-line algorithm. There are other cases with optimal on-line algorithms: when the metric space is tree-like ([CL]), when every distance is the same (paging problem, [ST]), or for the weighted cache problem ([CKPV]). So far the best general on-line algorithm is given by Koutsoupias and Papadimitriou ([KP1]). They proved that the work function algorithm is $(2k - 1)$-competitive. In another paper ([KP2]) they also showed that the work function algorithm is k-competitive if $k + 2 = n$. In this paper we give a simple proof that their algorithm is $(n - 1)$- competitive as well, hence for $k + 3 \leq n \leq 2k - 1$ this gives the best known upper bounds, and for $k = n - 1$ it proves that the work function algorithm is optimal.

Recently Koutsoupias in [K] indpendently proved this result. His approach was different, while our proof is based on potential functions, he analyzed the k-server problem with weak adversaries.

The outline of this note is as follows: in the next chapter we present a brief description of the work function algorithm, and state some relevant properties of it, without proofs. Then in the third chapter we prove our result.

2 Work Function Algorithm

Let $\mathcal{M}$, I and the request sequence ϱ is given. Then the work function w_ϱ maps configurations to nonnegative real numbers: $w_\varrho(X)$ is the optimal cost of servicing ϱ starting at I and ending at X. For a work function w_ϱ the resulting work function after the request r is $w_{\varrho r}$. The following proves to be very useful:

Fact 2.1 *For all X $w_{\varrho r}(X) \geq w_\varrho(X)$.*

Let us now define the algorithm itself.

Definition 2.2 (Work Function Algorithm) *Let ϱ be a request sequence and A be the configuration of the servers after satisfying ϱ. The work function algorithm services the new request at r by moving one of the servers to the configuration $B = A - a + r$ (i.e.,the server moves from a to r), where $w_{\varrho r}(B) + d(a, r)$ is minimal.*

The so called extended cost of a move, $\max_X\{w_{\varrho r}(X) - w_\varrho(X)\}$ in some way encapsulates the optimal and the on-line costs.

Fact 2.3 *If the sum of the extended costs is bounded above by $c + 1$ times the optimal cost plus a constant, then the work function algorithm is c-competitive.*

The notion of minimizer configurations is also an important one.

Definition 2.4 *A configuration A is called a minimizer with respect to the point a with respect to w_ϱ if*

$$w_\varrho(A) - \sum_{x\in A} d(a, x) = \min_X\{w_\varrho(X) - \sum_{x\in X} d(a, x)\}. \tag{1}$$

We finish decribing the relevant definitions and facts by the following lemma, the *Duality Lemma.*

Lemma 2.5 (Duality Lemma) *Let w_ϱ be a work function, and let $w_{\varrho r}$ be the resulting work function after request r. Then any minimizer A of r with respect to w_ϱ is also a minimizer of r with respect to $w_{\varrho r}$, and the extended cost of servicing the request r is $w_{\varrho r}(A) - w_\varrho(A)$.*

3 Proof of the $(n-1)$-competitivity

Let ϱ be the request sequence, and let the next request be r. Denote A_x the minimizer configuration of x with respect to w_ϱ, and B_x the minimizer configuration of x with respect to $w_{\varrho r}$. Let us define a potential function P_ϱ:

$$P_\varrho = \sum_x \left[w_\varrho(A_x) - \sum_{a\in A_x} d(x, a) \right]. \tag{2}$$

Analoguosly, we can define $P_{\varrho r}$. Observe, that $P_\emptyset$, the initial potential function is a constant. By $opt(\varrho)$ we denote the optimal cost of satisfaction of the request sequence ϱ.

Lemma 3.1 *(1) $P_{\varrho r} - P_\varrho \geq$ extended cost of the move from ϱ to ϱr and (2) $P_\varrho - n \cdot opt(\varrho) \leq$ constant.*

Proof. For proving (1) observe that A_r is a minimizer of r with respect to w_ϱ and $w_{\varrho r}$. Thus, $w_{\varrho r}(A_r) - \sum_{a \in A_r} d(r,a)$ is also a minimal expression, and $w_{\varrho r}(A_r) - w_\varrho(A_r)$= extended cost. Let now v be another point of $\mathcal{M}$, then $w_\varrho(A_v) - \sum_{a \in A_v} d(v,a) \leq w_\varrho(B_v) - \sum_{b \in B_v} d(v,b) \leq w_{\varrho r}(B_v) - \sum_{b \in B_v} d(v,b)$. This proves (1). For (2), observe that P_ϱ is the sum of optimal costs plus a constant. □

Theorem 3.2 *The work function algorithm is $(n-1)$-competitive.*

Proof. Summing up the potential function values over all requests we can see from Lemma 3.1 that this sum is at least the sum of the extended costs plus a constant. By Lemma 2.3 we can conclude the statement of the theorem. □

References

[CKPV] Chrobak, M., Karloff, H., Payne, T. and Vishwanathan, S., *New results on Server Problems*, SIAM Journal on Disc. Math.,4:172-181, 1991

[CL] Chrobak, M. and Larmore, L., *An optimal online algorithm for k servers on trees*, SIAM Journal on Computing, 20:144-148, 1991

[K] Koutsoupias, E., *Weak adversaries for the k-server problem*, FOCS 99, 444-449

[KP1] Koutsoupias, E. and Papadimitrou, C., *On the k-Server Conjecture*, STOC 94, pp. 507-511

[KP2] Koutsoupias, E. and Papadimitrou, C., *The 2-evader problem*, Information Processing Letters, 57(5):249-252, 1996

[MMS] Manasse, M. S., McGeoch L. A. and Sleator, D. D., *Competitive Algorithms for Server Problems*, Journal of Algorithms 11 (1990), pp. 208-230

[ST] Sleator, D. D., Tarjan, R. E., *Amortized Efficiency of List Update and Paging Rules*, Comm. of the ACM, February 1985, pp. 202-208