The online k-server problem with rejection[*]

E. Bittner[†] Cs. Imreh [‡] J. Nagy-György[§]

Abstract

In this work we investigate the online k-server problem where each request has a penalty and it is allowed to reject the requests. The goal is to minimize the sum of the total distance moved by the servers and the total penalty of the rejected requests. We extend the work function algorithm to this more general model and prove that it is $(4k-1)$-competitive. We also consider the problem for special cases: we prove that the work function algorithm is 5-competitive if $k=2$ and $(2k+1)$-competitive for any $k \geq 1$ if the metric space is the line. In the case of the line we also present the extension of the double-coverage algorithm and prove that it is $3k$-competitive. This algorithm has worse competitive ratio than the work function algorithm but it is much faster and memoryless. Moreover we prove that for any metric space containing at least $k+1$ points no online algorithm can have smaller competitive ratio than $2k+1$, and this shows that the work

[*]doi:10.1016/j.disopt.2014.03.002 This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency. Cs. Imreh was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences. J. Nagy-György was supported by the TÁMOP 4.2.4.A/2-11-1-2012-0001 National Excellence Program to develop and operate national personal supporting system for students and researchers.

[†]Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary, email: bittner.emese@gmail.com

[‡]Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary, email: cimreh@inf.u-szeged.hu

[§]Department of Mathematics, University of Szeged, Aradi Vértanúk tere 2, H-6720 Szeged, Hungary, email: Nagy-Gyorgy@math.u-szeged.hu

function algorithm has the smallest possible competitive ratio in the case of lines and also in the case $k = 2$.

Keywords: Online algorithms, competitive analysis, k-server problems

1 Introduction

The k-server problem can be formulated as follows. In the problem a metric space $\mathcal{M}$ is given with k mobile servers that occupy distinct points of the space and a sequence of requests (points). Each of the requests has to be served, by moving one server from its current position to the requested point. The goal is to minimize the total cost, that is the sum of the distances covered by the k servers. In the online version of the problem the requests arrive one by one and an online k-server algorithm serves each request immediately when it arrives, without any prior knowledge about the future requests. The online k-server problem has applications in planning maintenance service. The model can be also applied to upkeep or design of computer or sensor networks. In many of these applications it is a straightforward idea to allow the servers not to serve some of the requests. In this new model which is called k-server problem with rejection the i-th request is a pair $\varrho_i = (r_i, p_i)$, where r_i is a point of the space and $p_i > 0$ is the *penalty* for the rejection. Each request can be served the same way as in the classical k-server problem, or optionally it also can be rejected at the penalty given along with the request. The cost of an algorithm is the sum of the distances covered by the k servers plus the sum of the penalties of the rejected requests. In this paper we study the online version of the k-server with rejection problem where the requests arrive one by one.

Typically, the quality of an online algorithm is judged using competitive analysis. An online algorithm for a minimization problem is asymptotically c-competitive if its cost is never more than c times the optimal cost plus an additive absolute constant which is independent of the input. Without allowing the additive constant the algorithms are called competitive in the absolute sense. Here we use the asymptotic competitive ratio as it is usually done in the case of the online k-server problem.

Related works: The online k-server problem is one of the most known online problems. The problem is introduced in [15], where the first important results are presented. In [15] it is proved that k is a lower bound for every metric space with at least $k+1$ points, and in [14] the work function algorithm is presented which is $(2k-1)$-competitive for every metric space. The k-server conjecture states that there exists an algorithm that is k-competitive for any metric space. The problem was also investigated for special metric spaces. In the case of uniform space, where the problem is equivalent to the paging problem, a k-competitive algorithm is given in [18]. If the metric space is a line then a k-competitive algorithm is given in [5].

The idea of allowing the algorithm to reject some parts of the input appeared in other online problems as well. The most closely related problem is online paging with rejection, which problem is the k-server problem with rejection for uniform space and studied in [12]. There a $(2k+1)$-competitive algorithm is presented and it is shown that no algorithm with smaller competitive ratio exists. The more general caching with rejection problem is also considered in [12]. A $(2k+1)$-competitive algorithm is given in the bit model and in the cost model and a $(2k+2)$-competitive algorithm is presented for the general caching problem.

The first online model with rejection appeared in online scheduling [2]. In this model the algorithm is allowed to reject the jobs and the objective is to minimize the sum of the makespan of the schedule of accepted jobs and the total penalty of the rejected jobs. After the first paper some further online scheduling models with rejection were investigated. In [17] the problem where it is allowed to preempt the jobs is considered, in [9] and [16] the online scheduling problem with rejection where the algorithm has to purchase the machines is investigated. Online bin packing with rejection, where it is allowed to reject the items and the cost is the sum of the number of used bins and the total penalty of the rejected items is investigated in [8] and [11]. Online graph coloring with rejection is studied in [10].

Our results: We extend the work function algorithm to this more general model and we prove that it is $(4k-1)$-competitive. We also consider the problem in special cases: we prove that the work function algorithm is 5-competitive if $k=2$ and $(2k+1)$-competitive for the line. The work function algorithm is very difficult and has large memory requirements, therefore we also analyse a simpler, memoryless algorithm in the case of the line. This is an extension of the double-coverage algorithm into this more general model, and we show that it is $3k$-competitive. Moreover we prove that for any metric space containing at least $k+1$ points no online algorithm can have a smaller competitive ratio than $2k+1$.

2 Notions and notations

For an arbitrary online algorithm $\mathcal{A}$ and an input sequence or subsequence ϱ the cost of the solution given by $\mathcal{A}$ is denoted by $\mathcal{A}(\varrho)$. Moreover for an input sequence ϱ let $OPT(\varrho)$ denote the cost of the optimal offline solution. Then an online algorithm $\mathcal{A}$ is called c-competitive if there exists a constant b such that $\mathcal{A}(\varrho) \leq c \cdot OPT(\varrho) + b$ for any input sequence ϱ.

We will consider the online k-server problem with rejection problem as the special case of the metrical task system problem defined in [4]. Therefore first we recall the basic definitions and some fundamental results from the area of metrical task systems. In the metrical task system a finite metric space $\mathcal{S}$ of states is given. For notational convenience, if $z, y \in S$, we denote the distance from z to y as zy. In the metrical task system problem we have to execute a sequence of tasks and the cost of executing a task depends on the state where we are. Our goal is to minimize the total cost which is the sum of the execution costs and the state transition costs defined by the distances in $\mathcal{S}$. Thus in a metrical task system problem we have an initial state $x_0 \in S$ and a sequence of tasks $\overline{\tau} = \tau_1, \ldots, \tau_m$ is given. Each task is a function from $\mathcal{S}$ to $R^+ \cup \infty$. We denote by $\bar{\tau}_i$ the prefix of the first i tasks. Any sequence $\overline{x} = x_0, x_1, \ldots, x_m$, where $x_i \in S$, is called a schedule, or a service schedule for x_0, we use $\bar{x}_i$ to denote the prefix of the first $i+1$ elements of the schedule. We define

$$cost(x_0, \overline{\tau}, \overline{x}) = \sum_{i=1}^{m} (x_{i-1}x_i + \tau_i(x_i)).$$

Then the MTS problem is to find a service schedule with minimal cost. We will use $OPT(x_0, \overline{\tau})$ to denote this cost.

In the online MTS problem the tasks arrive one by one, and an online MTS algorithm has to execute each task changing into a selected state without any information about the further tasks. We will use the online work function algorithm which was developed for the solution of the online MTS problem. The work function is defined for every $i = 1, \ldots, m$ for each state $x \in \mathcal{S}$ and this is the minimal cost of serving all requests up to i, and finishing at state x. We can define this function w_i from $\mathcal{S}$ to $R^+ \cup \infty$ for $i = 1, \ldots, m$ as follows:

$$w_i(x) = \min_{\overline{x}_i \in \mathcal{S}} \{cost(x_0, \overline{\tau}_i, \overline{x}_i) + x_i x\}.$$

We also use the work function for $i = 0$, it is defined as $w_0(x) = x_0x$.

We say that w is reachable work function if there exist $x_0 \in S$ and $\tau_1, \dots, \tau_i$ tasks such that $w = w_i$.

We will use the following properties of the work function (see [7] for their proof):

$$w_i(x) \le w_i(y) + xy, \tag{1}$$

$$w_i(x) = \min_{y \in M} \{w_{i-1}(y) + \tau_i(y) + yx\}. \tag{2}$$

The work function algorithm (WFA) keeps track of the work function at each step. If the current state is s_{i-1} WFA chooses the state x that minimizes the value $w_i(x) + s_{i-1}x$, $i = 1, \dots, m$ at the ith step. The first part of the sum minimized by the algorithm is the total service cost ending at the state x (an algorithm which considers only this part is called retrospective), the second part is the simple greedy decision where the goal is to minimize the cost of the state transition.

In the analysis of the work function algorithm usually the extended cost is used. The basic idea is to consider the maximal changes in the value of the work function instead of investigating the actual states. It can be defined as follows.

Definition 1 *Let* $\nabla_i = \sup_x \{w_i(x) - w_{i-1}(x)\}$. *The extended cost is* $\nabla(x_0, \overline{\tau}) = \sum_{i=1}^m \nabla_i$.

We can use this extended cost to bound the competitive ratio of WFA as the following lemma states.

Lemma 2 ([7]) *For each metrical task system, start state* x_0 *and task sequence* $\overline{\tau}$*, the following inequality holds:* $WFA(x_0, \overline{\tau}) \le 2 \cdot \nabla(x_0, \overline{\tau}) - OPT(x_0, \overline{\tau})$.

Now we return to the k-server problem and the k-server problem with rejection. Recall that in the k-server problem we have a metric space $\mathcal{M}$ and k servers. In the online version request arrive one by one, and we have to send at least one server to the point where the request arrive without any information about the further requests. The k-server problem is a special case of the MTS problem, since it can be defined as an MTS problem on the metric space of the server configurations (see [7]). We call the k-element multisets of the metric space configurations, each element of the configuration gives a position of a server. For two configurations A and B of the servers let the distance of these configurations be the minimal total moving distance which is enough to move the servers from A to B. This distance is called the minimum matching distance. The matching between the points in the configuration which results the minimal distance is called the minimal matching between the configurations. It is easy to see that the set of configurations with this distance forms a metric space. The k-server problem can be described as an MTS on this metric space using $\tau_i(X) = 0$ if $r_i \in X$ and $\tau_i(X) = \infty$ if $r_i \notin X$ for any configuration X.

Our first important observation is that the k-server problem with rejection is also an MTS on the space of the configurations of the servers with minimum matching metric. In this case we have $\tau_i(X) = 0$ if $r_i \in X$ and $\tau_i(X) = p_i$ if $r_i \notin X$. We will consider the k-server problem with rejection as this MTS problem. We can apply the result about the general MTS problems, but they usually do not give good bounds for the k-server problem with rejection since they do not use the special structure. One important property is that we can suppose that in each step only one serve moves, therefore we only have to consider the configurations which differs only in one point. We note that it would be possible to define the work function and the work function algorithm directly to the k-server problem with rejection. The work function is the minimal cost of serving a request sequence (also allowing the rejection) and ending at a particular configuration. If the servers are at the configuration X before serving the request r_i then the WFA algorithm either chooses the server s where the function $w_{i-1}(X \setminus \{s\} \cup \{r_i\}) + sr_i$ is minimal to serve the request or rejects it if $w_{i-1}(x) + p_i$ is smaller than this minimum. But we will use some general result on the work function algorithm from the area of MTS therefore we decided to handle the k-server problem with rejection as an MTS while we present the results about WFA.

In the rest of the paper we use capital letters for the configurations and small letters for the points from the metric space of the servers. For a pair a, B of a point and a configuration $\sum_{b \in B} ab$ is denoted by aB. Furthermore for a multiset B and points a,b we will use $A - a + b$ to denote $A \setminus \{a\} \cup \{b\}$

We suppose that the starting configuration contains different points. Then the work function algorithm moves only one server to the request point if it has no server there, thus it is always in such configurations where all the points are different. By this observation it follows that one could define the configurations as sets instead of multisets. As in many papers on the k-server problem ([3],[7],[13], [14]) we allow multisets for the sake of simpler notations and proofs. Using multisets we can avoid the case disjunction when we use formulas like $X - a + b$.

On the other hand using only the configurations where the servers occupy different points of the space can be useful as our first observation on the WFA for the k-server with rejection problem shows.

Proposition 3 *WFA is* $(2k+1)$*-competitive for the* k*-server problem with rejection on the metric spaces containing exactly* $k+1$ *points.*

Proof. As we noted above we can suppose that the servers are at different points, but this yields that they can form only $k+1$ different configurations. If we consider the equivalent MTS problem problem then its metric space contains $k+1$ points. On the other hand it is known that WFA is $(2n-1)$-competitive for an n-state MTS (see [7]). We can apply this result and we obtain immediately that WFA is $(2k+1)$-competitive for the k-server problem with rejection in the case of these special metric spaces. □

We defined the MTS problem only on finite metric spaces. On the other hand the k-server problem is also investigated in infinite spaces. Later in this paper we will study the behavior of WFA on the line. The algorithm can be used on infinite metric spaces as follows. We can observe as it is done in [6] and [13] that the work function can obtain the minimal value only on the configurations which contain such points which are in the initial configuration or requested in the input. Therefore we can restrict the infinite space of configurations in each step to this finite subspace and we can use the algorithm as it is described above.

3 General metric spaces

Koutsoupias and Papadimitriou [14] showed that WFA is $(2k-1)$-competitive for the k-server problem. Extending their ideas into this more general model we can prove the following theorem.

Theorem 4 *WFA is* $(4k-1)$*-competitive for the* k*-server problem with rejection.*

Consider an initial configuration X_0 and a request sequence $(r_1, p_1), \ldots, (r_m, p_m)$. The following property of the work function plays crucial role in our proof.

Proposition 5 *For an arbitrary configuration X and $1 \le i \le m$*

$$w_i(X) = \min\{w_{i-1}(X) + p_i,\ \min_{x\in X}\{w_{i-1}(X - x + r_i) + r_i x\}\}.$$

Proof. Considering the right side of the equality (2) we obtain that $w_i(X) \le \min\{w_{i-1}(X) + p_i,\ \min_{x\in X}\{w_{i-1}(X - x + r_i) + r_i x\}\}$, since here we take the minimum on a smaller set. On the other hand, again by property (2) we can suppose that $w_i(X) = w_{i-1}(Y) + \tau_i(Y) + YX$ for some Y. If $r_i \notin Y$ then by property (1) we obtain that $w_i(X) = w_{i-1}(Y) + p_i + YX \ge w_{i-1}(X) + p_i$. If $r_i \in Y$ then let x be the point which is matched to r_i in the minimal matching between X and Y. Then $XY = (X - x)(Y - r_i) + xr_i = (X - x + r_i)Y + xr_i$. Therefore $w_i(X) = w_{i-1}(Y) + (X - x + r_i)Y + xr_i \ge w_{i-1}(X - x + r_i) + xr_i$ by property (1). Note that the proposition is also valid in the case when $r_i \in X$, in this special case the minimum is attained at $x = r_i$ and $w_i(X) = w_{i-1}(X)$. $\square$

To analyse the algorithm we need the property of quasiconvexity. As it is stated in [14] the basic idea behind the quasiconvexity is that it ensures that optimal solutions can be transformed into each other by sequences of swaps.

Definition 6 *The work function w is quasiconvex, if for all configurations X, Y there is a bijection $f : X \to Y$ with the following properties:*
(*) $f(X \cap Y) = X \cap Y$,
(**) if $X = A \cup B$ and $A \cap B = \emptyset$ then

$$w(X) + w(Y) \ge w(A \cup f(B)) + w(f(A) \cup B).$$

Lemma 7 *If w is reachable work function then w is quasiconvex.*

Proof. If w is a work function and g satisfies (**) then we can transform g into some f satisfying (*) and (**). Suppose that there is $u \in X \cap Y$ such that $g(u) \ne u$. Define $g'(u) = u$, $g'(g^{-1}(u)) = g(u)$ and $g'(z) = g(z)$ for $z \in X - \{u, g^{-1}(u)\}$. Without loss of generality we can assume that $g^{-1}(u) \in A$. If $u \in A$ then $g(A) = g'(A)$, therefore $w(X) + w(Y) \ge w(A \cup g'(B)) + w(g'(A) \cup B)$. If $u \notin A$ then using (**) for the sets $A + u$, $B - u$ we obtain that

$$\begin{aligned} w(X) + w(Y) &\ge w((A + u) \cup g(B - u)) + w(g(A + u) \cup (B - u)) \\ &= w(A \cup g'(B)) + w(g'(A) \cup B) \end{aligned}$$

and g' has at least one additional fixed point in $X \cap Y$ to g. By repeating this process, we obtain a quasiconvex f.

We prove that a function satisfying (**) exists by induction on the number of the requests. Let $w = w_0$ and X, Y be two configurations and $h_X : X \to X_0$, $h_Y : Y \to X_0$ be the minimum matching bijections (note that X_0 is the starting configuration). Now define the following function form X to Y: $g_0 = h_Y^{-1} \circ h_X$. Let $X = A \cup B$ and consider the sum $w_0(A \cup g_0(B)) + w_0(g_0(A) \cup B)$. The first part is the cost of the minimal matching between X_0 and $A \cup g_0(B)$ and the second is the cost of the minimal matching between X_0 and $g_0(A) \cup B$. On the other hand $w_0(X) + w_0(Y)$ is the sum of the costs of the minimal matchings between the configurations X, X_0 and Y, X_0. Denote these matchings by M_1 and M_2. Then the edges from M_1 can be used as edges between X_0 and B and between X_0 and A. Since g_0 is a bijection we obtain that $g_0(A) \cup g_0(B) = Y$, and the edges in M_2 can be used for $g_0(A)$ and $g_0(B)$. By the definition of g_0 it follows that the edges which leads to the elements of A in M_1 and the edges which lead to the element of $g_0(B)$ in M_2 have different endpoints in X_0. This yields that using the edges in M_1 and M_2 we also can obtain two matchings between the configurations $X_0, A \cup g_0(B)$ and $X_0, g_0(A) \cup B$. And this proves that $w_0(X) + w_0(Y) \geq w_0((A \cup g_0(B)) + w_0(g_0(A) \cup B)$. Therefore we obtained that g_0 satisfies (**), so there is an f_0 satisfying (*) and (**).

Let us now assume that w_{i-1} is quasiconvex. Let X and Y be arbitrary configurations and choose an arbitrary $A \subseteq X$ and let $B = X - A$. By Proposition 5 we get 3 cases.

Case 1. Suppose that $w_i(X) = w_{i-1}(X) + p_i$ and $w_i(Y) = w_{i-1}(Y) + p_i$. By induction, for w_{i-1} there is a bijection $f_{i-1} : X \to Y$ that satisfies (*) and (**). Set $f_i = f_{i-1}$. Then

$$
\begin{aligned}
w_i(X) + w_i(Y) &= w_{i-1}(X) + w_{i-1}(Y) + 2p_i \\
&\geq w_{i-1}(A \cup f_{i-1}(B)) + w_{i-1}(f_{i-1}(A) \cup B) + 2p_i \\
&= w_{i-1}(A \cup f_i(B)) + w_{i-1}(f_i(A) \cup B) + 2p_i \\
&\geq w_i(A \cup f_i(B)) + w_i(f_i(A) \cup B),
\end{aligned}
$$

where first we used the quasiconvexity of w_{i-1} for configurations X, Y and sets A, B and the last inequality follows from Proposition 5.

Case 2. Suppose that $w_i(X) = w_{i-1}(X - x + r_i) + r_i x$ and $w_i(Y) = w_{i-1}(Y) + p_i$. Without loss of generality we can assume that $x \in A$. By induction, for w_{i-1} there is a bijection $f_{i-1} : (X - x + r_i) \to Y$ that satisfies

(*) and (**). Set $g_i(x) = f_{i-1}(r_i)$ and $g_i(z) = f_{i-1}(z)$ if $z \neq x$. Then

$$\begin{aligned} w_i(X) + w_i(Y) &= w_{i-1}(X - x + r_i) + r_i x + w_{i-1}(Y) + p_i \\ &\geq w_{i-1}((A - x + r_i) \cup f_{i-1}(B)) + r_i x + \\ &\quad w_{i-1}(f_{i-1}(A - x + r_i) \cup B) + p_i \\ &= w_{i-1}((A - x + r_i) \cup g_i(B)) + r_i x + \\ &\quad w_{i-1}(g_i(A) \cup B) + p_i \\ &\geq w_i(A \cup g_i(B)) + w_i(g_i(A) \cup B), \end{aligned}$$

where first we used the quasiconvexity of w_{i-1} for configurations $X - x + r_i, Y$ and sets $A - x + r_i, B$, and the last inequality follows from Proposition 5.
Case 3. Suppose that $w_i(X) = w_{i-1}(X - x + r_i) + r_i x$ and $w_i(Y) = w_{i-1}(Y - y + r_i) + r_i y$. Without loss of generality we can assume that $x \in A$. By induction, for w_{i-1} there is a bijection $f_{i-1} : (X - x + r_i) \to (Y - y + r_i)$ that satisfies (*) and (**). Set $g_i(x) = y$ and $g_i(z) = f_{i-1}(z)$ if $z \neq x$. Then

$$\begin{aligned} w_i(X) + w_i(Y) &= w_{i-1}(X - x + r_i) + r_i x + w_{i-1}(Y - y + r_i) + r_i y \\ &\geq w_{i-1}((A - x + r_i) \cup f_{i-1}(B)) + r_i x + \\ &\quad w_{i-1}(f_{i-1}(A - x + r_i) \cup B) + r_i y \\ &= w_{i-1}((A - x + r_i) \cup g_i(B)) + r_i x + \\ &\quad w_{i-1}((g_i(A) - y + r_i) \cup B) + r_i y \\ &\geq w_i(A \cup g_i(B)) + w_i(g_i(A) \cup B). \end{aligned}$$

where we used the quasiconvexity of w_{i-1} for configurations $X - x + r_i, Y - y + r_i$ and sets $A - x + r_i, B$

Therefore there is an f_i satisfying (*) and (**). □

Next we define the (w, x)-minimizer configurations. These configurations will play important role in the proof. We will show later that the extended cost can be assigned to a minimizer.

Definition 8 *A configuration X is called (w, x)-minimizer if*

$$w(X) - \sum_{y \in X} yx = \min_Y \{w(Y) - \sum_{y \in Y} yx\}.$$

Lemma 9 *There exists a (w_i, x)-minimizer X such that $w_i(X) = w_{i-1}(X) + \tau_i(X)$.*

Proof. Suppose that Y is (w_i, x)-minimizer. Then we have a configuration X such that $w_i(Y) = w_{i-1}(X)+\tau_i(X)+XY$. Considering one such configuration we obtain that

$$\begin{aligned} w_i(Y) - \sum_{y\in Y} yx &= w_{i-1}(X) + \tau_i(X) + XY - \sum_{y\in Y} yx \\ &\geq w_{i-1}(X) + \tau_i(X) - \sum_{y\in X} yx \\ &\geq w_i(X) - \sum_{y\in X} yx, \end{aligned}$$

first we use $XY - \sum_{y\in Y} yx \geq -\sum_{y\in X} yx$ by the triangle inequality and then we use Proposition 5. On the other hand, since Y is a (w_i, x)-minimizer, equality holds in all places above, so $w_i(X) = w_{i-1}(X) + \tau_i(X)$ holds too. □

Now we are ready to prove the most important property of the (w_i, x)-minimizers. This lemma is usually called duality lemma since it connects a maximum property (maximal extended cost) into a minimal property of minimizers. It shows that the extended cost of serving the request r_i occurs at the (w_i, x)-minimizers. We can state this lemma as follows.

Lemma 10 *For every (w_{i-1}, r_i)-minimizer X we have that*

(a) X is also a (w_i, r_i)-minimizer,

(b) $\nabla_i = w_i(X) - w_{i-1}(X)$.

Proof. (a) Let X be a (w_{i-1}, r_i)-minimizer and Y be such a (w_i, r_i)-minimizer which satisfies Lemma 9.
If $r_i \in Y$ then $w_i(Y) = w_{i-1}(Y)$ by Lemma 9. Furthermore using Lemma 7 for the configurations Y, X we obtain that there is a bijection f from Y to X which satisfies (*) and (**). Let $x = f(r_i)$. Then we can use (**) to the sets $\{r_i\}, Y - r_i$ and by $f(r_i) = x$ we obtain that $w_{i-1}(Y) + w_{i-1}(X) \geq w_{i-1}(Y - r_i + x) + w_{i-1}(X - x + r_i)$ holds. Then we have

$$
\begin{aligned}
w_i(Y) - \sum_{y\in Y} yr_i &= w_{i-1}(Y) - \sum_{y\in Y} yr_i \\
&\geq w_{i-1}(Y - r_i + x) + w_{i-1}(X - x + r_i) - w_{i-1}(X) - \sum_{y\in Y} yr_i \\
&= w_{i-1}(Y - r_i + x) - \sum_{y\in Y - r_i + x} yr_i + \\
&\quad xr_i + w_{i-1}(X - x + r_i) - w_{i-1}(X) \\
&\geq w_{i-1}(X) - \sum_{y\in X} yr_i + w_i(X) - w_{i-1}(X) \\
&= w_i(X) - \sum_{y\in X} yr_i,
\end{aligned}
$$

where the second inequality holds since X is a (w_{i-1}, r_i)-minimizer and by Proposition 5. And this proves that X is also an (w_i, r_i)-minimizer in this case.
If $r_i \notin Y$ then $w_i(Y) = w_{i-1}(Y) + p_i$ by Lemma 9. So

$$
\begin{aligned}
w_i(Y) - \sum_{y\in Y} yr_i &= w_{i-1}(Y) - \sum_{y\in Y} yr_i + p_i \\
&\geq w_{i-1}(X) - \sum_{y\in X} yr_i + p_i \\
&\geq w_i(X) - \sum_{y\in X} yr_i,
\end{aligned}
$$

where the first inequality holds since X is a (w_{i-1}, r_i)-minimizer and the second inequality follows from Proposition 5. And this proves that X is also a (w_i, r_i)-minimizer in this case.

(b) Let X be a (w_{i-1}, r_i)-minimizer and Y an arbitrary configuration. If $w_i(X) < w_{i-1}(X) + p_i$ then there is an $x \in X$ such that $w_i(X) = w_{i-1}(X - x + r_i) + r_i x$. Furthermore using Lemma 7 for the configurations $X - x + r_i, Y$ we obtain that there exits a bijection f from $X - x + r_i$ to Y which satisfies (*) and (**). Let $y = f(r_i)$ and use (**) for the sets $\{r_i\}, X - x$. Then we obtain that

$$
w_{i-1}(X - x + r_i) + w_{i-1}(Y) \geq w_{i-1}(X - x + y) + w_{i-1}(Y - y + r_i).
$$

Then

$$\begin{aligned}
w_i(X) + w_{i-1}(Y) &= w_{i-1}(X - x + r_i) + w_{i-1}(Y) + r_i x \\
&\geq w_{i-1}(X - x + y) + w_{i-1}(Y - y + r_i) + r_i x \\
&= w_{i-1}(X - x + y) - \sum_{z \in X - x + y} r_i z + \\
&\quad w_{i-1}(Y - y + r_i) + r_i x + \sum_{z \in X - x + y} r_i z \\
&\geq w_{i-1}(X) - \sum_{z \in X} r_i z + w_{i-1}(Y - y + r_i) + r_i y + \sum_{z \in X} r_i z \\
&\geq w_{i-1}(X) + w_i(Y),
\end{aligned}$$

where the second inequality holds since X is a (w_{i-1}, r_i)-minimizer and the third inequality follows from Proposition 5. Considering the left and right sides of this inequality we obtain that $w_i(X) - w_{i-1}(X) \geq w_i(Y) - w_{i-1}(Y)$. If $w_i(X) = w_{i-1}(X) + p_i$ then

$$w_i(X) - w_{i-1}(X) = p_i \geq \max_Y \{w_i(Y) - w_{i-1}(Y)\} = \Delta_i$$

by Proposition 5.

□

Now define the following potential function on the configurations.

$$\Psi_i(X) = k w_i(X) + \sum_{x \in X} \min_Y \left\{ w_i(Y) - \sum_{y \in Y} yx \right\},$$

Furthermore let

$$\Psi_i = \min_X \Psi_i(X).$$

Then we have the following bound on the extended cost.

Theorem 11 *There is a constant C such that $\nabla(X_0, \overline{\tau}) \leq 2k \cdot OPT(X_0, \overline{\tau}) + C$.*

Proof. First we show that $\nabla_i \leq \Psi_i - \Psi_{i-1}$. Suppose that $\Psi_i = \Psi_i(X)$ and Y is a configuration which satisfies Lemma 10.
If $r_i \in X$ then let Z_x be a (w_i, x)-minimizer for each $x \in X$. Using the definition of Ψ and Y we obtain that

$$\begin{aligned}
\Psi_i - \nabla_i &= kw_i(X) + \sum_{x\in X}\left(w_i(Z_x) - \sum_{z\in Z_x} xz\right) + \\
&\quad \left(w_{i-1}(Y) - \sum_{z\in Y} r_i z\right) - \left(w_i(Y) - \sum_{z\in Y} r_i z\right) \\
&= kw_i(X) + \sum_{x\in X - r_i}\left(w_i(Z_x) - \sum_{z\in Z_x} xz\right) + \left(w_{i-1}(Y) - \sum_{z\in Y} r_i z\right) \\
&\geq kw_i(X) + \sum_{x\in X}\left(w_{i-1}(Z_x) - \sum_{z\in Z_x} xz\right) \geq \Psi_{i-1},
\end{aligned}$$

where the last inequality is valid since the work function increases monotonically ($w_i(Z) \geq w_{i-1}(Z)$) and since Y is a (w_{i-1}, r_i)-minimizer. If $r_i \notin X$ then $w_i(X) = w_{i-1}(X) + p_i$, and since the work function increases monotonically we obtain that $\Psi_i(X) \geq \Psi_{i-1}(X) + kp_i \geq \Psi_{i-1} + \nabla_i$, because $p_i \geq w_i(Y) - w_{i-1}(Y)$.

By summation over the request sequence we get that $\nabla(X_0, \overline{\tau}) = \sum_{i=1}^{m} \nabla_i \leq \Psi_m - \Psi_0$, where m is the number of the requests. Let A be the last configuration of the optimal offline algorithm and note that X_0 is the initial configuration. Then by Property 1 we obtain that

$$\Psi_m \leq kw_m(A) + \sum_{x\in A} \min_Y \left\{ w_m(Y) - \sum_{y\in Y} yx \right\} \leq$$
$$kw_m(A) + \sum_{x\in A} w_m(A) \leq 2k \cdot w_m(A) \leq 2k \cdot OPT(X_0, \overline{\tau})$$

and the lemma follows. We need the additive constant since Ψ_0 might be negative, but it is independent of the input sequence. □

By Lemma 2 we get the statement of Theorem 4 immediately.

Remark There is a gap of multiplicative factor 2 in our analysis of WFA in the same way as in the case of the original k-server problem. We think so that the reason is the same (we loose this factor by using the extended cost), and we conjecture that WFA is $(2k+1)$-competitive for the problem.

4 The special case of $k = 2$

In this section we consider the special case of $k = 2$. Then each configuration has two points, thus the metric space in the MTS problem contains the two element multisets of $\mathcal{M}$. We prove that WFA is 5-competitive in this case. Consider again an initial configuration X_0 and a request sequence $(r_1, p_1), \dots, (r_m, p_m)$. Let w_i be the work function. We define a new potential function. This function is defined on the points of the metric space as follows.

$$\Psi_i(x) = \min_{\{ab\} \in \mathcal{S}} \{w_i(xa) + w_i(xb) - ab\} + \min_{\{cd\} \in \mathcal{S}} \{w_i(cd) - xc - xd\},$$

Furthermore let

$$\Psi_i = \min_{x \in \mathcal{M}} \Psi_i(x).$$

This function satisfies the following inequality which will be used in the proof of the competitiveness.

Lemma 12 *For any $x \in M$ we have*

$$\Psi_i(x) \geq min\{\Psi_i(r_i), \Psi_{i-1}(x) + p_i\}$$

Proof. Let $\{ab\}$ be the configuration that minimizes the first part of the potential function and denote a (w_i, x)-minimizer by Z which satisfies Lemma 9. So $w_i(Z) = w_{i-1}(Z) + \tau_i(Z)$ and

$$\begin{aligned}\Psi_i(x) &= w_i(xa) + w_i(xb) - ab + w_i(Z) - \sum_{z \in Z} xz \\ &= w_i(xa) + w_i(xb) - ab + w_{i-1}(Z) + \tau_i(Z) - \sum_{z \in Z} xz\end{aligned}$$

Now by Proposition 5

$$w_i(xa) = \min \begin{cases} w_{i-1}(xa) + p_i \\ w_{i-1}(xr_i) + r_i a \\ w_{i-1}(r_i a) + xr_i \end{cases} \qquad w_i(xb) = \min \begin{cases} w_{i-1}(xb) + p_i \\ w_{i-1}(xr_i) + r_i b \\ w_{i-1}(r_i b) + xr_i \end{cases}$$

We prove the lemma by case disjunction.

(1.) If $r_i \in Z$ then $w_i(Z) = w_{i-1}(Z)$. Let $Z = \{zr_i\}$. Now we've got 9 cases, although by symmetry only the following 6 need to be considered.

(a) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(xb) + p_i$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(xb) + p_i - ab + w_i(zr_i) - xz - xr_i \\ &= w_{i-1}(xa) + w_{i-1}(xb) - ab + w_{i-1}(zr_i) - xz - xr_i + 2p_i \\ &\geq \Psi_{i-1}(x) + 2p_i\end{aligned}$$

where the inequality comes from the definition of the potential function.

(b) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(xr_i) + r_i b$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(xr_i) + r_i b - ab + w_i(zr_i) - xz - xr_i \\ &\geq (w_{i-1}(xa) + w_{i-1}(xr_i) - ar_i) + (w_{i-1}(zr_i) - xz - xr_i) + p_i \\ &\geq \Psi_{i-1}(x) + p_i\end{aligned}$$

where the first inequality comes from the triangle inequality ($r_i b - ab \geq -ar_i$) and the second one follows from the definition of the potential function.

(c) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(r_i b) + xr_i$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(r_i b) + xr_i - ab + w_i(zr_i) - xz - xr_i \\ &\geq w_{i-1}(xa) + w_{i-1}(xb) - ab + w_{i-1}(zr_i) - xz - xr_i + p_i \\ &\geq \Psi_{i-1}(x) + p_i\end{aligned}$$

where the first inequality holds because of Property (1) ($w_{i-1}(r_i b) + xr_i \geq w_{i-1}(xb)$) and the second one follows from the definition of the potential function.

(d) If $w_i(xa) = w_{i-1}(xr_i) + r_i a$ and $w_i(xb) = w_{i-1}(xr_i) + r_i b$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xr_i) + r_i a + w_{i-1}(xr_i) + r_i b - ab + w_i(zr_i) - xz - xr_i \\ &= w_i(xr_i) + r_i a + w_i(xr_i) + r_i b - ab + w_i(zr_i) - xz - xr_i \\ &\geq w_i(xr_i) + w_i(zr_i) - xz + w_i(xr_i) - xr_i - r_i r_i \\ &\geq \Psi_i(r_i)\end{aligned}$$

where the first inequality holds since $r_i a + r_i b - ab \geq 0$ (triangle inequality) and the second one comes from the definition of the potential function.

(e) If $w_i(xa) = w_{i-1}(xr_i) + r_i a$ and $w_i(xb) = w_{i-1}(r_i b) + xr_i$, then

$$\begin{aligned}
\Psi_i(x) &= w_{i-1}(xr_i) + r_i a + w_{i-1}(r_i b) + xr_i - ab + w_i(zr_i) - xz - xr_i \\
&= w_i(xr_i) + r_i a + w_i(r_i b) + xr_i - ab + w_i(zr_i) - xz - xr_i \\
&\geq w_i(xr_i) + w_i(zr_i) - xz + w_i(r_i b) - r_i b - r_i r_i \\
&\geq \Psi_i(r_i)
\end{aligned}$$

where the first inequality follows from triangle inequality ($r_i a - ab \geq -r_i b$) and the second one comes from the definition of the potential function.

(f) If $w_i(xa) = w_{i-1}(r_i a) + xr_i$ and $w_i(xb) = w_{i-1}(r_i b) + xr_i$, then

$$\begin{aligned}
\Psi_i(x) &= w_{i-1}(r_i a) + xr_i + w_{i-1}(r_i b) + xr_i - ab + w_i(zr_i) - xz - xr_i \\
&= w_i(r_i a) + xr_i + w_i(r_i b) + xr_i - ab + w_i(zr_i) - xz - xr_i \\
&\geq w_i(r_i a) + w_i(r_i b) - ab + w_i(zr_i) - zr_i - r_i r_i \\
&\geq \Psi_i(r_i)
\end{aligned}$$

where the first inequality holds since $xr_i - xz \geq -zr_i$ (triangle inequality) and the second one comes from the definition of the potential function.

(2.) If $r_i \notin Z$ then $w_i(Z) = w_{i-1}(Z) + p_i$. Let $Z = \{zv\}$. Again we have 9 cases, although by symmetry only 6 need to be considered.

(a) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(xb) + p_i$, then

$$\begin{aligned}
\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(xb) + p_i - ab + w_i(zv) - xz - xv \\
&= w_{i-1}(xa) + p_i + w_{i-1}(xb) + p_i - ab + w_{i-1}(zv) + p_i - xz - xv \\
&\geq \Psi_{i-1} + 3p_i
\end{aligned}$$

where the inequality comes from the definition of the potential function.

(b) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(xr_i) + r_i b$, then

$$\begin{aligned}
\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(xr_i) + r_i b - ab + w_i(zv) - xz - xv \\
&= w_{i-1}(xa) + p_i + w_{i-1}(xr_i) + r_i b - ab + w_{i-1}(zv) - xz - xv + p_i \\
&\geq w_{i-1}(xa) + w_{i-1}(xr_i) - r_i a + p_i + w_{i-1}(zv) - xz - xv + p_i \\
&\geq \Psi_{i-1}(x) + 2p_i
\end{aligned}$$

where the first inequality comes from the triangle inequality ($r_ib - ab \geq -r_ia$) and the second one follows from the definition of the potential function.

(c) If $w_i(xa) = w_{i-1}(xa) + p_i$ and $w_i(xb) = w_{i-1}(r_ib) + xr_i$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xa) + p_i + w_{i-1}(r_ib) + xr_i - ab + w_i(zv) - xz - xv \\ &= w_{i-1}(xa) + p_i + w_{i-1}(r_ib) + xr_i - ab + w_{i-1}(zv) - xz - xv + p_i \\ &\geq w_{i-1}(xa) + p_i + w_{i-1}(xb) - ab + w_{i-1}(zv) - xz - xv + p_i \\ &\geq \Psi_{i-1}(x) + 2p_i\end{aligned}$$

where the first inequality holds because of Property (1) by ($w_{i-1}(r_ib) + xr_i \geq w_{i-1}(xb)$) and the second one follows from the definition of the potential function.

(d) If $w_i(xa) = w_{i-1}(xr_i) + r_ia$ and $w_i(xb) = w_{i-1}(xr_i) + r_ib$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xr_i) + r_ia + w_{i-1}(xr_i) + r_ib - ab + w_i(zv) - xz - xv \\ &= w_{i-1}(xr_i) + r_ia + w_{i-1}(xr_i) + r_ib - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq w_{i-1}(xr_i) + r_ia + w_{i-1}(xb) - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq w_{i-1}(xr_i) + w_{i-1}(xb) - r_ib + w_{i-1}(zv) + p_i - xz - xv \\ &\geq \Psi_{i-1}(x) + p_i\end{aligned}$$

where the first inequality follows from Property (1) ($w_{i-1}(xr_i)+r_ib \geq w_{i-1}(xb)$), the second one holds since $r_ia - ab \geq -r_ib$ (triangle inequality) and the third one comes from the definition of the potential function.

(e) If $w_i(xa) = w_{i-1}(xr_i) + r_ia$ and $w_i(xb) = w_{i-1}(r_ib) + xr_i$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(xr_i) + r_ia + w_{i-1}(r_ib) + xr_i - ab + w_i(zv) - xz - xv \\ &= w_{i-1}(xr_i) + r_ia + w_{i-1}(r_ib) + xr_i - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq w_{i-1}(xr_i) + r_ia + w_{i-1}(xb) - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq w_{i-1}(xr_i) + w_{i-1}(xb) - r_ib + w_{i-1}(zv) + p_i - xz - xv \\ &\geq \Psi_{i-1}(x) + p_i\end{aligned}$$

where the first inequality follows from Property (1) ($w_{i-1}(r_ib)+xr_i \geq w_{i-1}(xb)$), the second one holds since $r_ia - ab \geq -r_ib$ (triangle inequality) and the third one comes from the definition of the potential function.

(f) If $w_i(xa) = w_{i-1}(r_i a) + xr_i$ and $w_i(xb) = w_{i-1}(r_i b) + xr_i$, then

$$\begin{aligned}\Psi_i(x) &= w_{i-1}(r_i a) + xr_i + w_{i-1}(r_i b) + xr_i - ab + w_i(zv) - xz - xv \\ &= w_{i-1}(r_i a) + xr_i + w_{i-1}(r_i b) + xr_i - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq w_{i-1}(xa) + w_{i-1}(xb) - ab + w_{i-1}(zv) + p_i - xz - xv \\ &\geq \Psi_{i-1}(x) + p_i\end{aligned}$$

where the first inequality follows from Property (1) ($w_{i-1}(r_i a)+xr_i \geq w_{i-1}(xa)$) and ($w_{i-1}(r_i b) + xr_i \geq w_{i-1}(xb)$) and the second one comes from the definition of the potential function. □

Using Lemma 12 we can prove the following bound on the extended cost.

Theorem 13 *There is a constant C such that $\nabla(X_0, \overline{\tau}) \leq 3 \cdot OPT(X_0, \overline{\tau}) + C$.*

Proof. First we show, that $\nabla_i \leq \Psi_i - \Psi_{i-1}$. Let $\Psi_i = \Psi_i(x)$.Then by Lemma 12 we get two cases.

1. Suppose that $\Psi_i(x) \geq \Psi_i(r_i)$. Then let A be a (w_i, r_i)-minimizer from Lemma 10, so $\nabla_i = w_i(A) - w_{i-1}(A)$ and A is a (w_{i-1}, r_i)-minimizer as well. Then for some b and c

$$\begin{aligned}\Psi_i - \nabla_i = \Psi_i(x) - \nabla_i &\geq \Psi_i(r_i) - \nabla_i \\ &= w_i(r_i b) + w_i(r_i c) - bc + w_i(A) - r_i A - w_i(A) + w_{i-1}(A) \\ &= w_i(r_i b) + w_i(r_i c) - bc + w_{i-1}(A) - r_i A \\ &= w_{i-1}(r_i b) + w_{i-1}(r_i c) - bc + w_{i-1}(A) - r_i A \\ &\geq \Psi_{i-1}(r_i) \geq \Psi_{i-1}\end{aligned}$$

2. Suppose that $\Psi_i(x) \geq \Psi_{i-1}(x) + p_i$. Then

$$\Psi_i = \Psi_i(x) \geq \Psi_{i-1}(x) + p_i \geq \Psi_{i-1} + \nabla_i,$$

because from Proposition 5 comes that $w_i(A) \leq w_{i-1}(A) + p_i$.

By summation over the request sequence we get that $\nabla(X_0, \overline{\tau}) = \sum_{i=1}^{m} \nabla_i \leq \Psi_m - \Psi_0$, where m is the number of the requests. Let $A = (x, y)$ be the last configuration of the optimal offline algorithm. Then

$$\begin{aligned}\Psi_m \leq \Psi_m(x) &\leq w_m(A) + w_m(A) - xy + w_m(A) - xA \\ &\leq 3w_m(A) = 3OPT(X_0, \overline{\tau}),\end{aligned}$$

and Ψ_0 is a constant which is independent of the input.

□

By Theorem 13 and Lemma 2 we immediately obtain the main result of this section.

Theorem 14 *WFA is 5-competitive if $k = 2$.*

5 The Line

In this section we consider the special case of the line. We prove that WFA is $(2k + 1)$-competitive for the k-server problem with rejection and for that we use the same technique which was used in [3]. First we assume that all requests are in a fixed interval $[a, b]$. Let us denote the configuration that contains ℓ copies of a and $k - \ell$ copies of b as $\{a^\ell b^{k-\ell}\}$. We shall call these configurations *extreme*. There are exactly $k+1$ extreme configurations, because $\ell = 0, \ldots, k$. The next lemma shows that we can generally assume that minimizers are extreme configurations.

Lemma 15 *Assume that all requests are in the interval $[a, b]$. For any point $z \in [a, b]$ and any work function w_i, there is an $\ell \in \{0, \ldots, k\}$ such that $\{a^\ell b^{k-\ell}\}$ is a minimizer of z with respect to w_i.*

Proof. Let X be a minimizer of z with respect to w_i with all points in the interval $[a, b]$. Assume that there is a point $x \in X$ in the interval $[a, z]$. Then $X - x + a$ is also a (w_i, z)-minimizer because

$$\begin{aligned} w_i(X - x + a) - \sum_{y \in X - x + a} zy &\leq (w_i(X) + ax) - \sum_{y \in X - x + a} zy \\ &= (w_i(X) + ax) - (\sum_{y \in X} zy + ax) = w_i(X) - \sum_{y \in X} zy \end{aligned}$$

where the first inequality follows from Property (1) and the second from the fact that $az = ax + xz$. Similarly, we can exchange all points of X for either a or b. □

Theorem 16 *There is a constant C such that $\nabla(X_0, \overline{\tau}) \leq (k+1) \cdot OPT(X_0, \overline{\tau}) + C$.*

Proof. First we show this statement for a fixed interval $[a, b]$, to present the main idea and then we extend it to the infinite line.
Define a new potential Ψ_i for the line which is the sum of the values of w_i on the extreme configurations:

$$\Psi_i = \sum_{j=0}^{k} w_i(\{a^j b^{k-j}\}).$$

By Lemma 15, there is an ℓ such that $\{a^\ell b^{k-\ell}\}$ is a minimizer of r_{i+1} with respect to w_i. The increase of the potential, $\Psi_{i+1} - \Psi_i$, is equal to the increase of the work function on all extreme configurations. Since the work function increases monotonically $(w_{i+1}(X) \geq w_i(X))$, we obtain that

$$\Psi_{i+1} - \Psi_i \geq w_{i+1}(\{a^\ell b^{k-\ell}\}) - w_i(\{a^\ell b^{k-\ell}\}),$$

which is the extended cost (∇_{i+1}) used to service r_{i+1}. By summation over the request sequence we get that $\nabla(X_0, \overline{\tau}) \leq \sum_{i=1}^{m} \nabla_i \leq \Psi_m - \Psi_0$, where m is the number of the requests.
By property 1 we have $w(X) - w(Y) \leq XY$ and since we are in the interval $[a, b]$ we have $XY \leq k \cdot ab$. So

$$\begin{aligned}
\Psi_m &= \sum_{j=0}^{k} w_m(\{a^j b^{k-j}\}) \\
&\leq (k+1) \cdot (OPT(X_0, \overline{\tau}) + k \cdot ab) \\
&= (k+1) \cdot OPT(X_0, \overline{\tau}) + k \cdot (k+1) \cdot ab
\end{aligned}$$

On the other hand $\Psi_0 \geq 0$, thus the total extended cost is bounded above by $(k+1) \cdot OPT(X_0, \overline{\tau}) + C$ for some constant C.

We now turn to the infinite line. We have to compute the constant and show that it depends only on the initial configuration. Let us first observe that we can again assume that all requests are in an interval $[a, b]$ where a is the leftmost request and b is the rightmost one. But now we can not assume that ab is constant (since it depends on the request sequence). Thus we have to show that the additive term is independent of ab. We prove that it depends only on the initial configuration X_0. To prove that denote the positions in the configuration by $x_1, x_2, \ldots, x_k$ and let $|X_0| = \sum_{x_i, x_j (i<j) \in X_0} x_i x_j$. Then

$$
\begin{aligned}
\Psi_0 &= \sum_{j=0}^{k} w_0(\{a^j b^{k-j}\}) = \sum_{j=0}^{k} X_0\{a^j b^{k-j}\} \\
&= \sum_{j=0}^{k} \left(\sum_{i=1}^{j} a x_i + \sum_{i=j+1}^{k} b x_i \right) \\
&= \sum_{j=0}^{k} \left(\sum_{i=1}^{j} a x_i + \sum_{i=j+1}^{k} (ab - a x_i) \right) \\
&= \sum_{j=0}^{k} \sum_{i=j+1}^{k} ab + \sum_{j=0}^{k} \left(\sum_{i=1}^{j} a x_i - \sum_{i=j+1}^{k} a x_i \right) \\
&= \sum_{j=0}^{k} \sum_{i=j+1}^{k} ab + \sum_{i=1}^{k} \left((k+1-i) a x_i - \sum_{j=i}^{k} a x_j \right) \\
&= \sum_{j=0}^{k} \sum_{i=j+1}^{k} ab - \sum_{x_i, x_j \in X_0,\ i \leq j} x_i x_j = \sum_{j=0}^{k} \sum_{i=j+1}^{k} ab - |X_0| = \frac{k(k+1)}{2} ab - |X_0|
\end{aligned}
$$

If X_m is the final configuration of the optimal off-line algorithm, then we can do exactly the same calculation as above to prove $\Psi_m \leq (k+1) w_m(X_m) + \frac{k(k+1)}{2} ab - |X_m|$.

It follows that the extended cost is bounded above by $\Psi_m - \Psi_0 \leq (k+1) w_m(X_m) - |X_m| + |X_0| \leq (k+1) w_m(X_m) + |X_0|$ which shows that the total extended cost is bounded above by $(k+1) OPT(X_0, \overline{\tau}) + C$ where the constant is independent of the input also in the case of the infinite line. □

Theorem 17 *WFA is $(2k+1)$-competitive for k-server problem with rejection in the line.*

Proof. By Theorem 16 and Lemma 2 we get the statement. □

WFA has a small competitive ratio in this case. On the other hand it has huge (exponential) memory requirements and also time complexity as it is shown in [6] and [13]. Therefore it is an interesting question to find faster algorithms with smaller memory requirements even if they have worse competitive ratio. We note that for the classical k-server problem on general space some simpler but not constant competitive WFA type algorithm have been studied in [1]. Here we investigate an extended version of the Double-Coverage algorithm of [5]. This extended algorithm works as follows.

Algorithm *EDC* **(Extended Double-Coverage)**
The algorithm serves the request r_i as follows.

(i) Consider the closest server to r_i, denote its distance from r_i by d.

(ii) If $d \le p_i/2$ then the algorithm serves the point by moving the server there. Moreover if there are some further servers on the opposite side of the request, then the closest one among them also moves distance d into the direction of the request.

(iii) If $d > p_i/2$ then the request is rejected and the server moves $p_i/2$ distance in the direction of the request. Moreover if there are some further servers on the opposite side of the request, then the closest among them also moves $p_i/2$ distance in the direction of the request.

Theorem 18 *Algorithm EDC is* $3k$*-competitive on the line.*

Proof. Consider an arbitrary sequence of requests and denote this input by ϱ. During the analysis of the procedure we suppose that one off-line optimal algorithm and EDC are running parallel on the input. We also suppose that each request is served first by the off-line algorithm and then by the on-line algorithm. Let $\le$ be the natural ordering on the points of the line. The servers of the on-line algorithm and also the positions of the servers are denoted by $s_1, \dots, s_k$, and the servers of the optimal off-line algorithm and also the positions of the servers are denoted by $x_1, \dots, x_k$. We suppose that for the positions $s_1 \le s_2 \le \cdots \le s_k$ and $x_1 \le x_2 \le \cdots \le x_k$ are always valid, this can be achieved by swapping the notations of the servers.

We prove the theorem by the potential function technique. The potential function assigns a value to the actual positions of the servers, so the on-line and off-line costs are compared using the changes of the potential function. Let us define the following potential function:

$$\Phi = k \sum_{i=1}^{k} x_i s_i + \sum_{i<j} s_j s_i.$$

We note that this is the same function which was used in [5]. It is interesting that we can use a potential function which is independent of the penalties. Denote the value of the potential function after serving the request ϱ_i by Φ_i. Moreover denote EDC_i and OPT_i the cost on ϱ_i of EDC and the offline algorithm respectively. This cost is the distance moved by the servers of the algorithm plus the penalty of ϱ_i if it is rejected by the algorithm. We show that the following statement is valid for the potential function.

Lemma 19 *For each $i \geq 1$, the following inequality holds*

$$\Phi_i - \Phi_{i-1} \leq k \cdot OPT_i - 1/3 \cdot EDC_i.$$

Proof. We distinguish the following cases, depending on the behavior of OPT and EDC.

Case 1.a Suppose that OPT serves the request and EDC moves only one server.
We consider only the case when the request point is smaller than s_1. The other possibility where the request is greater than s_k is symmetric and can be handled in the same way. First OPT serves the request, it moves one server there and the distance moved by the server is OPT_i. The movement of the offline server increases the first part of Φ by at most $k \cdot OPT_i$, the second part does not change. Let $\delta = \min\{r_i s_1, p_i/2\}$. Then the server of EDC moves distance δ. If $\delta = r_i s_1 \leq p_i/2$ then it reaches the request and serves it, thus $EDC_i = \delta$. If $\delta = p_i/2 < r_i s_1$, then EDC does not reach the request thus it also pays the penalty which is 2δ, therefore in this case $EDC_i = 3\delta$. In the first part of the potential function $x_1 s_1$ is decreased by δ (x_1 cannot be larger than the requested point since OPT has a server on it), in the second part $s_j s_1$ is increased by δ for each j. Summarizing we obtained that

$$\Phi_i - \Phi_{i-1} \leq k \cdot OPT_i - k \cdot \delta + (k-1)\delta = k \cdot OPT_i - \delta \leq k \cdot OPT_i - 1/3 \cdot EDC_i.$$

Case 1.b Suppose that OPT serves the request and EDC moves two servers.

Then EDC has servers on both sides of the request; suppose that the closest servers are s_j and s_{j+1}. We assume that s_j is closer to r_i, the other case is completely similar. The movement of the offline server increases the first part of Φ by at most $k \cdot OPT_i$, the second part does not change. In this case two servers of EDC move both of them a distance δ where $\delta = \min\{s_j r_i, p_i/2\}$. If $\delta = s_j r_i \leq p_i/2$ then s_j reaches the request and serves it, thus $EDC_i = 2\delta$. If $\delta = p_i/2 < d$, then EDC does not reach the request thus it also pays the penalty which is 2δ, therefore in this case $EDC_i = 4\delta$. Consider now the first sum of the potential function. The j-th and the $j+1$-th parts are changing. Since $x_l = r_i$ for some l, thus one of the j-th and the $j+1$-th parts decreases by δ and the increase of the other one is at most δ, thus the first sum does not increase. The change of the second sum of Φ is $\delta\big((j-1)-(k-j)-(j)+(k-(j+1))\big) = -2\delta$. Summarizing we obtain that

$$\Phi_i - \Phi_{i-1} \leq k \cdot OPT_i - 2 \cdot \delta \leq k \cdot OPT_i - 1/2 \cdot EDC_i \leq k \cdot OPT_i - 1/3 \cdot EDC_i.$$

Case 2.a Suppose that OPT rejects the request and EDC moves only one server.
Again we consider only the case when the request point is smaller than s_1. The other possibility where the request is greater than s_k is symmetric and can be handled in the same way. OPT rejects the request, thus it has cost p_i and Φ does not change during the step of OPT. Let $\delta = \min\{s_1 r_i, p_i/2\}$. Then the server of EDC moves distance δ. If $\delta = d \leq p_i/2$ then it reaches the request and serves it, thus $EDC_i = \delta$. If $\delta = p_i/2 < d$, then EDC does not reach the request thus it also pays the penalty which is 2δ, therefore in this case $EDC_i = 3\delta$. In the first part of the potential function $x_1 s_1$ might increase by at most δ (in this case OPT has no server on the requested point), in the second part $s_j s_1$ is increased by δ for each j. Summarizing we obtained that

$$\Phi_i - \Phi_{i-1} \leq k \cdot \delta + (k-1)\delta = 2k \cdot \delta - \delta \leq 2kp_i/2 - \delta \leq k \cdot OPT_i - 1/3 \cdot EDC_i.$$

Case 2.b Suppose that OPT rejects the request and EDC moves two servers.

Then EDC has servers on both sides of the request, suppose that the closest servers are s_j and s_{j+1}. We assume that s_j is closer to r_i, the other case is completely similar. OPT rejects the requests, thus it has cost p_i and Φ does not change during the step of OPT. Two servers of EDC move both of them a distance $\delta = \min\{s_j r_i, p_i/2\}$. If $\delta = s_j r_i \leq p_i/2$ then s_j reaches the request and serves it, thus $EDC_i = 2\delta$. If $\delta = p_i/2 < d$, then EDC does not reach the request thus it also pays the penalty which is 2δ, therefore in this case $EDC_i = 4\delta$. In the first part of the potential function $x_j s_j$ and $x_{j-1}s_{j-1}$ might increase by at most δ and we can see in the same way as in Case 1.b that the second part is decreased by 2δ. Therefore we obtained that

$$\Phi_i - \Phi_{i-1} \leq 2k \cdot \delta - 2\delta \leq 2kp_i/2 - 2\delta \leq k \cdot OPT_i - 1/2 \cdot EDC_i \leq k \cdot OPT_i - 1/3 \cdot EDC_i.$$

□

By Lemma 19 one can prove the theorem easily. In this case $\Phi_f - \Phi_0 \leq k \cdot \mathrm{OPT}(\varrho) - \mathrm{EDC}(\varrho)/3$, where Φ_f and Φ_0 are the final and the starting values of the potential function. Furthermore, Φ is nonnegative, so we obtain that $\mathrm{EDC}(\varrho) \leq 3k\mathrm{OPT}(\varrho) + 3\Phi_0$, which yields that the algorithm is $3k$-competitive (Φ_0 does not depend on the input sequence only on the starting position of the servers).

□

6 Lower bounds

We prove the following lower bound on the possible competitive ratio.

Theorem 20 *For any metric space containing at least $k+1$ points no online algorithm can have smaller competitive ratio than $2k+1$ for the k-server with rejection problem.*

Proof. We prove this statement by contradiction. Consider an arbitrary metric space $\mathcal{M}$ which contains at least $k+1$ points. Suppose that there exists an online algorithm $\mathcal{A}$ on $\mathcal{M}$ which has smaller competitive ratio than $2k+1$. Without loosing generality we can assume that this algorithm is lazy, which means that it does not move any of the servers if there is a server on the request point and it never uses more than one server to serve a request. Any algorithm which is not lazy can be modified into a lazy one without increasing its cost by postponing the movements which are not used directly to serve a request. Moreover we suppose that the starting configuration contains k different points. Let $X_0 = x_1, \ldots, x_k$ be the initial configuration of the servers and let $x_{k+1} \notin X_0$ be a further point of $\mathcal{M}$. Denote by M the maximal and by L the minimal distance among the distances $x_i x_j$, $1 \leq i < j \leq k+1$. We construct the following input sequence ϱ_N for $\mathcal{A}$. Let N be a large positive integer and $\varepsilon = 1/N^2$. Each element of ϱ_N consists of the point among $x_1, \ldots, x_{k+1}$ which is not in the current configuration of algorithm $\mathcal{A}$ and a penalty ε. If a point is different from the previous one (which means that the algorithm used a server to serve the previous request) then we call it new point. The request sequence ends when the number of server movements of $\mathcal{A}$ achieves N. If $\mathcal{A}$ never uses N server movements to serve the requests then it has a last server movement. After that each request will be on the same point and the algorithm rejects it, thus its cost will tends to ∞. On the other hand an optimal algorithm uses a server to cover this last point and has constant cost of at most $N \cdot M$ using at most maximal distances to cover all new point when they arrive, and this shows that the algorithm is not constant competitive in this case. Now let W_p denote the sum of the penalties paid by algorithm $\mathcal{A}$ and W_d denote the total distance moved by the servers of $\mathcal{A}$. Then $\mathcal{A}(\varrho_N) = W_p + W_d$. Denote $q_1, \ldots, q_N$ the sequence of the new points in the input. Then the request at q_i is served by the server which was on q_{i+1} (this is the only point without a server after serving q_i). This means that $W_d = \sum_{i=1}^{N} q_i q_{i+1}$, where q_{N+1} is the point among $x_1, \ldots, x_{k+1}$ which is not in the final configuration of the algorithm. Let us note here that $q_i q_{i-1} \geq L$ for each i, therefore $W_d \geq (N-1)L$, thus W_d tends to ∞ if N tends to ∞.

Now extend the idea from [12] with the technique which is used in [15] to calculate an upper bound on $OPT(I)$. We consider the following $2k+1$ different offline algorithms all of them serving all requests, denote them by $\mathrm{OFF}_1, \ldots, \mathrm{OFF}_{2k+1}$. First we define and analyse $\mathrm{OFF}_1, \ldots, \mathrm{OFF}_k$. Suppose that the servers of OFF_j are at points $x_1, x_2, \ldots, x_{j-1}, x_{j+1}, \ldots x_{k+1}$ in the starting configuration. We can move the servers into this starting configuration using an extra constant cost at most M. This step is used to cover the first request and also to ensure that these algorithms are in different starting configurations.

The algorithms serve the requests as follows. If an algorithm OFF_j has a server at the requested point which is at position q_i, then none of the servers moves. Otherwise the request is served by the server located at point q_{i-1}. Note that the algorithms have servers on k-different points among the $(k+1)$-points. Therefore the algorithms are well-defined, if q_i does not contain a server, then each of the other points among $x_1, x_2, \ldots, x_{k+1}$ contains a server, thus there is a server located at q_{i-1}. Moreover $q_1 = x_{k+1}$, thus at the beginning each algorithm has a server at the requested point.

Next we show that the servers of algorithms $\mathrm{OFF}_1, \ldots, \mathrm{OFF}_k$ are always in different configurations. At the beginning this property is valid because of the definition of the algorithms. Now consider the step where a request is served. Call the algorithms which do not move a server for serving the request stable, and the other algorithms unstable. We note that we will prove later that there is only one unstable algorithm in each step. The server configurations of the stable algorithms remain unchanged, so these configurations remain different from each other. Each unstable algorithm moves a server from point q_{i-1}. This point is the place of the last request, thus the stable algorithms have server at it. Therefore, an unstable algorithm and a stable algorithm cannot have the same configuration after serving the request. Furthermore, each unstable algorithms moves a server from q_{i-1} to q_i, thus the server configurations of the unstable algorithms remain different from each other.

So at the arrival of the request at point q_i the servers of the algorithms are in different configurations. On the other hand, each configuration has a server at point q_{i-1}, therefore there is only one configuration where there is no server located at point q_i. Consequently, the cost of serving q_i is $q_{i-1}q_i$ for one of the algorithms and 0 for the other algorithms.

Therefore

$$\sum_{j=1}^{k} \mathrm{OFF}_j(\varrho_N) \leq k \cdot M + \sum_{i=2}^{N} q_i q_{i-1} \leq k \cdot M + W_d.$$

Now we define the algorithms $OFF_{k+1}, \ldots, OFF_{2k+1}$. In the case of OFF_{k+i} first we move the servers to the points $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k+1}$. This can be done by using an extra constant cost of at most M. Then none of the servers move, if the algorithm has a server on the request point then it serves it, otherwise the request is rejected.

Clearly, the number of requests is $N + W_p/\varepsilon$, and each request point is rejected by one of the algorithms $OFF_{k+1}, \ldots, OFF_{2k+1}$ and served with cost 0 by the others. Therefore we obtain that

$$\sum_{i=1}^{k+1} \mathrm{OFF}_{k+i}(\varrho_N) \leq (k+1) \cdot M + (N + W_p/\varepsilon)\varepsilon \leq k \cdot M + 1/N + W_p.$$

Applying these bounds we get that

$$\mathcal{A}(\varrho_N) \geq \sum_{i=1}^{2k+1} \mathrm{OFF}_i(\varrho_N) - (2k+1)\cdot M - 1/N.$$

Therefore using $OPT(\varrho_N) \leq OFF_i(\varrho_N)$ we obtain that

$$\frac{\mathcal{A}(\varrho_N)}{OPT(\varrho_N)} \geq (2k+1) - \frac{(2k+1)\cdot M + 1/N}{OPT(\varrho_N)}.$$

Now let us consider the limit of this ratio under the assumption $N \to \infty$. Note that $OPT(\varrho_N)$ is an increasing function of N. If it is bounded then by $\mathcal{A}(\varrho_N) \geq W_d \geq (N-1)L$ we obtain that this ratio tends to ∞ thus the algorithm is not constant competitive. If $OPT(\varrho_N) \to \infty$ as $N \to \infty$ then the right side of the inequality tends to $2k+1$ as N tends to ∞ thus we obtain that the algorithm cannot be better than $(2k+1)$-competitive.

□

Corollary 21 *In the case of the online k-server problem with rejection algorithm* WFA *achieves the smallest possible competitive ratio for $k=2$, and also for the k-server problem with rejection if the metric space contains $k+1$ points or it is the line.*

Corollary 22 *In the case of the online k-server problem with rejection algorithm* EDC *achieves the smallest possible competitive ratio for the line if $k=1$.*

References

[1] A. Baumgartner, T. Rudec, R. Manger, The design and analysis of a modified work function algorithm for solving the on-line k-server problem, *Computing and Informatics*, **29**, 2010, 681-700.

[2] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie, Multiprocessor scheduling with rejection, *SIAM Journal on Discrete Mathematics*, **13**, 2000, 64–78.

[3] Y. Bartal, E. Koutsoupias, On the competitive ratio of the work function algorithm for the k-server problem, *Theoretical Computer Science*, **324(2-3)**, 2004, 337–345.

[4] A. Borodin, N. Linial, M.E. Saks, An optimal on-line algorithm for metrical task system,*Jopurnal of teh ACM*, **39(4)**, 1992, 745–763.

[5] M. Chrobak, H. Karloff, T. Payne, S. Vishwanathan, New results on the server problem, *SIAM Journal on Discrete Mathematics*, **4**, 1991, 172–181.

[6] M. Chrobak, L. L. Larmore, The server problem and online games, In *On-Line Algorithms: Proceedings of a DIMACS Workshop. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 7.*, AMS, 1992, 11–64.

[7] M. Chrobak, L. L. Larmore, Metrical task systems, the server problem and the work function algorithm, in *Online algorithms: The State of the Art* (A. Fiat, and G. J. Woeginger (eds.)), LNCS 1442, Springer-Verlag Berlin, 1998, 74–96.

[8] Gy. Dósa, Y. He, Bin packing problems with rejection penalties and their dual problems, *Information and Computation*, **204**, 2006, 795–815

[9] Gy. Dósa, Y. He, Scheduling with machine cost and rejection, *Journal of Combinatorial Optimization*, **12(4)**, 2006, 337–350.

[10] L. Epstein, A. Levin, G. J. Woeginger, Graph coloring with rejection, *Journal of Computer and System Sciences*, **77(2)**, 2011, 439-447.

[11] L. Epstein, Bin packing with rejection revisited, *Algorithmica*, **56(4)**, 2010, 505–528.

[12] L. Epstein, Cs. Imreh, A. Levin, J. Nagy-György, Online File Caching with Rejection Penalties, *Algorithmica*, to appear, doi 10.1007/s00453-013-9793-0

[13] E. Koutsoupias, The k-server problem, *Computer Science Review* **3(2)**, 2009, 105–118.

[14] E. Koutsoupias, C. Papadimitriou, On the k-server conjecture, *Journal of the ACM*, **42**, 1995, 971–983.

[15] M. Manasse, L. McGeoch, D. Sleator, Competitive algorithms for server problems, *Journal of Algorithms*, **11(2)**, 1990, 208–230.

[16] J. Nagy-György, Cs. Imreh, On-line scheduling with machine cost and rejection, *Discrete Applied Mathematics*, **155(18)**, 2007, 2546-2554.

[17] S.S. Seiden, Preemptive Multiprocessor Scheduling with Rejection, *Theoretical Computer Science*, **262**, 2001, 437–458.

[18] D. Sleator, R. E. Tarjan, Amortized efficiency of list update and paging rules, *Communications of the ACM*, **28**, 1985, 202–208.