The 3-Server Problem in the Plane

Wolfgang W. Bein * Marek Chrobak † Lawrence L. Larmore ‡

Abstract

In the k-server problem we wish to minimize, in an online fashion, the movement cost of k servers in response to a sequence of requests (we assume that $k \geq 2$). The request issued at each step is specified by a point r in a given metric space M. To serve this request, one of the k servers must move to r.

It is known that if M has at least $k+1$ points then no online algorithm for the k-server problem in M has competitive ratio smaller than k. The best known upper bound on the competitive ratio in arbitrary metric spaces, by Koutsoupias and Papadimitriou [9], is $2k-1$. There are only a few special cases for which k-competitive algorithms are known: for $k=2$, when M is a tree, or when M has at most $k+2$ points.

We prove that the Work Function Algorithm is 3-competitive for the 3-server problem in the Manhattan plane. As a corollary, we obtain a 4.243-competitive algorithm for 3 servers in the Euclidean plane. The best previously known competitive ratio for 3 servers in these metric spaces was 5.

1 Introduction

The *k-server problem* is defined as follows: we are given $k \geq 2$ mobile servers that reside in a metric space M. A sequence of requests is issued, where each request is specified by a point $r \in M$. To "satisfy" this request, one of the servers must be moved to r, at a cost equal to the distance from its current location to r. An algorithm $\mathcal{A}$ for the k-server problem decides which server should be moved at each step. $\mathcal{A}$ is said to be *online* if its decisions are made without the knowledge of future requests. Our goal is to minimize the total service cost.

$\mathcal{A}$ is *C-competitive* if the cost incurred by $\mathcal{A}$ to service each request sequence ϱ is at most C times the optimal (offline) service cost for ϱ, plus possibly an additive constant independent of ϱ. The *competitive ratio of* $\mathcal{A}$ is the smallest C for which $\mathcal{A}$ is C-competitive.

The competitive ratio is frequently used to study the performance of online algorithms for the k-server problem, as well as other optimization problems. We refer the reader to the book of Borodin and El-Yaniv [1] for a comprehensive discussion of competitive analysis.

The k-server problem was introduced by Manasse, McGeoch and Sleator [11], who proved that no online algorithm for k servers has competitive ratio smaller than k if a metric space has at least $k+1$ points, and they presented an algorithm for the 2-server problem which is 2-competitive, and thus optimal, for any metric space. They also presented the *k-Server Conjecture* which states that,

*Department of Computer Science, University of Nevada, Las Vegas, NV 89154. Email: bein@cs.unlv.edu.
Research supported by NSF grant CCR-9821009.
†Department of Computer Science, University of California, Riverside, CA 92521. Email: marek@cs.ucr.edu.
Research supported by NSF grants CCR-9503498 and CCR-9988360.
‡Department of Computer Science, University of Nevada, Las Vegas, NV 89154. Email: larmore@cs.unlv.edu.
Research supported by NSF grant CCR-9821009.

for each k, there exists an online algorithm for k servers which is k-competitive in any metric space. For $k > 2$, this conjecture has been settled only in a number of special cases, including trees and spaces with at most $k + 2$ points [3, 4, 10]. Even some simple-looking special cases remain open, for example the 3-server problem on the circle, in the plane, or in 6-point spaces.

Recent research on the k-server conjecture has focussed on the *Work Function Algorithm* (WFA), as a possible candidate for a k-competitive algorithm. WFA is a tantalizingly simple algorithm that, at each step, chooses a server so as to minimize the sum of two quantities: the movement cost at this step, and the optimal cost of the new configuration. (More formally, the latter quantity is the optimal cost of serving past requests and ending in that configuration.) Thus one can think of WFA as a combination of two greedy strategies. The first one, a *short-sighted greedy*, minimizes the cost of the current move. The second, a *retrospective greedy*, moves to the "best" configuration, that is to a configuration where the optimal algorithm would have its servers if the request sequence ended now.

Chrobak and Larmore [5, 6] prove that WFA is 2-competitive for $k = 2$. They introduce a quantity called *pseudocost*, which provides an upper bound on WFA's cost. Pseudocost is independent of the location of WFA's servers and thus it is easier to estimate than the actual cost function of WFA. For $k \geq 3$, Koutsoupias and Papadimitriou [8, 9] prove that WFA is $(2k - 1)$-competitive for k servers in arbitrary metric spaces. Their proof is based on the pseudocost method. The pseudocost approach can also be effectively used to prove that a generalization of WFA is competitive for other problems, including task systems [6], page migration [7], and the layered graph traversal problem [2].

The main result of this paper is that WFA is 3-competitive for 3 servers in the Manhattan plane. Our proof is based on the pseudocost method outlined above, extending the work from [5, 6, 9]. The main difficulty in competitive analysis, and in amortized analysis in general, is the construction of an appropriate potential function. In our proof we use the "lazy" potential function. The general idea behind this potential is that the "worst" request sequences are those on which the adversary cost is zero (see Section 3 for a formal definition). We divide the proof into two steps. First, we formulate certain conditions on a metric space M under which the lazy potential provides a certificate that WFA is 3-competitive in M. We then show that the Manhattan plane satisfies these conditions, allowing us to conclude that WFA for 3 servers is 3-competitive in the Manhattan plane. Since the Euclidean metric can be approximated by the Manhattan metric with a "distortion" of $\sqrt{2}$, we obtain a $3\sqrt{2}$-competitive algorithm for the 3-server problem in the Euclidean plane.

2 Preliminaries

Let M be a metric space. For points $x, y \in M$, we write xy to denote the distance between x and y. Unordered k-tuples of points in M will be called *configurations*, and they will represent positions of our k servers. Configurations will be denoted by capital letters $A, B, X, Y, \ldots$. We write XY to denote the minimum-matching distance between configurations X and Y. The set of configurations is itself a metric space under the minimum-matching metric.

For simplicity, we assume that the initial server configuration S^0 is fixed. We allow an algorithm to move any number of servers before or after each request, as long as after a request r is issued and before the next request (if any), at least one of the servers visits r. It is a simple exercise to verify that this additional freedom does not change the problem.

Work functions. Work functions provide information about the optimal cost of serving the past request sequence. For a request sequence ϱ, by $\omega_\varrho(X)$ we denote the minimum cost of serving ϱ

and ending in configuration X. We refer to ω_ϱ as the *work function after* ϱ. We use notation ω to denote any work function ω_ϱ, for some request sequence ϱ. Immediately from the definition of work functions we conclude that the optimal cost to service ϱ is $opt(\varrho) = \min_X \omega_\varrho(X)$.

For a given ϱ, we can compute ω_ϱ using dynamic programming. Initially, $\omega_\epsilon(X) = S^0X$, for each configuration X (ϵ is the empty request sequence). For a non-empty request sequence ϱ, if r is the last request in ϱ, write $\varrho = \sigma r$. Then ω_ϱ can be computed recursively as $\omega_\varrho = \omega_\sigma \wedge r$, where "$\wedge$" is the *update operator* defined as follows:

$$\omega \wedge r(X) \;=\; \min_{Y \ni r} \{\omega(Y) + YX\} \tag{U}$$

Note that $|\omega(X) - \omega(Y)| \leq XY$ for any work function ω and any configurations X and Y. This inequality we call the *Lipschitz property.*

If $r \in X$, then $\omega \wedge r(X) = \omega(X)$. If $r \notin X$, let Y be the configuration that contains r and minimizes $\omega(Y) + YX$, and let x be the point in X that is matched to r in the minimum matching between X and Y. Then $\omega(Y) + YX = \omega(Y) + rx + Y(X - x + r) \geq \omega(X - x + r) + rx$. Thus the update formula (U) can be rewritten as $\omega \wedge r(X) = \min_{x \in X} \{\omega(X - x + r) + rx\}$. Yet another, equivalent definition of update is that $\omega \wedge r$ is the unique maximal function which has the Lipschitz property and which agrees with ω on all configurations which contain r.

Koutsoupias and Papadimitriou [8, 9] prove that work functions also satisfy the following property, which they name *quasiconvexity*:

$$\omega(X) + \omega(Y) \;\geq\; \max_{x \in X - Y} \min_{y \in Y - X} \{\omega(X - x + y) + \omega(Y - y + x)\} \tag{Q}$$

The Lipschitz property and quasiconvexity will be used extensively in our calculations.

Maximizers. Define the *shadow* of ω as

$$\widetilde{\omega}(x) \;=\; \max_A \left\{ \sum_{a \in A} xa - \omega(A) \right\} \tag{1}$$

A configuration A is called an (ω, x)*-maximizer* if A maximizes the right-hand side in (1), that is, $\widetilde{\omega}(x) = \sum_{a \in A} xa - \omega(A)$. If r is the last request in ω, then $\omega(A) = \omega(A - b + r) + br$, for some $b \in A$. Using the triangle inequality, we obtain

$$\begin{aligned} \sum_{a \in A} xa - \omega(A) \;&=\; \sum_{a \in A} xa - br - \omega(A - b + r) \\ &\leq\; \sum_{a \in A - b + r} xa - \omega(A - b + r). \end{aligned}$$

We conclude that, without loss of generality, an (ω, x)-maximizer contains the last request.

Lemma 1 [9] *Suppose that A is an (ω, x)-maximizer. Then*

(a) *A is an $(\omega \wedge x, x)$-maximizer.*

(b) *A maximizes $\omega \wedge x(A) - \omega(A)$.*

Proof: The proofs for (a) and (b) use the quasiconvexity property and are quite similar. We give the proof for (b), and refer the reader to [9] for the proof of (a). (See also [6].)

To show (b), it is sufficient to prove that

$$\omega\wedge x(A) + \omega(B) \quad \geq \quad \omega(A) + \omega\wedge x(B) \tag{2}$$

for each configuration B. If $x \in B$ then $\omega(B) = \omega\wedge x(B)$ and (2) follows, since $\omega\wedge x \geq \omega$. Suppose $x \notin B$. Since A is a (ω, x)-maximizer, $ax - \omega(A) \geq bx - \omega(A - a + b)$ for all $a \in A$ and $b \notin A$. Then, using quasiconvexity, we have

$$\begin{aligned}
\omega\wedge x(A) + \omega(B) &= \min_{a\in A}\{\omega(A - a + x) + ax + \omega(B)\} \\
&\geq \min_{a\in A}\min_{b\in B-A}\{\omega(A - a + b) + ax + \omega(B - b + x)\} \\
&\geq \min_{b\in B-A}\{\omega(A) + bx + \omega(B - b + x)\} \\
&\geq \omega(A) + \omega\wedge x(B)
\end{aligned}$$

and (2) follows. □

The Work Function Algorithm. We define the *Work Function Algorithm* (WFA) to be an algorithm which chooses its service of the request sequence ϱ as follows: Suppose that WFA is in configuration S, and that the current work function is ω. On request r, WFA chooses some $x \in S$ which minimizes $xr + \omega\wedge r(S - x + r)$, and moves the server from x to r. If there is more than one choice which minimizes that quantity, the choice is arbitrary.

WFA can be seen as a "linear combination" of two greedy strategies. The first one, a *short-sighted greedy*, minimizes the cost xr in the given step. The second, a *retrospective greedy*, chooses the optimal configuration after r, that is, the configuration $S - x + r$ that minimizes $\omega\wedge r(S - x + r)$. The short-sighted greedy strategy is not competitive for any k. The retrospective greedy strategy is not competitive for $k \geq 3$, and its competitive ratio for $k = 2$ is at least 3.

Since $r \in S - x + r$, we have $\omega\wedge r(S - x + r) = \omega(S - x + r)$, so WFA can just as well minimize $xr + \omega(S - x + r)$. Yet another possible formulation is that WFA moves to a new configuration S' that contains r and minimizes $SS' + \omega(S')$. This is actually how the work-function algorithm is defined in the general setting of metrical task systems [1, 4], so let us refer to this formulation as *general* WFA (GWFA). GWFA always has a move in which only one server moves to the request point, but without this restriction, on some request sequences, GWFA can incur larger cost than WFA. Nevertheless, by examining the proof of Lemma 2, it is easy to see that our result applies to GWFA as well.

The pseudocost method. Pseudocost is a function that provides an upper bound on WFA's cost. More accurately, pseudocost actually bounds the *sum* of WFA's and the optimal costs. Since pseudocost is algorithm-independent, it is easier to deal with than the actual cost of WFA.

For any work function ω and $r \in M$, we consider the maximum increase of the work function if r is requested:

$$\nabla_r(\omega) \quad = \quad \max_X\{\omega\wedge r(X) - \omega(X)\}$$

As an immediate consequence of Lemma 1, we obtain:

Corollary 1 *Given a work function ω, let $\mu = \omega\wedge s$ for some $s \in M$. Then $\widetilde{\mu}(s) + \nabla_s(\omega) = \widetilde{\omega}(s)$.*

Suppose that $\varrho = r^1 \ldots r^n$, and let ω^t denote the work function after requests $r^1 \ldots r^t$. The *pseudo-cost* of ϱ is defined as $\nabla_\varrho = \sum_{t=1}^n \nabla_{r^t}\left(\omega^{t-1}\right)$. The lemma below establishes a relationship between the cost of WFA and the pseudocost.

Lemma 2 [1, 5, 9] $\nabla_\varrho \geq cost_{\mathrm{WFA}}(\varrho) + opt(\varrho)$.

Proof: Let $S^0, \ldots, S^n$ be the configurations of WFA for $\varrho = r^1 \ldots r^n$. The definitions of WFA and the update operator imply that $\omega^t(S^{t-1}) = \omega^t(S^t) + S^t S^{t-1}$. Since $opt(\varrho) \leq \omega^n(S^n)$ and $\omega^0(S^0) = 0$, we have

$$\begin{aligned}
cost_{\mathrm{WFA}}(\varrho) + opt(\varrho) &\leq \sum_{t=1}^{n} S^{t-1} S^t + \omega^n(S^n) \\
&= \sum_{t=1}^{n} \left[\omega^t(S^{t-1}) - \omega^t(S^t) \right] + \omega^n(S^n) \\
&= \sum_{t=1}^{n} \left[\omega^t(S^{t-1}) - \omega^{t-1}(S^{t-1}) \right] \\
&\leq \nabla_\varrho
\end{aligned}$$

□

According to the above lemma, in order to prove that WFA is C-competitive, it is sufficient to show that the pseudocost is $(C+1)$-competitive. To prove the latter fact, we use a standard potential argument, formalized in the lemma below.

Lemma 3 *Let $\Phi_{\omega,r} \in \mathbf{R}$ be defined for each work function ω with the last request r. Suppose that $\Phi_{\omega,r}$ satisfies the following properties:*

(OP) $\Phi_{\omega,r} + (C+1)\min(\omega) \;\geq\; 0,$

(UP) *If* $\mu = \omega \wedge s$ *then* $\Phi_{\mu,s} + \nabla_s(\omega) \;\leq\; \Phi_{\omega,r}.$

Then WFA *is C-competitive on M.*

Conditions (OP) and (UP) are referred to as the *offset property* and the *update property*, respectively, and $\Phi_{\omega,r}$ is called a *potential function.*

Proof: We use the notation from the previous lemma. Write $\Phi^t = \Phi_{\omega^t, r^t}$. By amortized summation:

$$\begin{aligned}
\nabla_\varrho &= \sum_{t=1}^{n} \nabla_{r^t}\left(\omega^{t-1}\right) \\
&\leq \sum_{t=1}^{n} \left(\Phi^{t-1} - \Phi^t\right) \\
&= \Phi^0 - \Phi^n \\
&\leq (C+1)\, opt(\varrho) + \Phi^0.
\end{aligned}$$

Φ^0 is independent of ϱ. By Lemma 2, it follows that WFA is C-competitive. □

3 The Lazy Potential

We now assume that $k = 3$, and we let M be an arbitrary but fixed metric space. In the remainder of this section, and later throughout the paper, we assume that ω is a work function with the last request r. Unless otherwise stated, small letters $a, b, c, \ldots$ possibly with accents or subscripts, denote points in M.

Our goal in this section is to construct a potential function that satisfies Lemma 3. Constructing a potential function is the main difficulty in proving competitiveness. One approach that applies to many online problems is the "lazy adversary" idea, which is used to prove that WFA is 2-competitive for 2 servers [5, 6].

Before we give the definition of such a potential, we informally describe how it can be derived. The intuition behind the potential is that it represents how much of the pseudocost we need to have saved in the past in order to be able to pay for future requests while remaining 3-competitive. One possible future request sequence is a "lazy" sequence, in which the adversary alternates requests on three points (the locations of his three servers) incurring no cost. Clearly, the current potential must be at least as large as the maximum pseudocost on such a sequence. So the maximum of the pseudocosts on lazy request sequences is a reasonable guess for a potential function. In the case of the server problem we need to be more careful, since not all server configurations are "equally good". Intuitively, configurations with servers being far apart require more savings than configurations in which servers are close to each other. So we modify the potential function by adding another term that takes the configuration "size" into account.

Let us describe now this idea in more detail. As usual, ω denotes the current work function, and r is the last request. The adversary must be in some configuration X, where $r \in X$. To construct a potential $\Phi_{\omega,r}$ we assume that the adversary will be "lazy" for sufficiently many requests, *i.e.*, that the adversary will request a sequence $\xi = s^1 \ldots s^m$ following r, where each $s^i \in X$. Furthermore, ξ is long enough to ensure that $\omega \wedge \xi = \chi$ is a *cone* on X, *i.e.*, $\chi(Y) = \chi(X) + XY$, for all Y. Clearly, we have $\chi(X) = \omega(X)$. We conclude (since ξ is assumed to be the "worst case" choice) that $\Phi_{\omega,r} \geq \nabla_\xi(\omega) + \Phi_{\chi,s^m}$. It remains to estimate Φ_{χ,s^m}. To this end, we use the following fact: if WFA is 3-competitive in all metric spaces, then there exists a potential function Φ such that $\Phi_{\chi,x} = \Sigma X - 4 \cdot \chi(X)$, where χ is a cone on X, $x \in X$, and ΣX denotes the sum of all distances in X. (Since we do not need this fact for proving our result, we omit the proof.) Thus our estimate of $\Phi_{\omega,r}$ is obtained by adding $\Sigma X - 4 \cdot \omega(X)$ to the maximum pseudocost on sequences in X, and then maximizing over all configurations X with $r \in X$.

We give, below, a closed form for this estimate, which we denote by $\Psi_{\omega,r}$. We call $\Psi_{\omega,r}$ the "lazy potential."

We define functions:

$$
\begin{aligned}
\overline{\omega}(x,y) &= \max_{a,a'} \left\{ ya + ya' - \omega(x,a,a') \right\} \\
\dot{\omega}(x) &= \max_{p,d,d'} \left\{ \overline{\omega}(x,p) + dd' - \omega(x,p,d) - \omega(x,p,d') \right\} \\
\Psi_{\omega,x} &= \widetilde{\omega}(x) + \dot{\omega}(x)
\end{aligned}
$$

Since, for each z, there is an (ω, z)-maximizer that contains the last request r, $\widetilde{\omega}$ and $\overline{\omega}$ are closely related, namely $\widetilde{\omega}(z) = \overline{\omega}(r,z) + rz$.

In this notation, the lazy potential of a work function ω with last request r is

$$\Psi_{\omega,r} = \widetilde{\omega}(r) + \dot{\omega}(r) = \overline{\omega}(r,r) + \dot{\omega}(r)$$

We point out that, although $\Psi_{\omega,r}$ is the lazy potential, we never prove that fact in this paper. The proof is rather involved, and is unnecessary for the correctness of the main results given here.

We now define three more functions:

$$
\begin{aligned}
\Lambda_{\omega,x} &= \max_{p,q,e,e'} \{-xp + \overline{\omega}(x,p) - xq + \overline{\omega}(x,q) - \omega(x,p,q) + ee' - \omega(x,e,e')\} \\
\Gamma_{\omega,x} &= \max_{p,q,d,d',f} \{-xp + \overline{\omega}(x,p) + xq + dd' - \omega(x,q,d) - \omega(x,q,d') + qf - \omega(x,p,f)\} \\
\widehat{\Psi}_{\omega,x} &= \max\{\Psi_{\omega,x}, \Lambda_{\omega,x}, \Gamma_{\omega,x}\}.
\end{aligned}
$$

We remind the reader that in the derivation of $\Psi_{\omega,r}$ we assume that the adversary will include the last request r in its configuration. If we allow request sequences on arbitrary configurations, not necessarily containing r, the resulting value is called the *semi-lazy potential* and it is equal to $\widehat{\Psi}_{\omega,r}$. We state this fact without proof.

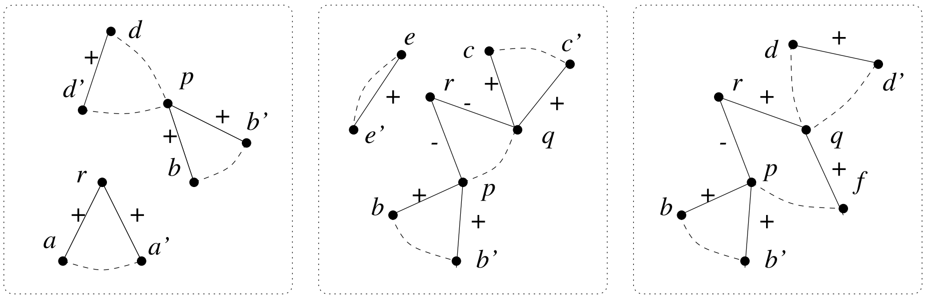

Figure 1: A graphical representation of functions $\Psi_{\omega,r}$, $\Lambda_{\omega,r}$ and $\Gamma_{\omega,r}$.

The formulas for $\Psi_{\omega,r}$, $\Lambda_{\omega,r}$ and $\Gamma_{\omega,r}$ are illustrated graphically in Figure 1. In this diagram, a solid line between x, y represents the distance xy, and a dashed line between x, y represents $\omega(r, x, y)$. All work function value terms are negative. The "+" or "−" labels on solid lines show whether the corresponding distance is a positive or a negative term.

If no ambiguity arises, we simplify notation by writing $\omega(x, y)$, instead of $\omega(r, x, y)$. The quasi-convexity property implies that

$$
\omega(x,y) + \omega(u,v) \quad \geq \quad \min \left\{ \begin{array}{c} \omega(x,u) + \omega(y,v) \\ \omega(x,v) + \omega(y,u) \end{array} \right\} \tag{3}
$$

for any x, y, u, v. Instead of the general quasiconvexity property, we will only use this last inequality in the calculations. In many cases, we use symmetry to eliminate the minimum or avoid splitting the computation into cases. For example, if $x = y$, then (3) implies that $\omega(x,x) + \omega(u,v) \geq \omega(x,u) + \omega(x,v)$. The same trick can be applied even if x, y are different but exchangeable in the formula under consideration.

Our proof involves extensive calculations involving the triangle inequality and the quasiconvexity property (3). To facilitate reading, we label the steps with symbols $\triangle$ or Q, respectively, to indicate which inequality is being applied at a given step. Some steps may have more than one label.

Before stating the main results of this section, we prove the following technical lemma.

Lemma 4 *Let ω be a work function over M with last request r. Then $\Psi_{\omega,s} \leq \widehat{\Psi}_{\omega,r}$ for all $s \in M$.*

Proof: Pick p, a, a', b, b', d, d' such that

$$
\Psi_{\omega,s} \;=\; sr \;+\; sa + sa' - \omega(a,a') \;+\; pb + pb' - \omega(s,b,b') + dd' - \omega(s,p,d) - \omega(s,p,d') \tag{4}
$$

For any x, y, we have

$$\omega(s,x,y) \quad = \quad \min \left\{ \begin{array}{c} \omega(x,y) + rs \\ \omega(s,x) + ry \\ \omega(s,y) + rx \end{array} \right\} \tag{5}$$

The proof is by analysis of cases, depending on which of the three choices in equation (5), for each of $\omega(s,b,b')$, $\omega(s,p,d)$ and $\omega(s,p,d')$, realizes the minimum. Because of the symmetries between b and b', and between d and d', we can reduce the number of cases to twelve. Our goal, in each case, is to bound formula (4) from above by an expression that is an instance of either $\Psi_{\omega,r}$, $\Lambda_{\omega,r}$ or $\Gamma_{\omega,r}$ for some choice of parameters. In most cases the resulting expression will be an instance of $\Psi_{\omega,r}$. In two cases we obtain an average of two instances of $\Psi_{\omega,r}$, while $\Lambda_{\omega,r}$ and $\Gamma_{\omega,r}$ appear in only one case each.

Case 1: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(p,d') + rs$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & sa - rs + sa' - rs - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + dd' - \omega(p,d) - \omega(p,d') \\ \leq {} & ra + ra' - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + dd' - \omega(p,d) - \omega(p,d') \qquad (\triangle,\triangle) \end{aligned}$$

Case 2: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(s,p) + rd'$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & sa - rs + dd' - rd' - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + sa' - \omega(p,d) - \omega(p,s) \\ \leq {} & ra + rd - \omega(a,d) \quad + pb + pb' - \omega(b,b') \quad + sa' - \omega(p,s) - \omega(p,a') \qquad (\triangle,\triangle,\mathrm{Q}) \end{aligned}$$

Case 3: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(s,p) + rd$, and $\omega(s,p,d') = \omega(s,p) + rd'$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & sa - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + sa' - 2\omega(s,p) \quad + dd' - rd - rd' \\ \leq {} & rs + ra - \omega(s,a) \quad + pb + pb' - \omega(b,b') \quad + sa' - \omega(s,p) - \omega(p,a') \qquad (\triangle,\triangle,\mathrm{Q}) \end{aligned}$$

Case 4: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(s,p) + rd$, and $\omega(s,p,d') = \omega(s,d') + rp$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & pb' - rp + dd' - rd - \omega(b,b') \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,d') \\ \leq {} & rb' + rd' - \omega(b',d') \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \qquad (\triangle,\triangle,\mathrm{Q}) \end{aligned}$$

Case 5: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(s,d) + rp$, and $\omega(s,p,d') = \omega(s,d') + rp$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & pb - rp + pb' - rp - \omega(b,b') \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \\ \leq {} & rb + rb' - \omega(b,b') \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \qquad (\triangle,\triangle) \end{aligned}$$

Case 6: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(s,p) + rd'$. After substituting in (4) and rearranging, we have

$$\begin{aligned} \Psi_{\omega,s} = {} & pb' - rb' + dd' - rd' - \omega(p,d) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \\ \leq {} & rp + rd - \omega(p,d) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \qquad (\triangle,\triangle) \end{aligned}$$

Case 7: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(s,p) + rd$, and $\omega(s,p,d') = \omega(s,p) + rd'$. After substituting in (4) and rearranging, we have

$$\begin{aligned}\Psi_{\omega,s} = &\ rs + pb' - rb' - \omega(s,p) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \quad + dd' - rd - rd' \\ \le &\ rs + rp - \omega(s,p) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \qquad (\triangle,\triangle)\end{aligned}$$

Case 8: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(s,p) + rd$, and $\omega(s,p,d') = \omega(s,d') + rp$. After substituting in (4) and rearranging, we have

$$\begin{aligned}\Psi_{\omega,s} = &\ rs + dd' - rd - \omega(s,d') \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \quad + pb' - rb' - rp \\ \le &\ rs + rd' - \omega(s,d') \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \qquad (\triangle,\triangle)\end{aligned}$$

Case 9: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(s,d) + rp$, and $\omega(s,p,d') = \omega(s,d') + rp$. After substituting in (4) and rearranging, we have

$$\begin{aligned}\Psi_{\omega,s} = &\ rs + pb - rp - \omega(s,b) \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \quad + pb' - rb' - rp \\ \le &\ rs + rb - \omega(s,b) \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \qquad (\triangle,\triangle)\end{aligned}$$

Case 10: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(s,d') + rp$. If $\omega(s,b) + \omega(p,d) \ge \omega(p,b) + \omega(s,d)$ then after substituting in (4) and rearranging, we have

$$\begin{aligned}\Psi_{\omega,s} = &\ pb - \omega(s,b) \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(p,d) - \omega(s,d') \quad + pb' - rb' - rp \\ \le &\ rp + rb - \omega(p,b) \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \qquad (\triangle,\triangle,\mathrm{Q})\end{aligned}$$

Otherwise, by quasiconvexity, we can assume that $\omega(s,b) + \omega(p,d) \ge \omega(b,d) + \omega(p,s)$. If $\omega(p,d) + \omega(s,d') \ge \omega(d,d') + \omega(s,p)$, then

$$\begin{aligned}\Psi_{\omega,s} = &\ dd' - \omega(p,d) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,b) - \omega(s,d') \quad + pb' - rb' - rp \\ \le &\ rd + rd' - \omega(d,d') \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,b) - \omega(s,p) \qquad (\triangle,\triangle,\mathrm{Q})\end{aligned}$$

Else, $\omega(p,d) + \omega(s,d') \ge \omega(p,d') + \omega(s,d)$, and we have

$$\begin{aligned}\Psi_{\omega,s} = &\ \tfrac{1}{2} \cdot \big[\, dd' - \omega(s,b) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(p,d) - \omega(s,b) \\ &\qquad + \ pb - \omega(p,d) \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d') - \omega(s,d') \,\big] \quad + pb' - rb' - rp \\ \le &\ \tfrac{1}{2} \cdot \big[\, rb + rd - \omega(b,d) \quad + sa + sa' - \omega(a,a') \quad + pb - \omega(s,p) - \omega(s,b) \big] \qquad (\triangle,\triangle,\ \mathrm{Q}) \\ &\ + \tfrac{1}{2} \cdot \big[rp + rd' - \omega(p,d') \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \big] \qquad (\triangle,\ \mathrm{Q})\end{aligned}$$

In each case so far, except the last one, we obtained an expression that is an instance of $\Psi_{\omega,r}$ for some choice of parameters. In this last case, the final value is an average of two instances of $\Psi_{\omega,r}$. In the remaining two cases that follow we will obtain expressions that are instances of $\Lambda_{\omega,r}$, $\Gamma_{\omega,r}$, or $\Psi_{\omega,r}$.

Case 11: $\omega(s,b,b') = \omega(s,b) + rb'$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(p,d') + rs$. After substituting in (4) and rearranging, we have

$$\begin{aligned}\Psi_{\omega,s} = &\ -rs + sa + sa' - \omega(a,a') \quad + pb' - rb' + dd' - \omega(p,d) - \omega(p,d') \quad + pb - \omega(s,b) \\ \le &\ -rs + sa + sa' - \omega(a,a') \quad + rp + dd' - \omega(p,d) - \omega(p,d') \quad + pb - \omega(s,b) \qquad (\triangle) \\ \le &\ \Gamma_{\omega,r}\end{aligned}$$

Case 12: $\omega(s,b,b') = \omega(b,b') + rs$, $\omega(s,p,d) = \omega(p,d) + rs$, and $\omega(s,p,d') = \omega(s,d') + rp$. If $\omega(s,d') + \omega(p,d) \geq \omega(s,p) + \omega(d,d')$ then, after substituting in (4) and rearranging, we have

$$\begin{aligned}
\Psi_{\omega,s} = & -rp + pb + pb' - \omega(b,b') \quad - rs + sa + sa' - \omega(a,a') \quad - \omega(p,d) \quad + dd' - \omega(s,d') \\
\leq & -rp + pb + pb' - \omega(b,b') \quad - rs + sa + sa' - \omega(a,a') \quad - \omega(s,p) \quad + dd' - \omega(d,d') \qquad \text{(Q)} \\
\leq & \ \Lambda_{\omega,r}
\end{aligned}$$

Otherwise, by quasiconvexity, $\omega(s,d') + \omega(p,d) \geq \omega(s,d) + \omega(p,d')$. Then

$$\begin{aligned}
\Psi_{\omega,s} = & \ \tfrac{1}{2} \cdot \big[\, sa + sa' - 2 \cdot rs - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + dd' - \omega(s,d') - \omega(p,d) \\
& \quad pb + pb' - 2 \cdot rp - \omega(b,b') \quad + sa + sa' - \omega(a,a') dd' - \omega(s,d') - \omega(p,d) \,\big] \\
\leq & \ \tfrac{1}{2} \cdot \big[ra + ra' - \omega(a,a') \quad + pb + pb' - \omega(b,b') \quad + dd' - \omega(p,d) - \omega(p,d') \big] \qquad (\triangle,\triangle,\text{Q}) \\
& + \tfrac{1}{2} \cdot \big[rb + rb' - \omega(b,b') \quad + sa + sa' - \omega(a,a') \quad + dd' - \omega(s,d) - \omega(s,d') \big] \qquad (\triangle,\triangle) \\
\leq & \ \Psi_{\omega,r}
\end{aligned}$$

As explained earlier, each of the remaining cases is symmetric to one of the cases 1-12. This completes the proof. □

The following theorem shows that, for any metric space M, if the semi-lazy potential is not larger than the lazy potential on M, then the 3-server problem on M is 3-competitive. In the next section, we prove that this property holds for the Manhattan plane. We also have shown (but have not included the proof in this paper) that the result also holds for any 6-point space.

Theorem 1 *Let M be any metric space. If $\widehat{\Psi}_{\omega,r} \leq \Psi_{\omega,r}$ for every work function ω over M with last request r, then* WFA *for 3 servers on M is 3-competitive.*

Proof: We apply Lemma 3. First, we verify the offset property (OP). Suppose that ω is minimized on configuration $X = \{r,a,b\}$. By choosing suitable points in the formula for $\Psi_{\omega,r}$, we obtain

$$\begin{aligned}
\Psi_{\omega,r} \ & \geq \ ra + rb - \omega(r,a,b) + aa + ab - \omega(r,a,b) + bb - \omega(r,a,b) - \omega(r,a,b) \\
& = \ ra + rb + ab - 4 \cdot \omega(r,a,b),
\end{aligned}$$

and (OP) follows. Condition (UP) follows immediately from Lemma 4 and Corollary 1:

$$\begin{aligned}
\Psi_{\mu,s} + \nabla_s(\omega) & = \widetilde{\mu}(s) + \dot{\mu}(s) + \nabla_s(\omega) \\
& = \widetilde{\omega}(s) + \dot{\mu}(s) && \text{(by Corollary 1)} \\
& = \widetilde{\omega}(s) + \dot{\omega}(s) \\
& = \Psi_{\omega,s} \\
& \leq \widehat{\Psi}_{\omega,r} && \text{(by Lemma 4)} \\
& \leq \Psi_{\omega,r} && \text{(by hypothesis)}
\end{aligned}$$

□

4 WFA in the Manhattan Plane

In this section we assume that $k = 3$ and the given metric space is the Manhattan plane $\mathbf{R}^2_1$, that is $\mathbf{R}^2$ with the city-block metric: $xy = |x^1 - y^1| + |x^2 - y^2|$, where x^1, x^2 denote the coordinates of a point x.

The main result of this paper is the following theorem.

Theorem 2 *WFA is 3-competitive for 3 servers in the Manhattan plane.*

Proof: Lemmas 5 and 6 below imply that $\widehat{\Psi}_{\omega,r} \leq \Psi_{\omega,r}$, and the result follows immediately from Theorem 1. $\square$

It remains to show that $\Lambda_{\omega,r}, \Gamma_{\omega,r} \leq \Psi_{\omega,r}$. The proofs are presented in the two lemmas below. As in the previous section, in our calculations we extensively use the triangle inequality, the Lipschitz property and quasiconvexity. We label the steps with symbols $\triangle$, L or Q, respectively, to indicate which inequality is being applied at a given step. Some steps may have more than one label. Instead of the general quasiconvexity property, we use the simplified formula (3), and, whenever possible, we avoid splitting the computation into cases by using symmetries.

Lemma 5 *Let ω be a work function with last request r in the Manhattan plane. Then $\Lambda_{\omega,r} \leq \Psi_{\omega,r}$.*

Proof: Pick $p, b, b', q, c, c', e, e'$ such that

$$\Lambda_{\omega,r} \quad = \quad -rp + pb + pb' - \omega(b, b') - rq + qc + qc' - \omega(c, c') - \omega(p, q) + ee' - \omega(e, e') \qquad (6)$$

Let R be a rectangle with corners x, z, y, t (ordered clockwise) that contains all points in formula (6). Point b must be located between p and some corner of R, say y, that is, $pb = py - by$. Then $pb - \omega(b, b') = py - by - \omega(b, b') \leq py - \omega(y, b')$ by the Lipschitz property of ω. Thus we can replace b by y in (6). By the same argument, each of the points b', c, c', e and e' can be replaced by a corner of R. Furthermore, without loss of generality, e and e' will be in opposite corners. Therefore,

$$\Lambda_{\omega,r} \quad = \quad -rp + py + pb' - \omega(y, b') - rq + qc + qc' - \omega(c, c') - \omega(p, q) + ee' - \omega(e, e') \qquad (7)$$

where $b', c, c', e, e' \in \{x, z, y, t\}$ and $\{e, e'\} = \{x, y\}$ or $\{z, t\}$. We consider a number of cases, depending on which points are in which corners. Note that in the formula for Λ, triples (p, b, b') and (q, c, c') can be exchanged. Further, b can be exchanged with b', c can be exchanged with c', and e can be exchanged with e'. We use these symmetries to reduce the number of cases to consider. Our goal, in each case, is to bound formula (7) from above by an expression that is an instance of $\Psi_{\omega,r}$, for some choice of parameters.

Case 1: $b' = x$. Then $py + pb' = py + px = xy$. After substituting in (7) and rearranging, we have

$$\begin{aligned}
\Lambda_{\omega,r} &= ee' - \omega(e, e') \quad + qc + qc' - \omega(c, c') \quad + xy - \omega(x, y) - \omega(p, q) - rp - rq \\
&\leq ee' - \omega(e, e') \quad + qc + qc' - \omega(c, c') \quad + xy - \omega(q, x) - \omega(p, y) - pq && (\triangle,\text{Q}) \\
&\leq re + re' - \omega(e, e') \quad + qc + qc' - \omega(c, c') \quad + xy - \omega(q, x) - \omega(q, y) && (\triangle,\text{L})
\end{aligned}$$

Case 2: $b' = y$. There are three subcases.

Case 2.1: $c = y$. After substituting in (7) and rearranging, we have

$$\begin{aligned}
\Lambda_{\omega,r} &= -rp + py - rq + qc' - \omega(y, c') \quad + yp + yq - \omega(p, q) \quad + ee' - \omega(y, y) - \omega(e, e') \\
&\leq ry + rc' - \omega(y, c') \quad + yp + yq - \omega(p, q) \quad + ee' - \omega(y, e) - \omega(y, e') && (\triangle,\ \triangle,\ \text{Q})
\end{aligned}$$

Case 2.2: $c = x$. Using $py = xy - xp$, after substituting in (7) and rearranging, we have

$$\begin{aligned}
\Lambda_{\omega,r} &= ee' - \omega(e, e') \quad + xy + xy - \omega(y, y) \quad + qc' - \omega(p, q) - xp - \omega(x, c') \quad + qx - rp - rq - xp \\
&\leq re + re' - \omega(e, e') \quad + xy + xy - \omega(y, y) \quad + qc' - \omega(x, q) - \omega(x, c') && (\triangle,\text{L})
\end{aligned}$$

Case 2.3: $c \notin \{x,y\}$. Without loss of generality, we can assume that $c = z$. The case when $c' = t$ is symmetric to Case 1, so we can assume that $c' = z$ as well. Also, we can assume that $e = x$ and $e' = y$. Using $xy = xp + py$, after substituting and rearranging, we have

$$\begin{aligned}\Lambda_{\omega,r} = &\ -rp + py - rq + qz - \omega(z,z) \quad + xy + xy - \omega(y,y) \quad + qz - \omega(p,q) - xp - \omega(x,y)\\ \leq &\ ry + rz - \omega(y,z) \quad + xy + xy - \omega(y,y) \quad + qz - \omega(x,q) - \omega(x,z) \qquad (\triangle,\ \triangle,\ \mathrm{L},\ \mathrm{Q})\end{aligned}$$

Case 3: $b' \notin \{x,y\}$. Without loss of generality, $b' = z$. The case when $c = c'$ or c, c' are opposite are symmetric to Case 2 and Case 1. Taking symmetries into account, we only have three subcases.

Case 3.1: $c = x$ and $c' = t$. Since $qx = xy - yq$ and $qt = zt - qz$, after substituting and rearranging, we have

$$\begin{aligned}\Lambda_{\omega,r} = &\ ee' - \omega(e,e') \quad + yp + yx - \omega(x,t) \quad + zt - \omega(y,z) - \omega(p,q) - qy \quad + pz - rp - rq - qz\\ \leq &\ ee' - \omega(e,e') \quad + yp + yx - \omega(p,x) \quad + zt - \omega(y,z) - \omega(q,t) - qy \qquad (\triangle,\mathrm{Q})\\ \leq &\ re + re' - \omega(e,e') \quad + yp + yx - \omega(p,x) \quad + zt - \omega(y,z) - \omega(y,t) \qquad (\triangle,\mathrm{L})\end{aligned}$$

Case 3.2: $c = y$ and $c' = t$. Since $qt = zt - qz$, after substituting and rearranging, we have

$$\begin{aligned}\Lambda_{\omega,r} = &\ ee' - \omega(e,e') \quad + yp + yq - \omega(p,q) \quad + zt - \omega(y,z) - \omega(y,t) \quad + pz - rp - rq - qz\\ \leq &\ re + re' - \omega(e,e') \quad + yp + yq - \omega(p,q) \quad + zt - \omega(y,z) - \omega(y,t) \qquad (\triangle,\triangle)\end{aligned}$$

Case 3.3: $c = y$ and $c' = z$. Without loss of generality, $e = x$ and $e' = y$. Then

$$\begin{aligned}\Lambda_{\omega,r} = &\ -rp + py - rq + qz - \omega(y,z) \quad + yq + yx - \omega(x,y) \quad + pz - \omega(p,q) - \omega(y,z)\\ \leq &\ ry + rz - \omega(y,z) \quad + yq + yx - \omega(q,x) \quad + pz - \omega(y,p) - \omega(y,z) \qquad (\triangle,\ \triangle,\mathrm{Q})\end{aligned}$$

In each case we obtain an expression that is an instance of $\Phi_{\omega,r}$ for some choice of parameters. The proof is now complete. □

Lemma 6 *Let ω be a work function with last request r in the Manhattan plane. Then $\Gamma_{\omega,r} \leq \Psi_{\omega,r}$.*

Proof: Pick points p, b, b', q, d, d', f such that

$$\Gamma_{\omega,r} \quad = \quad -rp + pb + pb' - \omega(b,b') + rq + dd' - \omega(q,d) - \omega(q,d') + qf - \omega(p,f) \qquad (8)$$

Let R be a rectangle with corners x, z, y, t (ordered clockwise) that contains all points in formula (8). By an argument identical to the one in the proof of Lemma 5, we can assume that $b, b', f, d, d' \in \{x,y,z,t\}$ and $\{d,d'\} = \{x,y\}$ or $\{z,t\}$. In the proof we use the symmetries between b, b', and between d, d', to reduce the number of cases to consider. Our goal, in each case, is to bound formula (8) above by an expression that is an instance of $\Psi_{\omega,r}$ for some choice of parameters. One case (Case 3.1) is exceptional; instead of one instance of $\Psi_{\omega,r}$, we obtain an average value of two instances.

Case 1: $f \in \{d,d'\}$. Without loss of generality, $f = d = x$ and $d' = y$. After substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = &\ -rp + pb + rq - \omega(b,b') \quad + xq + xy - \omega(q,y) \quad + b'p - \omega(q,x) - \omega(x,p)\\ \leq &\ rb + rq - \omega(b,q) \quad + xq + xy - \omega(q,y) \quad + b'p - \omega(x,b') - \omega(x,p) \qquad (\triangle,\mathrm{Q})\end{aligned}$$

Case 2: b and b' are opposite. Without loss of generality, $b = x$ and $b' = y$. Since $px + py = xy = rx + ry$, after substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & rx + ry - \omega(x,y) \quad + qf + rq - rp - \omega(p,f) \quad + dd' - \omega(q,d) - \omega(q,d') \\ \leq {} & rx + ry - \omega(x,y) \quad + qf + qp - \omega(p,f) \quad + dd' - \omega(q,d) - \omega(q,d') && (\triangle)\end{aligned}$$

Case 3: $b = b'$. Without loss of generality, $b = b' = y$. Since $xy = xp + py$, by the triangle inequality and the Lipschitz condition, $-rp + 2 \cdot py - \omega(p,f) \leq -rx + 2 \cdot xy - \omega(x,f)$. Therefore, we may assume that $p = x$. Then

$$\Gamma_{\omega,r} \quad = \quad -rx + 2 \cdot xy - \omega(y,y) + rq + dd' - \omega(q,d) - \omega(q,d') + qf - \omega(x,f)$$

There are three subcases.

Case 3.1: $f = x$. After substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & \tfrac{1}{2} \cdot \big[-rx + xy + rq - \omega(y,y) \quad + yx + yx - \omega(x,x) \quad + dd' - \omega(q,d) - \omega(q,d') \\ & - rx + xy + rq - \omega(y,y) \quad + qx + qx - \omega(x,x) \quad + dd' - \omega(q,d) - \omega(q,d') \big] \\ \leq {} & \tfrac{1}{2} \cdot \big[ry + rq - \omega(y,q) \quad + yx + yx - \omega(x,x) \quad + dd' - \omega(y,d) - \omega(y,d') \big] && (\triangle,\mathrm{Q}) \\ & + \tfrac{1}{2} \cdot \big[ry + rq - \omega(y,q) \quad + qx + qx - \omega(x,x) \quad + dd' - \omega(q,d) - \omega(q,d') \big] && (\triangle,\mathrm{Q})\end{aligned}$$

Case 3.2: $f = y$. Note that $qy = xy - qx$. If $\omega(x,y) + \omega(q,d) \geq \omega(x,d) + \omega(y,q)$, after substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & -rx + xy + rq - \omega(x,y) \quad + xy + xy - \omega(y,y) \quad + dd' - \omega(q,d) - qx - \omega(q,d') \\ \leq {} & ry + rq - \omega(y,q) \quad + xy + xy - \omega(y,y) \quad + dd' - \omega(x,d) - \omega(x,d') && (\triangle,\mathrm{L},\mathrm{Q})\end{aligned}$$

Otherwise, by quasiconvexity, $\omega(x,y) + \omega(q,d) \geq \omega(x,q) + \omega(y,d)$. After substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & -rx + xy + rq - \omega(y,y) \quad + yx + yx - \omega(x,y) - qx \quad + dd' - \omega(q,d) - \omega(q,d') \\ \leq {} & ry + rq - \omega(y,y) \quad + yx + yx - \omega(x,q) - qx \quad + dd' - \omega(y,d) - \omega(q,d') && (\triangle,\mathrm{Q}) \\ \leq {} & ry + rq - \omega(y,q) \quad + yx + yx - \omega(x,x) \quad + dd' - \omega(y,d) - \omega(y,d') && (\mathrm{L},\mathrm{Q})\end{aligned}$$

Case 3.3: $f \notin \{x,y\}$. Without loss of generality, $f = z$. By the symmetry of d and d', there are only two subcases.

Case 3.3.1: $d = x$ and $d' = y$. After substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & -rx + xy + rq - \omega(y,q) \quad + xy + xy - \omega(y,y) \quad + qz - \omega(x,q) - \omega(x,z) \\ \leq {} & ry + rq - \omega(y,q) \quad + xy + xy - \omega(y,y) \quad + qz - \omega(x,q) - \omega(x,z) && (\triangle)\end{aligned}$$

Case 3.3.2: $d = z$ and $d' = t$. Then $qz = zt - qt$. After substituting and rearranging, we have

$$\begin{aligned}\Gamma_{\omega,r} = {} & -rx + xy + rq - \omega(y,y) \quad + zt + zt - qt - \omega(q,t) \quad + xy - \omega(z,x) - \omega(q,z) \\ \leq {} & ry + rq - \omega(y,q) \quad + zt + zt - \omega(t,t) \quad + xy - \omega(z,x) - \omega(z,y) && (\triangle,\mathrm{L},\mathrm{Q})\end{aligned}$$

Case 4: $b \neq b'$, and b, b' are not opposite. Without loss of generality, $b = y$ and $b' = z$. By the symmetry of d, d', and the symmetry of b, b', we can assume that $d = x$ and $d' = y$. Then

$$\Gamma_{\omega,r} = -rp + py + pz - \omega(y,z) + rq + xy - \omega(q,x) - \omega(q,y) + qf - \omega(p,f)$$

The case where $f \in \{d, d'\} = \{x, y\}$ is covered by Case 1. So we only have two subcases.

Case 4.1: $f = t$. Then $qt = zt - qz$. After substituting and rearranging, we have

$$\begin{aligned} \Gamma_{\omega,r} &= -rp + py + rq - \omega(y,q) \quad + zt + zp - \omega(t,p) \quad + xy - qz - \omega(q,x) - \omega(z,y) \\ &\leq ry + rq - \omega(y,q) \quad + zt + zp - \omega(t,p) \quad + xy - \omega(z,x) - \omega(z,y) \qquad (\triangle,\mathrm{L}) \end{aligned}$$

Case 4.2: $f = z$. If $\omega(q,x) + \omega(p,z) \geq \omega(q,z) + \omega(p,x)$ then, after substituting and rearranging, we have

$$\begin{aligned} \Gamma_{\omega,r} &= rq - rp + pz - \omega(q,x) \quad + yp + yx - \omega(p,z) \quad + qz - \omega(y,q) - \omega(y,z) \\ &\leq rq + rz - \omega(q,z) \quad + yp + yx - \omega(p,x) \quad + qz - \omega(y,q) - \omega(y,z) \qquad (\triangle,\mathrm{Q}) \end{aligned}$$

Otherwise, by quasiconvexity, $\omega(q,x) + \omega(p,z) \geq \omega(p,q) + \omega(z,x)$, so

$$\begin{aligned} \Gamma_{\omega,r} &= rq - rp + py - \omega(y,q) \quad + zp + zq - \omega(p,z) \quad + xy - \omega(q,x) - \omega(z,y) \\ &\leq ry + rq - \omega(y,q) \quad + zp + zq - \omega(p,q) \quad + xy - \omega(z,x) - \omega(z,y) \qquad (\triangle,\mathrm{Q}) \end{aligned}$$

In each case except Case 3.1, we obtain an expression that is an instance of $\Phi_{\omega,r}$ for some choice of parameters. In Case 3.1, the final expression is an average of two instances of $\Phi_{\omega,r}$. The proof is now complete. □

Corollary 2 *There is a $3\sqrt{2}$-competitive algorithm for 3 servers in the Euclidean plane.*

Proof: Write $\|x,y\|_2$ and $\|x,y\|_1$ for the Euclidean and the city-block metric, respectively. Then $\frac{1}{\sqrt{2}}\|x,y\|_1 \leq \|x,y\|_2 \leq \|x,y\|_1$, for any two points $x, y \in \mathbf{R}^2$. For any request sequence in the plane, pretend that the metric is the city-block metric and follow WFA. The algorithm's cost in $\mathbf{R}^2_2$ is at most its cost in $\mathbf{R}^2_1$. The optimal cost in $\mathbf{R}^2_1$ is at most the optimal cost in $\mathbf{R}^2_2$ times $\sqrt{2}$. The result follows immediately from Theorem 2. □

5 Final Comments

We have proved that WFA is 3-competitive for 3 servers in the Manhattan plane. This immediately implies that WFA is also 3-competitive in $\mathbf{R}^2_\infty$, the plane with the supnorm metric, because $\mathbf{R}^2_\infty$ and $\mathbf{R}^2_1$ are isometric. WFA is also 3-competitive for 3 servers in any 6-point space. The result is obtained by proving that $\Lambda_{\omega,r} \leq \Psi_{\omega,r}$ and $\Gamma_{\omega,r} \leq \Psi_{\omega,r}$ for any space with at most 6 points, and then applying Theorem 1. The needed proofs, not given in this paper, involve a number of cases, and are quite similar to the proofs of Lemmas 5 and 6.

We believe that our method can be used to prove that WFA is 3-competitive in other metric spaces of interest. We conjecture that the inequalities $\Lambda, \Gamma \leq \Psi$ are true on the circle. We also conjecture that the same method will work in $\mathbf{R}^2_2$ (improving the ratio from Corollary 2 to 3).

Theorem 1 does not apply to arbitrary metric spaces. We have an example of a metric space M and a function ω on configurations in M which is quasiconvex and has the Lipschitz property, for which Theorem 1 fails, namely $\widehat{\Psi}_{\omega,r} > \Psi_{\omega,r}$. Nevertheless, by further pursuit of our approach and study of adversary sequences that generalize the lazy adversary, we hope to obtain a function that is a potential for WFA in an arbitrary metric space.

References

[1] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[2] William R. Burley. Traversing layered graphs using the work function algorithm. *Journal of Algorithms*, 20:479–511, 1996.

[3] Marek Chrobak, Howard Karloff, Tom H. Payne, and Sundar Vishwanathan. New results on server problems. *SIAM Journal on Discrete Mathematics*, 4:172–181, 1991.

[4] Marek Chrobak and Lawrence L. Larmore. An optimal online algorithm for k servers on trees. *SIAM Journal on Computing*, 20:144–148, 1991.

[5] Marek Chrobak and Lawrence L. Larmore. The server problem and on-line games. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 7, pages 11–64, 1992.

[6] Marek Chrobak and Lawrence L. Larmore. Metrical task systems, the server problem, and the work function algorithm. In *Online Algorithms: State of the Art*, pages 74–94. Springer-Verlag, 1998.

[7] Marek Chrobak, Lawrence L. Larmore, Nick Reingold, and Jeffery Westbrook. Page migration algorithms using work functions. *Journal of Algorithms*, 24(1):124–157, 1997.

[8] Elias Koutsoupias and Christos Papadimitriou. On the k-server conjecture. In *Proc. 26th Symp. Theory of Computing*, pages 507–511, 1994.

[9] Elias Koutsoupias and Christos Papadimitriou. On the k-server conjecture. *Journal of the ACM*, 42:971–983, 1995.

[10] Elias Koutsoupias and Christos Papadimitriou. The 2-evader problem. *Information Processing Letters*, 57:249–252, 1996.

[11] Mark Manasse, Lyle A. McGeoch, and Daniel Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11:208–230, 1990.