Settling the Randomized $k$-sever Conjecture on Some Special Metrics

Wenbin Chen

Abstract

The $k$-server problem is one of the most fundamental online problems, which is introduced by Manasse, McGeoch and Sleator [27, 28]. The problem is to schedule $k$ mobile servers to serve a sequence of requests in a metric space with the minimum possible movement distance. The randomized $k$-sever conjecture states that there exists $O(\log k)$-competitive randomized algorithms for the $k$-sever problem. The conjectures has been open for over 24 years.

In this paper, we settle the randomized $k$-sever conjecture for the following metric spaces: line, circle, Hierarchically well-separated tree (HST), if $k = 2$ or $n = k + 1$ for arbitrary metric spaces. Specially, we show that there are $O(\log k)$-competitive randomized $k$-sever algorithms for above metric spaces. For any general metric space with $n$ points, we show that there is an $O(\log k \log n)$-competitive randomized $k$-sever algorithm, which improved the previous best competitive ratio $O(\log^2 k \log^3 n \log \log n)$ by Nikhil Bansal et al. (FOCS 2011, pages 267-276).

Above algorithms refer to lazy algorithms, i.e., algorithms move only one server to serve the requested point only if the requested point is not served. In addition, we still show that there exists a $O(\log k)$-competitive randomized non-lazy algorithm for the $k$-sever problem.

Keywords: $k$-sever problem; Online algorithm; Primal-Dual method; Randomized algorithm;

*Email:cwb2011@gzhu.edu.cn
†Department of Computer Science, Guangzhou University, P.R. China
‡State Key Laboratory for Novel Software Technology, Nanjing University, P.R. China
1 Introduction

In online computation area, the $k$-sever problem is one of most influential and extensively studied problems and is called as the "holy grail" problem in this field by many researchers. The $k$-sever problem refer to the following problem. Suppose there is a metric space $S$ with $n$ points and $k$ mobile servers located at some positions of the metric space. Given a requested points sequence $\rho$, every request must be served by moving a server to the requested point, only if there is no server at that point. For serving a requests sequence $\rho$, the cost of an algorithm is defined to be the total distance traveled by these servers. Given a request sequence, the task is to devise an online algorithm with the minimum cost. An online algorithm has to design a method serving the current request without any knowledge of the future requests, i.e. it has to decide to how to move a server to the requested point based on the previous and current requests. In contrast, an offline algorithm can move a server by the entire request sequence.

In 1990, Manasse et al. introduced the $k$-sever problem as a generalization of several important online problems such as paging and caching problems [28] (Its conference version is [27]), in which they proposed a 2-competitive algorithm for the 2-sever problem and a $n-1$-competitive algorithm for the $n-1$ sever problem in a $n$-point metric space. They still showed that any deterministic online algorithm for the $k$-sever problem is of competitive ratio at least $k$. They proposed the well-known $k$-sever conjecture: for the $k$-sever problem on any metric space with more than $k$ different points, there exists a deterministic online algorithm with competitive ratio $k$.

It was in [28] shown that the $k$-sever conjecture holds for two special cases: $k = 2$ and $n = k+1$. The $k$-sever conjecture also holds for the $k$-sever problem on a uniform metric. The special case of the $k$-sever problem on a uniform metric is called the paging (also known as caching) problem. Slator and Tarjan have proposed a $k$-competitive algorithm for the paging problem [30]. For some other special metrics such as line, tree, there existed $k$-competitive online algorithms. Yair Bartal and Elias Koutsoupias show that the Work Function Algorithm for the $k$-sever problem is of $k$-competitive ratio in the following special metric spaces: the line, the star, and any metric space with $k+2$ points [15]. Marek Chrobak and Lawrence L. Larmore proposed the $k$-competitive Double-Coverage algorithm for the $k$-sever problem on trees [20].

For the $k$-sever problem on the general metric space, the $k$-sever conjecture remain open. Fiat et al. were the first to show that there exists an online algorithm of competitive ratio that depends only on $k$ for any metric space: its competitive ratio is $\Theta((k!)^3)$. The bound was improved later by Grove who showed that the harmonic algorithm is of competitive ratio $O(k^2k^k)$ [24]. The result was improved to $(2^k \log k)$ by Y.Bartal and E. Grove [13]. A significant progress was achieved by Koutsoupias and Papadimitriou, who proved that the work function algorithm is of competitive ratio $2k-1$ [26].

Although previous mentioned results are about the deterministic online algorithms, people are also interested in designing randomized online algorithms for the $k$-sever problem. Generally, people believe that randomized online algorithms can produce better competitive ratio than their deterministic counterparts. For example, there are several $O(\log k)$-competitive algorithms for the paging problem and a $\Omega(\log k)$ lower bound on the competitive ratio in [23, 29, 1, 4]. Although there were much work [16, 12, 14], the $\Omega(\log k)$ lower bound is still best lower bound in the randomized case. Recently, N. Bansal et al. propose the first polylogarithmic-competitive randomized algorithm for the $k$-sever problem on a general metric space [3]. Their randomized algorithm is of competitive ratio $O(\log^2 k \log^3 n \log \log n)$ for any metric space with $n$ points, which improves on the deterministic $2k-1$ competitive ratio of Koutsoupias and Papadimitriou whenever $n$ is
sub-exponential $k$.

For the $k$-server problem on the general metric space, it is widely conjectured that there is an $O(\log k)$-competitive randomized algorithm, which is called as the randomized $k$-server conjecture. For the paging problem (it corresponds to the $k$-server problem on a uniform metric), there is $O(\log k)$-competitive algorithms \cite{23,29,1}. For the weighted paging problem (it corresponds to the $k$-server problem on a weighted star metric space), there were also $O(\log k)$-competitive algorithms \cite{4,9} via the online primal-dual method. More extensive literature on the $k$-server problem can be found in \cite{25,17}.

Our Results and Techniques

We show that there are $O(\log k)$-competitive randomized $k$-server algorithms for for the following metric spaces: line, circle, HST, if $k = 2$ or $n = k + 1$ for arbitrary metric spaces, which settle the randomized $k$-server conjecture for these metric spaces. We also show that there exists a randomized $k$-server algorithm of $O(\log k \log n)$-competitive ratio for any metric space with $n$ points, which improved the previous best competitive ratio $O(\log^2 k \log^3 n \log \log n)$ by Nikhil Bansal et al \cite{3}.

In order to get our results, we use the online primal-dual method, which is developed by Buchbinder and Naor et al. in recent years. Buchbinder and Naor et al. have used the primal-dual method to design online algorithms for many online problems such as covering and packing problems, the ad-auctions problem and so on \cite{4,5,6,7,8}.

First, we propose a primal-dual formulation for the fraction $k$-server problem. Then, we design a $(2 \ln k + 1)$-competitive online algorithm for the fraction $k$-server problem. Based on the known relationship between the fraction $k$-server problem and the randomized $k$-server problem, we get our results.

Above randomized algorithm refer to lazy algorithms. In addition, we still show that there exists a $O(\log k)$-competitive randomized non-lazy algorithm for the $k$-server problem.

2 Preliminaries

In this section, we give some basic definitions. Sleator and Tarjan introduced the concept of competitive analysis in \cite{30}, which is used as a standard way to measure the performance of an online algorithm.

**Definition 2.1.** (Competitive ratio adapted from \cite{31}) For a deterministic online algorithm $DALG$, we call it $r$-competitive if there exists a constant $c$ such that for any request sequence $\rho$, $\text{cost}_{DALG}(\rho) \leq r \cdot \text{cost}_{OPT}(\rho) + c$, where $\text{cost}_{DALG}(\rho)$ and $\text{cost}_{OPT}(\rho)$ are the costs of the online algorithm $DALG$ and the best offline algorithm $OPT$ respectively.

For a randomized online algorithm, we have a similar definition of competitive ratio:

**Definition 2.2.** (Adapted from \cite{31}) For a randomized online algorithm $RALG$, we call it $r$-competitive if there exists a constant $c$ such that for any request sequence $\rho$, $\mathbb{E}[\text{cost}_{RALG}(\rho)] \leq r \cdot \text{cost}_{OPT}(\rho) + c$, where $\mathbb{E}[\text{cost}_{RALG}(\rho)]$ is the expected cost of the randomized online algorithm $RALG$.

Generally, online problems are stated against an adversary who can find the worst sequence for every online algorithm. For randomized problems, there exists three possible adversaries: oblivious
adversaries, adaptive adversaries, fully adaptive adversaries. An oblivious adversary denote that none of our random bits are told to the adversary. In our paper, we focus on the oblivious adversary’s situation.

An online algorithm of the k-sever problem is called lazy if the algorithm move only one sever to serve the requested point only if the requested point is not served, otherwise do nothing. Based on the triangle inequality of the distance, it is easy to know that any non-lazy algorithm can be changed to a lazy algorithm with less cost. Without loss of generality, a randomized algorithm refer to a lazy algorithm in the paper except for special explanation.

For the metric space S, a possible position of the k servers is called configuration. Without loss of generality, we can suppose that all initial positions of k severs are different. For a lazy algorithm, every configuration is a subset of k points in S. For a non-lazy algorithm, every configuration is a multiset of k points in S.

In order to analyze randomized algorithms for the k-sever problem, D. Turkoglu introduce the fractional k-sever problem. On the fractional k-sever problem, severs are viewed as fractional entities as opposed to units and an online algorithm can move fractions of servers to the requested point. In order to introduce the formal definition of the fractional k-sever problem, Turkoglu introduce the definition of a weight function.

Definition 2.3. (Weight function adapted from [31]) At time t, after servicing the t-th request, a weight function Wt : S → R satisfy the following properties

1. Wt(p) ≥ 0 for all p ∈ S.
2. Wt(pt) = 1, where pt is the requested point at time t.
3. The support St of the weight function Wt is finite, where St = {p|Wt(p) > 0, p ∈ S} is the set of points with positive weight.
4. ∑p∈St Wt(p) = k.

In order to accommodate the initial configuration, it is assumed that W0 is integral. Note: from the probability view, the value Wt(p) can be viewed as the probability of having a server at the point p at time t.

Based on the weight function, the fractional k-sever problem is defined as follows.

Definition 2.4. (Fractional k-sever problem adapted from [31]) Suppose that there are a metric space S and a total of k fractional severs located at the points of the metric space as indicated by a weight function W. Given a sequence of requests, each request must be served by providing one unit server at requested point, i.e., the weight function W must be updated accordingly to satisfy the above properties by moving fractional servers to the requested point. The cost of an algorithm for servicing a sequence of requests is the cumulative weighted sum of the cost incurred by each sever, where moving a w fraction of a server for a distance of δ costs wδ.

In [10, 11], Bartal introduce the definition of a Hierarchical well-separated tree (HST), into which a general metric can be embedded with a probability distribution. The internal nodes of the HST represent clusters and its leaves correspond to the points of the metric space. For any internal node, the distance from it to its parent node is α times of the distance from it to its child node. The number α is called the stretch of the HST. An HST with stretch α is called a α-HST. Fakcharoenphol et al. showed the following result [21].

Lemma 2.5. If there is a γ-competitive randomized algorithm for the k-sever problem on an α-HST with all requests at the n leaves, then there exists an O(γα log n)-competitive randomized online algorithm for the k-server problem on any metric space with n points.
3 A Fractional Primal-Dual Algorithm and Randomized Algorithms

In this section, we fist give the LP formulation and its dual formulation for the fractional $k$-sever problem. Then, we design a $(2 \ln k + 1)$-competitive algorithm via the primal-dual method. Finally, the fractional algorithm is transformed to randomized algorithms for different cases.

We suppose that the metric space $S$ has $n$ points $\{1, \ldots, n\}$ and $d(x, y)$ is the distance between $x$ and $y$ for any two points $x, y \in \{1, \ldots, n\}$. Let $p_1, p_2, \ldots, p_T$ be the requested points sequence until time $T$, where $p_t$ is the request point at time $t$. Since a point must be severed by one unit sever when it is requested, we can assume that each point is of one unit space. The variable $u_{p,t}$ denote the fraction of one unit space in the point $p$ at time $t$, i.e., $u_{p,t} = 1 - W^t(p)$, where $W^t$ is the weight function at time $t$ (See Definition 2.3). Then, at time $t$, $\max\{0, u_{p,t} - u_{p,t-1}\}$ denote the moved fractional mass of severs from point $p$ to point $p_t$. Thus, its moved distance is $d(p, p_t)(u_{p,t} - u_{p,t-1})$. Hence, we introduce a new variable $z_{p,t} = \max\{0, u_{p,t} - u_{p,t-1}\}$. Suppose that at time 0, the set of $k$-sever’s initial positions is $I = \{p_1, \ldots, p_k\}$. Thus, we can build a LP formulation for the fractional k-sever problem as follows.

\[
(P) \quad \text{Minimize } \sum_{t=1}^{T} \sum_{p=1}^{n} d(p, p_t) z_{p,t} + \sum_{t=1}^{T} \infty \cdot u_{p,t} \\
\text{Subject to } \forall t > 0 \text{ and } S \subseteq [n] \text{ with } |S| > k, \sum_{p \in S} u_{p,t} \geq |S| - k; \quad (3.1) \\
\forall t > 0 \text{ and point } p, z_{p,t} \geq u_{p,t} - u_{p,t-1}; \quad (3.2) \\
\forall t > 0 \text{ and point } p, z_{p,t}, u_{p,t} \geq 0; \quad (3.3) \\
\text{For } t = 0 \text{ and any point } p \in I, u_{p,0} = 0; \quad (3.4) \\
\text{For } t = 0 \text{ and any point } p \notin I, u_{p,0} = 1; \quad (3.5)
\]

The first primal constraint (3.1) states that at any time $t$, if we take any set $S$ of vertices with $|S| > k$, then $\sum_{p \in S} u_{p,t} = |S| - \sum_{x \in S \setminus S} W^t(x) \geq |S| - \sum_{x \in S} W^t(x) = |S| - k$, i.e., the total spaces of vertices that has not been severed by fractional severs is at least $|S| - k$. The variables $z_{p,t}$ denote the fraction mass of fractional severs on the point $p$ that are moved at time $t$. The fourth and fifth constraints ((3.4) and (3.5)) enforce the initial positions of $k$ severs are $p_1, \ldots, p_k$. The first term in the object function is the sum of the moved distance and the second term enforces the requirement that point $p_t$ must be served by a sever at time $t$ (i.e., $u_{p_t,t} = 0$).

Note: essentially, above primal formulation is that the state space method of Artificial Intelligence is used to denote the fractional $k$-sever problem. We believe that this method can be used to denote many online problems.

Its dual formulation is as follows.

\[
(D) \quad \text{Maximize } \sum_{t=1}^{T} \sum_{S \subseteq [n], |S| > k} (|S| - k) a_{S,t} + \sum_{p \notin I} \gamma_p \\
\text{Subject to } \forall t \text{ and } p \neq p_t, \sum_{S \subseteq [n]} a_{S,t} - b_{p,t} + b_{p,t+1} \leq 0 \\
\forall t = 0 \text{ and } \forall p, b_{p,1} + \gamma_p \leq 0 \\
\forall t > 0 \text{ and } p, b_{p,t} \leq d(p, p_t) \\
\forall t > 0 \text{ and } p \text{ and } |S| > k, a_{S,t}, b_{p,t} \geq 0.
\]

In the dual formulation, the variable $a_{S,t}$ corresponds to the constraint of the type (3.1); the variable $b_{p,t}$ corresponds to the constraint of the type (3.2); The variable $\gamma_p$ corresponds to the
constraint of the type (3.4) and (3.5).

Note that our primal-dual formulation belong to the extended primal-dual framework proposed by N.Bansal et al.[9], i.e., constrains have both positive and negative terms.

Based on above primal-dual formulation, we design an online algorithm for the fractional \( k \)-sever problem as follows (see Algorithm 3.1).

1: At time \( t = 0 \), we set \( b_{p,1} = \gamma_p = 0 \) for all \( p \).
2: At time \( t \geq 1 \), when a request \( p_t \) arrives:
3: Initially, we set \( u_{p,t} = u_{p,t-1} \) for all \( p \), and \( b_{p,t+1} \) is initialized to \( b_{p,t} \). Set \( u_{p,t} = b_{p,t} = 0 \).
4: Let \( S = \{ p : u_{p,t} < 1 \} \).
5: If the primal constraint corresponding for set \( S \) is satisfied, then do nothing.
6: Otherwise, do the following:
7: While \( \sum_{p \in S} u_{p,t} < |S| - k \):
8: For all vertices \( p \in S \setminus \{ p_t \} \),
9: Increasing \( a_{S,t} \) and \( b_{p,t} \) at the same rate.
10: Increase \( u_{p,t} \) by the following function:
   \[ u_{p,t} \leftarrow \frac{1}{k} \cdot \exp \left( \frac{b_{p,t}}{d(p,p_t)} \cdot \ln(1 + k) \right) - 1. \]
11: If some \( u_{p,t} \) reaches the value of 1, then we undate \( S \leftarrow S \setminus \{ p \} \).

**Algorithm 3.1:** The online primal-dual algorithm for the fractional \( k \)-sever problem.

**Theorem 3.1.** The online algorithm for the fractional \( k \)-sever problem is of competitive ratio \( 2(1 + \ln k) \).

**Proof.** Let \( P \) denote the value of the objective function of the primal solution and \( D \) denote the value of the objective function of the dual solution. Initially, let \( P = 0 \) and \( D = 0 \). In the following, we prove three claims:

1. The primal solution produced by the algorithm is feasible.
2. The dual solution produced by the algorithm is feasible.
3. \( P \leq 2 \ln(1 + k) D \).

By three claims and weak duality of linear programs, the theorem follows immediately.

First, we prove the claim (1) as follows. For the particular set \( S = \{ p : u_{p,t} < 1 \} \), When the While iteration (line 7 of above algorithm) stop, \( \sum_{p \in S} u_{p,t} \geq |S| - k \), i.e. the primal constraint for set \( S \) is satisfied. Then we show that the primal constraints for all sets are satisfied. The following two cases are considered.

The first case: for any \( S' \), if \( S' \subseteq S \), then \( \sum_{p \in S'} u_{p,t} \geq |S'| - k \). The reason is as follows.

Since \( \sum_{p \in S'} u_{p,t} + \sum_{p \in S \setminus S'} u_{p,t} = \sum_{p \in S} u_{p,t} \geq |S| - k \), we get:

\[
\sum_{p \in S'} u_{p,t} \geq |S| - k - \sum_{p \in S \setminus S'} u_{p,t} \\
\geq |S| - k - \sum_{p \in S \setminus S'} 1 \\
= |S| - k - |S' \setminus S| \\
= |S'| - k.
\]

The second case: for any \( S'' \) with \( |S''| > k \), then \( \sum_{p \in S''} u_{p,t} \geq |S''| - k \). The reason is as follows.

Let \( S' = S'' \cap S \). Then,
\[
\begin{align*}
\sum_{p \in S'} u_{p,t} & = \sum_{p \in S'} u_{p,t} + \sum_{p \in S'' \setminus S'} u_{p,t} \\
& \geq |S'| - k + |S'' \setminus S'| \quad \text{(By } S' \subseteq S \text{ and the first case and when } p \notin S, u_{p,t} = 1) \\
& = |S''| - k.
\end{align*}
\]

Second, we prove the claim (2) as follows. For any point \( p \), \( b_{p,t} \leq d(p, p_t) \) holds. The reason is: when \( b_{p,t} = d(p, p_t) \), \( u_{p,t} = 1 \) by \( u_{p,t} \leftarrow \frac{1}{k} \cdot (\exp(\frac{b_{p,t}}{d(p, p_t)}) \cdot \ln(1 + k)) - 1 \). Then \( S \leftarrow S \setminus \{p\} \), i.e. \( p \) is removed from \( S \).

Finally, since the increasing ratio of \( a_{S,t} \) and \( b_{p,t} \) is the same for all \( p \in S \setminus \{p_t\} \), the dual constraint is satisfied.

Third, we prove claim (3) as follows. If the algorithm increases the variables \( a_{S,t} \) at some time \( t \), then: \( \frac{\partial D}{\partial a_{S,t}} = |S| - k \).

In the primal formulation, the constraint (3.1) can enforce \( z_{p,t} = u_{p,t} - u_{p,t-1} \) for the optimal solution. Hence, we can always set \( z_{p,t} = u_{p,t} - u_{p,t-1} \) in our algorithm. Thus, in the primal cost, it is easy to compute: \( \frac{dz_{p,t}}{db_{p,t}} = \frac{du_{p,t}}{db_{p,t}} = \frac{\ln(1+k)}{d(p, p_t)} (u_{p,t} + 1) \).

Since the increasing ratio of \( a_{S,t} \) and \( b_{p,t} \) is the same, the derivative of the cost in the primal is as follows.

\[
\frac{\partial P}{\partial a_{S,t}} = \frac{\partial P}{\partial b_{p,t}} = \sum_{p \in S \setminus \{p_t\}} d(p, p_t) \cdot \frac{du_{p,t}}{db_{p,t}} = \sum_{p \in S \setminus \{p_t\}} \ln(1+k)(u_{p,t} + 1) \leq \ln(1+k)((|S| - k) + \frac{|S| - 1}{k}) \leq 2\ln(1+k) \cdot (|S| - k). \quad \text{(When } |S| \geq k + 1, \frac{|S| - 1}{k} \leq |S| - k) \\
= 2\ln(1+k) \cdot \frac{\partial D}{\partial a_{S,t}}.
\]

Where the first inequality holds since \( \sum_{p \in S} u_{p,t} < |S| - k \), the reason is that the constraint at time \( t \) is not satisfied otherwise the algorithm stop increasing the variable \( u_{p,t} \).

Thus, we get \( P \leq 2\ln(1+k)D \).

Let \( OPT \) be the cost of the best offline algorithm. \( P_{\min} \) be the optimal primal solution and \( D_{\max} \) be the optimal dual solution. Then, \( P_{\min} \leq OPT \) since \( OPT \) is a feasible solution for the primal program. Based on the weak duality, \( D_{\max} \leq P_{\min} \). Hence, \( \frac{P}{OPT} \leq \frac{P}{P_{\min}} \leq \frac{2\ln(1+k)D}{P_{\min}} \leq 2\ln(1+k). \)

So, the competitive ratio of this algorithm is \( 2\ln(1+k) \).

\[\square\]

In [31], Duru Türkoğlu study the relationship between fractional version and randomized version of the \( k \)-sever problem, which is given as follows.

**Lemma 3.2.** The fractional \( k \)-sever problem is equivalent to the randomized \( k \)-sever problem on the line or circle, or if \( k = 2 \) or \( k = n - 1 \) for arbitrary metric spaces.

Thus, we get the following conclusion by Theorem 3.1:

**Theorem 3.3.** There is a randomized algorithm with competitive ratio \( 2(1 + \ln k) \) for the \( k \)-sever problem on the line or circle or if \( k = 2 \) or \( k = n - 1 \) for arbitrary metric spaces.

In [3], Nikhil Bansal et al. design a \( O(\log^2 k \log^3 n \log \log n) \)-competitive randomized algorithm for the \( k \)-sever problem, which is a combinatorial algorithm. Nikhil Bansal et al. show the following conclusion.
Lemma 3.4. Let \( T \) be a \( \sigma \)-HST with \( \sigma > 5 \). Then any online fractional \( k \)-sever algorithm on \( T \) can be converted into a randomized \( k \)-sever algorithm on \( T \) with an \( O(1) \) factor loss in the competitive ratio.

Thus, we get the following conclusion by Theorem 3.1:

Theorem 3.5. Let \( T \) be a \( \sigma \)-HST with \( \sigma > 5 \). There is a randomized algorithm for the \( k \)-sever problem with a competitive ratio of \( O(\log k) \) on \( T \).

Thus, we get the following conclusion by Lemma 2.5:

Theorem 3.6. For any metric space, there is a randomized algorithm for the \( k \)-sever problem with a competitive ratio of \( O(\log k \log n) \).

Note that previous randomized algorithm refer to randomized lazy algorithm. In the following, we consider randomized non-lazy algorithm for the \( k \)-sever problem. We now show that an randomized non-lazy algorithm can be obtained from a fractional \( k \)-sever algorithm on any metric spaces with the same competitive ratio. The reduction use the ideas in \([2, 3]\) which are developed for the weighted caching and the \( k \)-sever problem on HST.

At time \( t \), let the state \( P_t \) of a randomized non-lazy algorithm be the probability distribution on the configurations which can be multi-sets of \( k \) points in \( S \). For a configuration \( C \) at time \( t \), let \( Pr_t(C) \) denote its probability. First, we show the following conclusion: any fractional \( k \)-sever algorithm on arbitrary metric space can be simulated by a randomized non-lazy algorithm at the same cost.

Lemma 3.7. For a fractional \( k \)-sever algorithm, assume that \( W^0, W^1, \ldots \) be its a sequence of weight function states. Then there is an online procedure that maintains a sequence of randomized non-lazy \( k \)-sever states \( P^0, P^1, \ldots \) satisfying the following properties:

1. At any time \( t \), the state \( P^t \) is consistent with the fractional state \( W^t \), i.e., for every \( p \), \( \sum_{C: p \in C} Pr_t(C) = W^t(p) \).

2. At time \( t \), if the fractional state changes from \( W^{t-1} \) to \( W^t \) at time \( t \), incurring a movement cost of \( c_t \), then the state \( P^{t-1} \) can be modified to a state \( P^t \) while incurring the same cost.

Proof. Given a fractional \( k \)-sever algorithm \( ALG \), the shift of some fractional severs for one point to another point, we call it a basic movement. Obviously, the basic movement span the set of all possible movements. Hence, if we simulate about the basic movement at the same cost, we can simulate all movements at the same cost.

Thus, for a basic movement, when \( \epsilon \) fraction severs of \( p_i \) points is moved to the requested point \( p_t \) at the time \( t \), the moving cost is \( ed(p_i, p_t) \).

At this time, from \( P^{t-1} \), we choose \( \epsilon \) measure of configurations \( C \) that contain \( p_i \), and replace \( p_i \) with \( p_t \). We call these changed configurations \( C' \). Let \( Pr_t(C') = \epsilon \) and \( Pr_t(C) = Pr_{t-1}(C) - \epsilon \).

Then, the probability distribute \( P^t \) is still consistent with the weigh function \( W^t \). The cost of shift is also \( ed(p_i, p_t) \), which is the same cost produced by the fractional algorithm. Note that it is possible the new configurations become the multi-sets. So, it is a randomized non-lazy algorithm with the same cost.

From the Theorem 3.1, we get:
Theorem 3.8. There is a $O(\log k)$-competitive randomized non-lazy algorithm for the $k$-sever problem.

In order to settle the randomized $k$-sever conjecture completely, we propose the following conjectures.

Conjecture 1: Any online fractional $k$-sever algorithm on any metric space can be converted into a randomized $k$-sever algorithm with an $O(1)$ factor loss in the competitive ratio.

Conjecture 2: Any randomized non-lazy algorithm can be changed to a randomized lazy algorithm with less cost.

Obviously, any one of above two conjectures implies the randomized $k$-sever conjecture.

4 Conclusion

In this paper, for the following metric spaces: line, circle, HST, if $k = 2$ or $n = k + 1$ for arbitrary metric spaces, we settle the randomized $k$-sever conjecture. Specially, we show that there exists randomized algorithms with $O(\log k)$-competitive ratio for the $k$-sever problem on above metric spaces. For any metric space with $n$ points, we show that there exist a randomized algorithm with $O(\log k \log n)$-competitive ratio for the $k$-sever problem, which improved the previous best competitive ratio $O(\log^2 k \log^3 n \log \log n)$. In addition, we still show that there is a randomized non-lazy algorithm with competitive ratio $O(\log k)$ for the $k$-sever problem.

Acknowledgments

We would like to thank the anonymous referees for their careful readings of the manuscripts and many useful suggestions.

Wenbin Chen’s research has been partly supported by the National Natural Science Foundation of China (NSFC) under Grant No.11271097., the research projects of Guangzhou education bureau under Grant No. 2012A074. and the project KFKT2012B01 from State Key Laboratory for Novel Software Technology, Nanjing University.

References

[1] Dimitris Achlioptas, Marek Chrobak, and John Noga. Competitive analysis of randomized paging algorithms. Theoretical Computer Science, 234(1-2):203-218, 2000.

[2] Avrim Blum, Carl Burch, and Adam Kalai. Finely-competitive paging. Proceedings of the 40th Annual Symposium on Foundations of Computer Science, page 450-456, 1999.

[3] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, Joseph Naor. A Polylogarithmic-Competitive Algorithm for the $k$-Server Problem. FOCS 2011, pages 267-276.

[4] Nikhil Bansal, Niv Buchbinder, and Joseph (Seffi) Naor. A primal-dual randomized algorithm for weighted paging. Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, pages 507-517, 2007.

[5] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. Proc. 14th European Symp. on Algorithms (ESA), pp. 253-264, 2007.
[6] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. Proc. 12th European Symp. on Algorithms (ESA), volume 3669 of Lecture Notes in Comput. Sci., pages 689–701. Springer, 2005.

[7] N. Buchbinder and J. Naor. Improved bounds for online routing and packing via a primal-dual approach. Proc. 47th Symp. Foundations of Computer Science, pages 293–304, 2006.

[8] Niv Buchbinder, Joseph Naor. The Design of Competitive Online Algorithms via a Primal-Dual Approach. Foundations and Trends in Theoretical Computer Science 3(2-3): 93–263 (2009).

[9] Nikhil Bansal, Niv Buchbinder, and Joseph (Seffi) Naor. Towards the randomized k-server conjecture: A primal-dual approach. Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 40-55, 2010.

[10] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, pages 184-193, 1996.

[11] Yair Bartal. On approximating arbitrary metrics by tree metrics. Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pages 161-168, 1998.

[12] Yair Bartal, Bela Bollobas, and Manor Mendel. A ramsey-type theorem for metric spaces and its applications for metrical task systems and related problems. Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, pages 396-405, 2001.

[13] Yair Bartal and Eddie Grove. The harmonic k-server algorithm is competitive. Journal of the ACM, 47(1):1-15, 2000.

[14] Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric ramsey-type phenomena. Proceedings of the 35th Annual ACM Symposium on Theory of Computing, pages 463C472, 2003.

[15] Yair Bartal, Elias Koutsoupias. On the competitive ratio of the work function algorithm for the k-sever problem. Theoretical Computer Science, 324(2-3): 337-345.

[16] Avrim Blum, Howard J. Karloff, Yuval Rabani, and Michael E. Saks. A decomposition theorem and bounds for randomized server problems. Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science, pages 197-207, 1992.

[17] Allan Borodin and Ran El-Yaniv. Online computation and competitive analysis. Cambridge University Press, 1998.

[18] M. Chrobak and L. Larmore. An optimal on-line algorithm for k-servers on trees. SIAM Journal on Computing, 20(1): 144-148, 1991.

[19] A. Coté, A. Meyerson, and L. Poplawski. Randomized k-server on hierarchical binary trees. Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pages 227-234, 2008.

[20] B. Csaba and S. Lodha. A randomized on-line algorithm for the k-server problem on a line. Random Structures and Algorithms, 29(1): 82-104, 2006.
[21] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. Proceedings of the 35th Annual ACM Symposium on Theory of Computing, pages 448-455, 2003.

[22] A. Fiat, Y. Rabani, and Y. Ravid. Competitive $k$-server algorithms. Journal of Computer and System Sciences, 48(3): 410-428, 1994.

[23] Amos Fiat, Richard M. Karp, Michael Luby, Lyle A. McGeoch, Daniel Dominic Sleator, and Neal E. Young. Competitive paging algorithms. Journal of Algorithms, 12(4): 685-699, 1991.

[24] Edward F. Grove. The harmonic online $k$-server algorithm is competitive. Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, pages 260-266, 1991.

[25] Elias Koutsoupias. The $k$-server problem. Computer Science Review. Vol. 3. No. 2. Pages 105-118, 2009.

[26] Elias Koutsoupias and Christos H. Papadimitriou. On the $k$-server conjecture. Journal of the ACM, 42(5): 971-983, 1995.

[27] M.S. Manasse, L.A. McGeoch, and D.D. Sleator. Competitive algorithms for online problems. Proceedings of the 20th Annual ACM Symposium on Theory of Computing, pages 322-333, 1988.

[28] M. Manasse, L.A. McGeoch, and D. Sleator. Competitive algorithms for server problems. Journal of Algorithms, 11: 208-230, 1990.

[29] Lyle A. McGeoch and Daniel D. Sleator. A strongly competitive randomized paging algorithm. Algorithmica, 6(6): 816-825, 1991.

[30] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. Communications of the ACM, 28(2): 202-208, 1985.

[31] Duru Türkoğlu. The $k$-server problem and fractional analysis. Master’s Thesis, The University of Chicago, 2005. [http://people.cs.uchicago.edu/~duru/papers/masters.pdf](http://people.cs.uchicago.edu/~duru/papers/masters.pdf)