SIAM J. COMPUT.
Vol. 43, No. 1, pp. 96–125

THE GENERALIZED WORK FUNCTION ALGORITHM IS COMPETITIVE FOR THE GENERALIZED 2-SERVER PROBLEM*

RENÉ SITTERS†

Abstract. The generalized 2-server problem is an online optimization problem where a sequence of requests has to be served at minimal cost. Requests arrive one by one and need to be served instantly by at least one of two servers. We consider the general model where the cost function of the two servers may be different. Formally, each server moves in its own metric space and a request consists of one point in each metric space. It is served by moving one of the two servers to its request point. Requests have to be served without knowledge of future requests. The objective is to minimize the total traveled distance. The special case where both servers move on the real line is known as the CNN problem. We show that the generalized work function algorithm, WFA_λ, is constant competitive for the generalized 2-server problem. Further, we give an outline for a possible extension to $k \geqslant 2$ servers and discuss the applicability of our techniques and of the work function algorithm in general. We conclude with a discussion on several open problems in online optimization.

Key words. competitive analysis, k-server problem, online algorithms, work function algorithm, CNN problem, metrical service system

AMS subject classifications. 68W27, 68Q25

DOI. 10.1137/120885309

1. Introduction.

The work function algorithm is a generic algorithm for online optimization problems. For many problems, it gives the optimal competitive ratio or it is conjectured to be optimal. For example, it has the best known ratio of $2k-1$ for the k-server problem [25], which is probably the most appealing and well-studied problem in online optimization, and the work function algorithm is conjectured to have an optimal ratio of k. There are many papers that deal with this classical work function algorithm. More powerful, but less well known is the *generalized* work function algorithm, WFA_λ, which is the standard work function algorithm with an additional parameter λ. A result by Burley [9] shows that the generalized algorithm can indeed be strictly more powerful than the standard work function algorithm.

The (generalized) work function algorithm may be computationally expensive and pretty hard to analyze, but things can be much better for special cases. For example, the simple doubling algorithm for the cow path problem is mimed by the generalized work function algorithm $\mathrm{WFA}_{0.5}$. Another example is the (optimal) move-to-front algorithm for the list update problem which can be seen as the work function algorithm WFA_1. The running time of the work function algorithm very much depends on the complexity of computing offline solutions. For example, the work function algorithm for traversing layered graphs can be implemented in linear time while its analysis is quite involved [9]. Further, the performance of the work function algorithm may be much better in practice than what is guaranteed in theory (see, for example, [6]). For some problems, the work function algorithm is optimal but there are more efficient alternatives. For example, it is k-competitive for weighted caching [5] but the elegant double coverage (DC) algorithm [11] has the same optimal ratio. However, the DC

*Received by the editors July 19, 2012; accepted for publication (in revised form) November 21, 2013; published electronically January 28, 2014.
http://www.siam.org/journals/sicomp/43-1/88530.html

†VU University Amsterdam, the Netherlands (r.a.sitters@vu.nl), and CWI Amsterdam, the Netherlands (sitters@cwi.nl).

algorithm is not extendable to arbitrary metric spaces. For some hard problems, the work function algorithm is basically the only algorithm known. Examples are the (deterministic) k-server problem and the generalized 2-server problem that we discuss in this paper.

We say that an algorithm ALG for an online minimization problem is *c-competitive* ($c \geqslant 1$) if there is a constant c_0 such that for every instance $\mathcal{I}$ of the problem, the algorithms cost $\text{ALG}(\mathcal{I})$ and the optimal cost $\text{OPT}(\mathcal{I})$ satisfy

$$\text{ALG}(\mathcal{I}) \leqslant c \cdot \text{OPT}(\mathcal{I}) + c_0.$$

The competitive ratio of the algorithm is the infimum over c such that ALG is c-competitive. The competitive ratio of the online minimization problem is the infimum over c such that there is a c-competitive algorithm.

In the *generalized 2-server problem* we are given a server, whom we will call the $\mathbb{X}$-server, moving in a symmetric metric space $\mathbb{X}$, and a server, the $\mathbb{Y}$-server, moving in a symmetric metric space $\mathbb{Y}$. A starting point $(\mathcal{O}^{\mathbb{X}}, \mathcal{O}^{\mathbb{Y}}) \in \mathbb{X} \times \mathbb{Y}$ is given and requests $(x, y) \in \mathbb{X} \times \mathbb{Y}$ are presented online one by one. Requests are served by moving one of the servers to the corresponding point in its metric space and the choice of which server to move is made without knowledge of the future requests. The objective is to minimize the sum of the distances traveled by the two servers. The special case $\mathbb{X} = \mathbb{Y} = \mathbb{R}$ is known as the CNN problem [26, 27]. This problem can be seen as a single server moving in $\mathbb{R}^2$ with the L_1-norm and each request is a point in $\mathbb{R}^2$ which is served if the x- or y-coordinate of the server and request coincide.

Research on the CNN problem started more than ten years ago but despite the simplicity of the problem and its importance for the theory of online optimization, the problem is still not well understood. The CNN problem first appeared in a paper of Koutsoupias and Taylor [26, 27]. They conjectured that the generalized work function algorithm WFA_λ has a constant competitive ratio[1] for any $\lambda \in (0, 1)$. They also conjectured that the generalized work function algorithm is competitive for the generalized 2-server problem. In this paper we settle both conjectures. The constant that follows from our proof is large and we do not present an upper bound on its value. Hence, the gap between known lower and upper bound remains large. The first competitive algorithm was given in [33] and in its journal version [32]. The importance of the new result here is that we analyze the generalized work function algorithm which is applicable to *any* metrical service systems. Our techniques here are more involved than those in [32] and are interesting for online optimization in general. This is discussed in section 6.

As the name suggests, the generalized 2-server problem originates from the *classical* 2-server problem in which $\mathbb{X} = \mathbb{Y}$ and $x = y$ for every request, i.e., each request is a point in the metric space and we have to decide which server to move to the requested point. The k-server problem (with $k \geqslant 2$ servers) is one of the most studied problems in online optimization. A recent survey of the k-server problem is given by Koutsoupias [23]. The k-server problem on a uniform metric space is the *paging problem*. In the *weighted* k-server problem a weight is assigned to each server (of the classical problem) and the total cost is the weighted sum of the distances. The weighted k-server problem is a special case of the generalized k-server problem.

All online optimization problems mentioned in this article belong to the class of *metrical task systems* (for a definition see section 1.3 and [7]). Given multiple metrical task systems, the *sum problem* [27] is again a metrical task system and is

[1] The λ in [27] corresponds to $1/\lambda$ in our notation.

defined as follows: At each step we receive one request for each task system and we have to serve at least one of those requests. The CNN problem is the sum of two trivial problems: in both problems there is one server moving on the real line and each request consists of a single point. Koutsoupias and Taylor [27] emphasize the importance of the CNN problem: "*It is a very simple sum problem, which may act as a stepping stone towards building a robust (and less ad hoc) theory of online computation.*" Indeed, our techniques are useful for sum problems in general and we hope it leads to a better insight and hence further simplifications and generalizations in the theory of online computation.

1.1. Known competitive ratios. No *memoryless* (randomized) algorithm can have a finite competitive ratio for the CNN problem [15, 27, 35] while a finite ratio is possible if we are allowed to store the entire given sequence (or at least the current work function) [32, 33]. The algorithms in the latter two papers are complex and the ratio very high but they do apply to the generalized 2-server problem as well. See [10] for a review of [33]. For the classical k-server problem, the work function algorithm is $(2k-1)$-competitive for any metric space [25] and it is conjectured to be even k-competitive [28, 25]. This famous *k-server conjecture* was posed more than two decades ago and is still open. The competitive ratio of the *weighted* k-server problem is much higher. Fiat and Ricklin [17] prove that for any metric space with at least $k+1$ points there exists a set of weights such that the competitive ratio of any deterministic algorithm is at least $k^{\Omega(k)}$. Koutsoupias and Taylor [27] prove that any deterministic online algorithm for the weighted 2-server problem has a competitive ratio of at least $6+\sqrt{17} > 10.12$ even if the underlying metric space is the line and [15] shows that any memoryless randomized algorithm has unbounded competitive ratio in this case. These two lower bounds apply to the CNN problem as well since it contains the weighted 2-server problem on the line as a special case.

1.2. More special cases and variants. The *orthogonal* CNN problem [22] is the special case of the CNN problem in which each request either shares the x-coordinate or the y-coordinate with the previous request. Iwama and Yonezawa [22] give a 9-competitive algorithm and a lower bound of 3 is given in [1].

In the *continuous* CNN problem [1], there is one request which follows a continuous path in $\mathbb{R}^2$ and the online server must serve it continuously by aligning either horizontally or vertically. It generalizes the orthogonal version in the sense that any c-competitive algorithm for the continuous problem implies a c-competitive algorithm for the orthogonal problem. Augustine and Gravin [1] give a 6.46-competitive memoryless algorithm (improving the 9 from [22] mentioned above).

The *axis-bound* CNN problem was introduced by Iwama and Yonezawa [20, 21] and is the special case in which the server can only move on the x- and y-axes. They give an upper bound of 9 and a lower bound of $4+\sqrt{5}$. The lower bound was raised to 9 in [4]. That paper also gives an alternative 9-competitive algorithm by formulating it as a 2-*point request problem* [9]. Finally, the *box bound* CNN problem [3] is the restriction in which the server can move only on the boundary of a rectangle and requests are inside the rectangle. The problem can be transformed into the 4-point request problem [3]. An upper bound of 88.71 for the latter problem follows from the paper by Burley [9].

For the weighted 2-server problem, the only known competitive algorithm follows from the one for the generalized 2-server problem. For the special case of a *uniform* metric space (where all distances are 1), Chrobak and Sgall [15] prove that the work function algorithm is 5-competitive and that no better ratio is possible. They also

give a 5-competitive randomized, memoryless algorithm for uniform spaces, and a matching lower bound. Further, they consider a version of the problem in which a request specifies two points to be covered by the servers, and the algorithm must decide which server to move to which point. For this version, they show a 9-competitive algorithm and prove that no better ratio is possible. Finally, Verhoeven [35] shows that no memoryless randomized algorithm can be competitive for the CNN problem under an even weaker definition of memoryless than used in [15] and [27].

1.3. Metrical task systems and metrical service systems. Borodin, Linial, and Saks [7] introduced the problem of *metrical task systems*, a generalization of all online problems discussed here. Such system is a pair $\mathcal{S} = (\mathbb{M}, \mathcal{T})$, where $\mathbb{M}$ is a metric space and $\mathcal{T}$ a set of tasks. Each task $\tau \in \mathcal{T}$ is defined by a function $\tau : \mathbb{M} \to \mathbb{R}^+$ which gives for each $s \in \mathbb{M}$ the cost of serving the task while being in s. In an online instance, the tasks are given one by one and the objective is to minimize the total traveled distance (starting from given origin $\mathcal{O}$) plus the total service cost. The system is called *unrestricted* if $\mathcal{T}$ consists of all nonnegative real functions on $\mathbb{M}$. The authors of [7] show that the competitive ratio is exactly $2m-1$ for the unrestricted metrical task system on any metric space on m points.

A restricted model is that of *metrical service systems*, introduced in [12], [13], and [29]. (In [29] it is called *forcing task systems*.) Such a system is a pair $\mathcal{S} = (\mathbb{M}, \mathcal{R})$, where $\mathbb{M}$ is a metric space and $\mathcal{R}$ a set of requests where each request $r \in \mathcal{R}$ is a subset of $\mathbb{M}$. The system is called *unrestricted* if $\mathcal{R}$ consists of all subsets of $\mathbb{M}$. Metrical service systems correspond to metrical task systems for which $\tau : \mathbb{M} \to \{0, \infty\}$ for each task τ. Manasse, McGeoch, and Sleator [29] give an optimal $(m-1)$-competitive algorithm for the unrestricted metrical service system on any metric space on m points.

The generalized 2-server problem is a metrical service system: There is one server moving in the product space $\mathbb{M} = \mathbb{X} \times \mathbb{Y}$ and any pair $(x, y) \in \mathbb{X} \times \mathbb{Y}$ defines a request $r(x, y) = \{\{x\} \times \mathbb{Y}\} \cup \{\mathbb{X} \times \{y\}\} \subset \mathbb{M}$. The distance between points (x_1, y_1) and (x_2, y_2) in $\mathbb{X} \times \mathbb{Y}$ is $d((x_1, y_1), (x_2, y_2)) = d^{\mathbb{X}}(x_1, x_2) + d^{\mathbb{Y}}(y_1, y_2)$, where $d^{\mathbb{X}}$ and $d^{\mathbb{Y}}$ are the distance functions of the metric spaces $\mathbb{X}$ and $\mathbb{Y}$.

The work function algorithm is optimal for metrical task and metrical service systems in the sense that it is, respectively, $(2m-1)$- and $(m-1)$-competitive on any metric space of at most m points [14]. This is not of direct use for the CNN problem since the metric space, $\mathbb{R}^2$, has an unbounded number of points.

1.4. The work function algorithm: WFA$_\lambda$. The work function algorithm appeared for the first time in [12] but was discovered independently by others (see [25]). We use it here only for metrical service systems but it works the same for metrical task systems.

DEFINITION 1.1. *Given a metrical service system $\mathcal{S} = (\mathbb{M}, \mathcal{R})$ and origin $\mathcal{O} \in \mathbb{M}$, and given a request sequence σ, the work function $W_\sigma : \mathbb{M} \to \mathbb{R}^+$ is defined as follows. For any point $s \in \mathbb{M}$, $W_\sigma(s)$ is the length of the shortest path that starts in $\mathcal{O}$, ends in s, and serves σ.*

We assume here that the work function is well-defined (which is always true if the metric space is finite). Thus, we assume that for any $\sigma = r_1, \ldots, r_n$ and any point $s \in \mathbb{M}$ there are points $s_i \in r_i$ $(i = 1, \ldots, n)$ such that $d(\mathcal{O}, s_1) + d(s_1, s_2) + \cdots + d(s_{n-1}, s_n) + d(s_n, s) \leqslant d(\mathcal{O}, t_1) + d(t_1, t_2) + \cdots + d(t_{n-1}, t_n) + d(t_n, s)$ for any set of points $t_i \in r_i$ $(i = 1, \ldots, n)$. Clearly, the work function is well-defined for the generalized 2-server problem since we may assume that for each s_i, both its *coordinates* are from requests given so far. See [13] for a sufficient condition for the work function to be well-defined.

For a work function W_σ we say that point s is *dominated* by point t if $W_\sigma(s) = W_\sigma(t) + d(s,t)$. We define the *support* of W_σ as

$$\operatorname{supp}(W_\sigma) = \{s \in \mathbb{M} : s \text{ is not dominated by any other point}\}.$$

Let σr denote the sequence σ followed by request r. If $W_{\sigma r}$ is a well-defined work function then $\operatorname{supp}(W_{\sigma r}) \subseteq r$ since for any point $s \notin r$ there exists a point $t \in r$ such that $W_{\sigma r}(s) = W_{\sigma r}(t) + d(s,t)$. For more properties and a deeper analysis of the work function (algorithm) see, for example, [8], [9], and [23].

The generalized work function algorithm is a work-function-based algorithm parameterized by some constant $\lambda \in (0,1]$. We denote it by WFA_λ.

DEFINITION 1.2. *For any request sequence σ and any new request r, the* generalized work function algorithm WFA_λ *moves the server from the position s it had after serving σ to any point*

$$s' \in \operatorname*{Argmin}_{t \in \mathbb{M}} \{W_{\sigma r}(t) + \lambda d(s,t)\}. \tag{1.1}$$

This minimum may not be well-defined if the request r contains infinitely many points of the metric space. This is no problem for the generalized 2-server problem since the minimum is attained for some t with both coordinates of the given requests [32]. From (1.1), we see that

$$W_{\sigma r}(s') + \lambda d(s,s') \leqslant W_{\sigma r}(t) + \lambda d(s,t) \quad \text{for any point } t \in \mathbb{M}.$$

Using the triangle inequality, we get that for any $t \in \mathbb{M}$

$$W_{\sigma r}(s') \leqslant W_{\sigma r}(t) + \lambda(d(s,t) - d(s,s')) \leqslant W_{\sigma r}(t) + \lambda d(s',t). \tag{1.2}$$

If $\lambda < 1$ then (1.2) implies that s' is not dominated by any other point, whence $s' \in \operatorname{supp}(W_{\sigma r}) \subseteq r$. We see that if the moves of WFA_λ are well-defined then the choice of $\lambda < 1$ *ensures* that the point s' always serves the last request and we may *replace* $t \in \mathbb{M}$ by $t \in r$ in Definition 1.2.

For $\lambda = 0$, the generalized work function algorithm corresponds to the algorithm that always moves to the endpoint of an optimal solution, and for $\lambda = \infty$ it corresponds to the greedy algorithm (if we take $t \in r$ instead of $t \in \mathbb{M}$ in (1.1)). The standard work function algorithm has $\lambda = 1$ and was first used in [12] and has been studied extensively. The general form was defined in [12] as well but was used only shortly after in [13] where it is called the λ-cheap-and-lazy strategy. They show that WFA_λ with $\lambda = 1/3$ is optimal for the 2-point request problem. Burley generalized this and showed that WFA_λ is $O(k2^k)$-competitive for the *k-point request problem* (where λ depends on k).

In most papers, λ is placed before $W_{\sigma r}$ in (1.1) instead of before $d(s,t)$, as we do here. Also, sometimes α is used instead of λ. For example, Burley [9] uses $\alpha > 1$ and the following definition of the work function algorithm: $s' \in \operatorname{Argmin}_{t \in \mathbb{M}}\{\alpha W_{\sigma r}(t) + d(s,t)\}$. Replacing α by $1/\lambda$ matches our definition. Our choice was partly for an aesthetic reason: Now, the term λ appears much more often in the paper than the term $1/\lambda$. But also in the definitions of the *extended cost* and *slack function* (section 2), using $\lambda < 1$ seems the natural choice.

1.5. Paper outline and proof sketch. The main part of this paper is devoted to the CNN problem (Theorem 3.1). The generalization to arbitrary metric spaces (Theorem 4.1) is more complex and we do this in a separate section. The proof of

Theorem 3.1 is based on no less than 21 lemmas. To obtain a better insight into the relation between lemmas we mention after each lemma where it is used. The proof of Theorem 4.1 uses exactly the same lemmas (only some constants are different) and we indicate how to adjust the proofs of these lemmas.

Before giving a sketch of the proof, we give a brief outline of the paper. In section 2 we list some properties of the work function algorithm. These hold for any metrical service system and can be found in several other papers, e.g., [9, 14, 25]. Further, we introduce the closely related *slack function* and list some of its properties. In section 3 we present our potential function for the CNN problem together with some of its properties. Although the potential function is defined for the CNN problem, the theory in sections 3.1 and 3.2 applies to any metrical service system on $\mathbb{R}^2$. In section 3.3 we state some properties of the CNN problem which do not depend on the potential and in section 3.4 we put everything together and apply the potential to the CNN problem. In section 4 we show how to modify the proof for general metric spaces, i.e., we prove that the generalized work function algorithm is constant competitive for the generalized 2-server problem. In section 5 we give a sketch of a possible extension to higher dimensions. Finally, in section 6 we discuss several open problems in online optimization.

There are several reasons for giving a separate CNN proof. First, the reader has the option of just reading the CNN proof and skipping the more difficult general proof. Nevertheless, we believe that the generalization is relatively easy to digest once the reader has worked through the CNN proof and it may be even easier this way than when we would present only the general proof. One reason is that the CNN problem can be seen as moving points in the Euclidean plane which makes the proof easier to visualize than the proof for the general case.

Our potential function has a long description and may seem unintuitive at first. It is a linear combination of two functions: $\mathcal{F}$ and $\mathcal{G}$. Function $\mathcal{F}$ is a special case of the potential function that was used in [32] to give the first constant competitive algorithm for the generalized 2-server problem. When we use only $\mathcal{F}$ as our potential function and follow the line of proof that we use here, then the analysis fails. Taking $\mathcal{G}$ as potential function does not work either. However, the two functions are in a way complementary and if we take a linear combination of the two functions then the proof goes through.

Next, we give a very short technical sketch of the proof, which applies to both the CNN problem and the general problem. This part can be ignored but it may be very helpful for readers that are familiar with analysis of the work function algorithm. Definitions and formulas given here are presented in more detail later.

The potential function Φ_σ assigns a real value to each request sequence σ. It has the following form:

$$\Phi_\sigma = (1-\gamma) \min_{s_1,s_2,s_3 \in \mathbb{M}} \mathcal{F}_\sigma(s_1,s_2,s_3) + \gamma \min_{s_1,s_2,s_3 \in \mathbb{M}} \mathcal{G}_\sigma(s_1,s_2,s_3).$$

The functions $\mathcal{F}_\sigma : \mathbb{M}^3 \to \mathbb{R}$ and $\mathcal{G}_\sigma : \mathbb{M}^3 \to \mathbb{R}$ depend on the sequence σ. Further, $\mathbb{M} = \mathbb{X} \times \mathbb{Y}$ and $\gamma \in (0,1)$ is a constant. The initial value is zero and in general it is upper bounded by the optimal value of the sequence so far, i.e., $\Phi_\sigma \leqslant \textsc{Opt}_\sigma$. We consider two arbitrary, subsequent requests r' and r'' and show that the increase $\Phi'' - \Phi'$ of the potential function for the new request r'' is at least some constant c times the so called *extended cost* for r'', denoted by $\nabla_{r''}$ (Definition 2.1):

$$\Phi'' - \Phi' \geqslant c\nabla_{r''}. \tag{1.3}$$

Then, taking the sum over all requests in the entire sequence ρ of the instance,[2] we find that the total increase in the potential function is at least c times the total extended cost, denoted by ∇_ρ. We get $\nabla_\rho \leqslant (1/c)\Phi_\rho \leqslant (1/c)\textsc{Opt}_\rho$. Proof of competitiveness then follows directly from Lemma 2.2.

We now give some more details of (1.3). Let σ' be a request sequence which ends with r'. It is followed by r'' and we denote $\sigma'' = \sigma' r''$. Let s_1, s_2, s_3 be a minimizer of $\mathcal{F}_{\sigma''}$. By construction of $\mathcal{F}$, all three points will serve the last request r''. (The same holds for $\mathcal{G}_{\sigma''}$.) We distinguish between *Case A*: $|\{s_1, s_2, s_3\}| \leqslant 2$, and *Case B*: $|\{s_1, s_2, s_3\}| = 3$. In the following, $c_1, c_2, c_3, c_4 > 0$ are specific constants depending on λ. For Case A, we show that

$$\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_1 \nabla_{r''} \quad \text{and} \quad \min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant 0,$$

where $\nabla_{r''}$ is the extended cost for r'' w.r.t. σ'. Hence, the increase for Φ is at least $(1-\gamma)c_1\nabla_{r''}$ and (1.3) holds with $c = (1-\gamma)c_1$. Note that, if we were always in Case A then there would be no need for function $\mathcal{G}$. The more difficult part of the proof is Case B. In that case, it is easy to show that the increase for the minimum of function $\mathcal{F}$ is

$$\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_2 \min\{\partial x, \partial y\},$$

where $\partial x = d^{\mathbb{X}}(x', x'')$ and $\partial y = d^{\mathbb{Y}}(y', y'')$. This is not enough to prove (1.3) if $\min\{\partial x, \partial y\} \ll \nabla_{r''}$. So, let us consider the extreme case that $\min\{\partial x, \partial y\} = 0$. Intuitively, we should be fine if we can handle this. The function $\mathcal{G}$ was designed exactly for this case. More precisely, we show that if $\min\{\partial x, \partial y\} = 0$ then

$$\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant c_3 \nabla_{r''}. \tag{1.4}$$

Hence, the increase for Φ is at least $\gamma c_3 \nabla_{r''}$ and (1.3) holds with $c = \gamma c_3$. In general, we prove that

$$\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant c_3 \nabla_{r''} - c_4 \min\{\partial x, \partial y\}.$$

The increase in Φ becomes at least

$$\gamma c_3 \nabla_{r''} + ((1-\gamma)c_2 - \gamma c_4) \min\{\partial x, \partial y\},$$

which becomes at least $\gamma c_3 \nabla_{r''}$ by choosing $\gamma < c_2/(c_2 + c_4)$. Again, (1.3) applies with $c = \gamma c_3$.

Finally, a few words on how to prove (1.4). Let (s_1, s_2, s_3) be a minimizer of $\mathcal{G}_{\sigma''}$. Remember that, for function $\mathcal{F}$, we argued that we are fine if the cardinality of $\{s_1, s_2, s_3\}$ is at most 2. But for function $\mathcal{G}$ we can enforce this situation by using the fact that the two subsequent requests are aligned. Say that $\partial y = 0$. It will turn out that the only interesting case is when s_1, s_2, s_3 are all on the line $y = y' = y''$. (Since otherwise, (1.4) will follow almost directly.) For this situation, we prove (see Lemma 3.6) that one of the three points is redundant in the sense that there are points $u_1, u_2 \in \{s_1, s_2, s_3\}$ such that $\mathcal{G}_{\sigma''}(u_1, u_2, u_2) = \mathcal{G}_{\sigma''}(s_1, s_2, s_3)$. Then, (1.4) is proven in a similar way as is done for $\mathcal{F}$ in Case A.

Summarizing, function $\mathcal{F}$ works fine as a potential function on its own, except for the case that $\min\{\partial x, \partial y\} \approx 0$. In a way, the difficult part is reduced to an

[2] In this paper, ρ always refers to the entire given sequence, i.e., no requests are given after ρ. We mainly use σ otherwise.

easier situation where the two subsequent requests are on a line and we designed a potential function $\mathcal{G}$ that takes care of this situation. In other words, the difficult case in the proof is reduced to a problem of lower dimension. This insight led to the generalization to higher dimensions discussed in section 5.

2. Preliminaries. This section applies to any metrical service system. For the analysis of the generalized work function algorithm we make extensive use of two concepts: *extended cost* and *slack*. The first is an amortized cost of the (general) work function algorithm. It was introduced together with the work function algorithm in [12] (where it is called pseudocost) and has been used in every analysis of the work function algorithm. The slack function was defined by Burley [9] and was also used in [32]. Its definition comes naturally with that of extended cost and its use enhances the analysis.

2.1. The extended cost.

DEFINITION 2.1. *For request sequence σ and request r, the extended cost for r is*

$$\nabla_r(W_\sigma) = \max_{s\in\mathbb{M}} \min_{t\in r} \left[W_\sigma(t) + \lambda d(s,t) - W_\sigma(s)\right].$$

For $\rho = r_1 r_2 \cdots r_n$, we define the total extended cost as $\nabla_\rho = \sum_{i=1}^{n} \nabla_{r_i}(W_{r_1\cdots r_{i-1}})$.

The definition of extended cost matches that in [32] and matches the commonly used extended cost in the case $\lambda = 1$. It also matches the definition by Burley [9], although the notation is quite different. The intuition behind extended cost becomes clear from the following lemma and its proof.

LEMMA 2.2. *Let ∇_ρ be the total extended cost of sequence ρ. If $\nabla_\rho \leqslant c\mathrm{OPT}_\rho$ for some constant c and any request sequence ρ then WFA_λ is $(c-1)/\lambda$-competitive. (Used in proof of Theorem 3.1.)*

Proof. Assume the online server is in point s' after it served the initial sequence σ and moves to t' to serve a new request r. Since we maximize over $s \in \mathbb{M}$ in Definition 2.1 we have

$$\begin{aligned}\nabla_r(W_\sigma) &\geqslant \min_{t\in r}\left[W_\sigma(t) + \lambda d(s',t) - W_\sigma(s')\right] \\ &= \min_{t\in r}\left[W_{\sigma r}(t) + \lambda d(s',t) - W_\sigma(s')\right].\end{aligned}$$

By the definition of WFA_λ, the minimum on the right side is attained for $t = t'$. Therefore,

$$\nabla_r(W_\sigma) \geqslant W_{\sigma r}(t') + \lambda d(s',t') - W_\sigma(s').$$

Rewriting we get

$$d(s',t') \leqslant \frac{1}{\lambda}\left(\nabla_r(W_\sigma) + W_\sigma(s') - W_{\sigma r}(t')\right).$$

This gives an upper bound for the cost of WFA_λ for serving some single request r. Let q be the point where the algorithm ends after serving ρ. Summing up over all requests in ρ we get that the total cost for WFA_λ is at most

$$\frac{1}{\lambda}(\nabla_\rho + W_\epsilon(\mathcal{O}) - W_\rho(q)) \leqslant \frac{1}{\lambda}(\nabla_\rho - \mathrm{OPT}_\rho) \leqslant \frac{1}{\lambda}(c-1)\mathrm{OPT}_\rho. \qquad \square$$

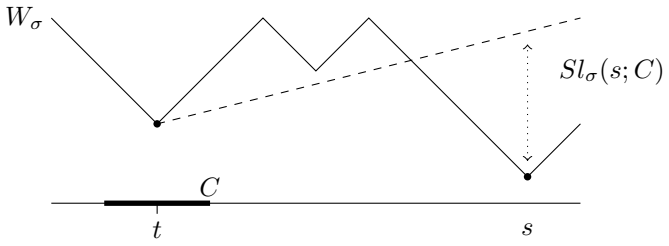

FIG. 2.1. *The slack of s with respect to line segment C is attained in $t \in C$.*

2.2. The slack function. We use the concept of the *slack* of a point relative to another point. Intuitively, the slack of a point s with respect to a point t is the amount that the work function value in s can increase before the generalized work function algorithm moves from s to t. More precisely, the generalized work function algorithm, being in point s after serving sequence σ, moves away from s after a new request r is given if there is a point t such that $W_{\sigma r}(t) + \lambda d(s,t) \leqslant W_{\sigma r}(s)$. The slack is the difference between the left and right sides of this inequality. More generally, we define the slack of a point with respect to a subset of $\mathbb{M}$. See Figure 2.1.

DEFINITION 2.3. *Given a request sequence σ, we define the slack of a point $s \in \mathbb{M}$ with respect to a (possibly infinite) set of points $C \subseteq \mathbb{M}$ as*

$$Sl_\sigma(s;C) = \min_{t \in C}\{W_\sigma(t) + \lambda d(s,t)\} - W_\sigma(s).$$

If C contains only one point t then we simply write $Sl_\sigma(s;t)$ instead of $Sl_\sigma(s;\{t\})$. If C is a closed subset of $\mathbb{M}$ then the minimum is well-defined.

Using the slack function makes the proof shorter and more intuitive. For example, we can rewrite the extended cost, $\nabla_r(W_\sigma)$, for request sequence σ and new request r in terms of the slack function.

$$\nabla_r(W_\sigma) = \max_{s \in \mathbb{M}}\{\min_{t \in r}\{W_\sigma(t) + \lambda d(s,t)\} - W_\sigma(s)\} = \max_{s \in \mathbb{M}} Sl_\sigma(s;r). \tag{2.1}$$

In the remainder of this section, we list some properties of the slack function. The first property (2.2) follows directly from its definition and from the work function being Lipschitz continuous with constant 1. For any $s, t \in \mathbb{M}$

$$Sl_\sigma(s;t) \leqslant (1+\lambda) d(s,t). \tag{2.2}$$

The next lemma also follows directly from the definition of slack.

LEMMA 2.4. *If $C_1 \subseteq C_2 \subseteq \mathbb{M}$ then for any $s \in \mathbb{M}$ we have $Sl_\sigma(s;C_1) \geqslant Sl_\sigma(s;C_2)$. (Used in proof of many lemmas.)*

The lemma above is mostly used in the form $t \in C \subseteq \mathbb{M}$ implies $Sl_\sigma(s;t) \geqslant Sl_\sigma(s;C)$.

LEMMA 2.5. *For any set of points $C \subset \mathbb{M}$ there is a point $s \in C$ such that $Sl_\sigma(s;C) = 0$. (Used in proof of Lemma 3.4.)*

Proof. Let $s \in \operatorname{Argmin}\{W_\sigma(t) \mid t \in C\}$. Then, for any $t \in C$

$$Sl_\sigma(s;t) = W_\sigma(t) + \lambda d(s,t) - W_\sigma(s) \geqslant 0.$$

Clearly, $Sl_\sigma(s;s) = 0$. Hence, $Sl_\sigma(s;C) = \min_{t \in C} Sl_\sigma(s;t) = 0$. ∎

The next lemma shows a transitivity property of slack.

LEMMA 2.6. *Let $s_1, s_2, s_3 \in \mathbb{M}$ such that $d(s_1, s_2) + d(s_2, s_3) = d(s_1, s_3)$. Then $Sl_\sigma(s_3; s_1) = Sl_\sigma(s_3; s_2) + Sl_\sigma(s_2; s_1)$. (Used in proof of Lemma* 3.6.*)*

Proof.

$$\begin{aligned} Sl_\sigma(s_3; s_1) &= W_\sigma(s_1) + \lambda d(s_1, s_3) - W_\sigma(s_3) \\ &= W_\sigma(s_1) + \lambda(d(s_1, s_2) + d(s_2, s_3)) - W_\sigma(s_3) \\ &= W_\sigma(s_1) + \lambda d(s_1, s_2) - W_\sigma(s_2) + W_\sigma(s_2) + \lambda d(s_2, s_3) - W_\sigma(s_3) \\ &= Sl_\sigma(s_2; s_1) + Sl_\sigma(s_3; s_2). \qquad \square \end{aligned}$$

The next lemma generalizes Lemma 2.4.

LEMMA 2.7. *Let $C_1, C_2 \subseteq \mathbb{M}$ and $\delta \in \mathbb{R}^+$.If for every point $u_1 \in C_1$ there is a point $u_2 \in C_2$ with $d(u_1, u_2) \leqslant \delta$, then for every $s \in \mathbb{M}$*

$$Sl_\sigma(s; C_1) \geqslant Sl_\sigma(s; C_2) - (1 + \lambda)\delta.$$

(Used in proof of Lemma 3.2.*)*

Proof. Let $u_1 \in C_1$ be such that $Sl_\sigma(s; C_1) = Sl_\sigma(s; u_1)$. There is a point $u_2 \in C_2$ such that $d(u_1, u_2) \leqslant \delta$.

$$\begin{aligned} &Sl_\sigma(s; C_1) - Sl_\sigma(s; C_2) \\ &\quad = Sl_\sigma(s; u_1) - Sl_\sigma(s; C_2) \\ &\quad \geqslant Sl_\sigma(s; u_1) - Sl_\sigma(s; u_2) \\ &\quad = W_\sigma(u_1) + \lambda d(u_1, s) - W_\sigma(s) - (W_\sigma(u_2) + \lambda d(u_2, s) - W_\sigma(s)) \\ &\quad = W_\sigma(u_1) - W_\sigma(u_2) + \lambda(d(u_1, s) - d(u_2, s)) \\ &\quad \geqslant W_\sigma(u_1) - W_\sigma(u_2) - \lambda d(u_1, u_2) \\ &\quad \geqslant -d(u_1, u_2) - \lambda d(u_1, u_2) \\ &\quad = -(1 + \lambda)\delta. \qquad \square \end{aligned}$$

LEMMA 2.8. *Let $s, t \in \mathbb{M}$ and $C \subset \mathbb{M}$. Then,*

(a) *$Sl_\sigma(t; C) \geqslant Sl_\sigma(s; C) - (1 + \lambda)d(s, t)$, and*

(b) *$Sl_\sigma(t; C) \geqslant Sl_\sigma(s; C) + (1 - \lambda)d(s, t)$, if t dominates s w.r.t. σ.*

(Follows from Lemma 2.4. *Used in proofs of Lemmas* 3.2, 3.10, *and* 3.14.*)*

Proof. Let $u \in C$ be such that $Sl_\sigma(t; C) = Sl_\sigma(t; u)$. Then,

$$\begin{aligned} Sl_\sigma(t; C) &= Sl_\sigma(t; u) \\ &= Sl_\sigma(s; u) - \lambda d(u, s) + \lambda d(u, t) + W_\sigma(s) - W_\sigma(t) \\ &\geqslant Sl_\sigma(s; u) - \lambda d(s, t) + W_\sigma(s) - W_\sigma(t) \\ &\geqslant Sl_\sigma(s; C) - \lambda d(s, t) + W_\sigma(s) - W_\sigma(t). \end{aligned}$$

The first inequality is given by the triangle inequality and the second by Lemma 2.4. In general, $W_\sigma(s) - W_\sigma(t) \geqslant -d(s, t)$, which implies (a). If t dominates s then we have the stronger bound $W_\sigma(s) - W_\sigma(t) = d(s, t)$. $\square$

3. The CNN problem. A simple example shows that the standard work function algorithm WFA_1 has unbounded competitive ratio for the CNN problem: Take $(0, 0)$ as the origin and consider the request sequence $(1, 2), (2, 2), (3, 2), \ldots (m, 2)$ for arbitrary m. The optimal solution moves from $(0, 0)$ to $(0, 2)$ but the work function algorithm follows the path $(0, 0), (1, 0), (2, 0), \ldots, (m, 0)$. (There are no draws.) The competitive ratio for this instance is $m/2$.

THEOREM 3.1. *The generalized work function algorithm WFA_λ is constant competitive for the* CNN *problem for any constant λ with $0 < \lambda < 1$.*

All the lemmas of the previous section apply to metrical service systems in general. In this section, we restrict ourselves to the CNN problem. It is convenient to insist

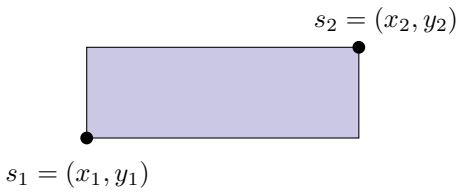

FIG. 3.1. *The shaded area is* $\textsc{Box}(s_1, s_2)$*, used in the potential function.*

on writing $\mathbb{M}$ for the metric space although we now have $\mathbb{M} = \mathbb{R}^2$. We make a subtle distinction between the *request point* $(x', y') \in \mathbb{R}^2$ and the corresponding *request* as defined by the metrical service system: $r(x', y') = \{\{(x, y) \in \mathbb{R}^2 \mid x = x' \text{ or } y = y'\}$.

3.1. The potential function. Our potential function is defined for any metrical service system on $\mathbb{R}^2$ but we only use it for the CNN problem.

One of the ingredients is the set $\textsc{Box}(s_1, s_2)$ (see Figure 3.1) defined as follows. Given points $x_1, x_2 \in \mathbb{R}$, we denote by $[x_1, x_2]$ the interval between x_1 and x_2 (we allow $x_2 < x_1$, i.e., $[x_2, x_1] = [x_1, x_2]$). Note that at this point we use the restriction to the real line since this is not well-defined for a general metric space. (In section 4, where the proof is generalized to arbitrary metric spaces we shall start from this point.)

Given points $s_1 = (x_1, y_1) \in \mathbb{M}$ and $s_2 = (x_2, y_2) \in \mathbb{M}$ we denote the set of points in the rectangle spanned by these points by

$$\textsc{Box}(s_1, s_2) = \{(x, y) \in \mathbb{M} \mid x \in [x_1, x_2] \text{ and } y \in [y_1, y_2]\}.$$

Let $0 < \alpha < 1/2$ and $0 < \gamma < 1$. We define the functions $\mathcal{F}_\sigma : \mathbb{M}^3 \to \mathbb{R}$ and $\mathcal{G}_\sigma : \mathbb{M}^3 \to \mathbb{R}$ as

$$\mathcal{F}_\sigma(s_1, s_2, s_3) = W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) - \alpha Sl_\sigma(s_3; \{s_1, s_2\}),$$
$$\mathcal{G}_\sigma(s_1, s_2, s_3) = W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) - \alpha Sl_\sigma(s_3; \textsc{Box}(s_1, s_2)).$$

The two functions only differ in the last term. The potential function Φ_σ is

$$\Phi_\sigma = (1 - \gamma) \min_{s_1, s_2, s_3 \in \mathbb{M}} \mathcal{F}_\sigma(s_1, s_2, s_3) + \gamma \min_{s_1, s_2, s_3 \in \mathbb{M}} \mathcal{G}_\sigma(s_1, s_2, s_3).$$

The numbers α and γ will depend only on λ and we fix their precise values later. It is good to mention here that the proof works for any small enough values of α and γ. More precisely, the proof works if we pick any α with $0 < \alpha \leqslant \alpha_0$ for some α_0 depending on λ and then pick any γ with $0 < \gamma \leqslant \gamma_0$ for some γ_0 depending on λ and α.

Comprehensive notation. To simplify the analysis we define one more function $\mathcal{H}_\sigma : \mathbb{M}^2 \to \mathbb{R}$. It corresponds to the first two terms of $\mathcal{F}_\sigma$ and $\mathcal{G}_\sigma$.

$$\mathcal{H}_\sigma(s_1, s_2) = W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) = \frac{1}{2} W_\sigma(s_1) + \frac{1}{2} W_\sigma(s_2) - \frac{\lambda}{2} d(s_1, s_2).$$

We can rewrite $\mathcal{F}_\sigma$ and $\mathcal{G}_\sigma$ as

$$\mathcal{F}_\sigma(s_1, s_2, s_3) = \mathcal{H}_\sigma(s_1, s_2) - \alpha Sl_\sigma(s_3; \{s_1, s_2\}),$$

$$\mathcal{G}_\sigma(s_1, s_2, s_3) = \mathcal{H}_\sigma(s_1, s_2) - \alpha Sl_\sigma(s_3; \textsc{Box}(s_1, s_2)).$$

For a request sequence σ, we denote $\min_{s_1,s_2,s_3\in\mathbb{M}} \mathcal{F}_\sigma(s_1, s_2, s_3)$ simply by $\min \mathcal{F}_\sigma$ and make a similar simplification of notation for $\mathcal{G}$ and $\mathcal{H}$. A shorter notation for the potential function becomes

$$\Phi_\sigma = (1-\gamma)\min \mathcal{F}_\sigma + \gamma \min \mathcal{G}_\sigma.$$

Note that $\mathcal{H}_\sigma$ is symmetric in s_1 and s_2 and, consequently, also $\mathcal{F}_\sigma$ and $\mathcal{G}_\sigma$ are symmetric in s_1 and s_2. This property is not essential but enhances the argumentation at some points.

3.2. Properties of the potential function. In this section we list some properties of the potential function Φ_σ which hold for any metrical service system on $\mathbb{M} = \mathbb{R}^2$ and arbitrary corresponding request sequence σ. In section 3.3, we restrict the analysis to the CNN problem.

The functions $\mathcal{F}_\sigma$ and $\mathcal{G}_\sigma$ are constructed such that in the minimum all three points s_1, s_2, s_3 are on the last request, at least if α is small enough. This is stated in Lemma 3.3. The next lemma is preliminary for this lemma and several others.

LEMMA 3.2. *Let $t \in \mathbb{M}$ dominate $s \in \mathbb{M}$ (w.r.t. σ) and let $\delta = d(s,t)$. Then, for any $s_1, s_2, s_3 \in \mathbb{M}$,*

$$\begin{aligned}
&\text{(a)} \quad \mathcal{F}_\sigma(s_1, s_2, s) - \mathcal{F}_\sigma(s_1, s_2, t) \geqslant \delta \cdot \alpha(1-\lambda),\\
&\text{(b)} \quad \mathcal{F}_\sigma(s_1, s, s_3) - \mathcal{F}_\sigma(s_1, t, s_3) \geqslant \delta \cdot \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right),\\
&\text{(c)} \quad \mathcal{F}_\sigma(s, s_2, s_3) - \mathcal{F}_\sigma(t, s_2, s_3) \geqslant \delta \cdot \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right),\\
&\text{(d)} \quad \mathcal{G}_\sigma(s_1, s_2, s) - \mathcal{G}_\sigma(s_1, s_2, t) \geqslant \delta \cdot \alpha(1-\lambda),\\
&\text{(e)} \quad \mathcal{G}_\sigma(s_1, s, s_3) - \mathcal{G}_\sigma(s_1, t, s_3) \geqslant \delta \cdot \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right),\\
&\text{(f)} \quad \mathcal{G}_\sigma(s, s_2, s_3) - \mathcal{G}_\sigma(t, s_2, s_3) \geqslant \delta \cdot \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right).
\end{aligned}$$

(Follows from Lemmas 2.7 and 2.8. *Used in proof of Lemmas* 3.3, 3.7, 3.12, 3.13, *and* 3.14.*)*

Proof. Statement (a) follows directly from Lemma 2.8(b) with $C = \{s_1, s_2\}$:

$$\mathcal{F}_\sigma(s_1, s_2, s) - \mathcal{F}_\sigma(s_1, s_2, t) = \alpha Sl_\sigma(t; \{s_1, s_2\}) - \alpha Sl_\sigma(s; \{s_1, s_2\}) \geqslant \alpha(1-\lambda)\delta.$$

The same holds for (d) but now with $C = \textsc{Box}(s_1, s_2)$. By symmetry of $\mathcal{F}$ and $\mathcal{G}$ in their first two arguments, it only remains to prove statements (b) and (e). We start with (b):

$$\begin{aligned}
&\mathcal{F}_\sigma(s_1, s, s_3) - \mathcal{F}_\sigma(s_1, t, s_3)\\
&\quad = \frac{1}{2}\left(Sl_\sigma(t; s_1) - Sl_\sigma(s; s_1)\right) + \alpha\left(Sl_\sigma(s_3; \{s_1, t\}) - Sl_\sigma(s_3; \{s_1, s\})\right).
\end{aligned} \tag{3.1}$$

For the first part of (3.1) we use Lemma 2.8(b):

$$Sl_\sigma(t; s_1) - Sl_\sigma(s; s_1) \geqslant (1-\lambda)\delta. \tag{3.2}$$

For the second part we apply Lemma 2.7 with $C_1 = \{s_1, t\}$ and $C_2 = \{s_1, s\}$. The

condition of Lemma 2.7 is satisfied for $\delta = d(s,t)$. We have

$$Sl_\sigma(s_3;\{s_1,t\}) - Sl_\sigma(s_3;\{s_1,s\}) \geqslant -(1+\lambda)\delta. \tag{3.3}$$

Combining (3.2) and (3.3) we get (b). The proof of (e) is similar. We apply (3.2) and Lemma 2.7 with $C_1 = \text{Box}(s_1,t)$ and $C_2 = \text{Box}(s_1,s)$. The condition of Lemma 2.7 is satisfied for $\delta = d(s,t)$:

$$\begin{aligned}
&\mathcal{G}_\sigma(s_1,s,s_3) - \mathcal{G}_\sigma(s_1,t,s_3)\\
&\quad= \frac{1}{2}\left(Sl_\sigma(t;s_1) - Sl_\sigma(s;s_1)\right) + \alpha\left(Sl_\sigma(s_3;\text{Box}(s_1,t)) - Sl_\sigma(s_3;\text{Box}(s_1,s))\right)\\
&\quad\geqslant \frac{1}{2}(1-\lambda)\delta - \alpha(1+\lambda)\delta. \qquad \square
\end{aligned}$$

Note that all the right-hand sides in Lemma 3.2 are strictly positive if $0 < \alpha < (1-\lambda)/(2(1+\lambda))$. We assume this from now on.

LEMMA 3.3. *If $\mathcal{F}_{\sigma r}$ or $\mathcal{G}_{\sigma r}$ is minimized in (s_1,s_2,s_3) then $s_1,s_2,s_3 \in r$. (Follows from Lemma* 3.2. *Used in proof of Lemmas* 3.13 *and* 3.14.*)*

Proof. Any point is dominated by a point in the last request. Take any triple of points in $\mathbb{M}$. If at least one of the points is not in r then Lemma 3.2 tells us that we can replace it by a point of the last request, r, such that the values of $\mathcal{F}_\sigma$ and $\mathcal{G}_\sigma$ become strictly smaller. $\square$

LEMMA 3.4. $\min \mathcal{F}_\sigma \leqslant \min \mathcal{G}_\sigma \leqslant \min \mathcal{H}_\sigma$. *(Follows from Lemmas* 2.4 *and* 2.5. *Used in proof of Lemma* 3.15.*)*

Proof. For any $s_1,s_2 \in \mathbb{M}$ there is a point s_3 such that $Sl_\sigma(s_3;\text{Box}(s_1,s_2)) = 0$. (See Lemma 2.5.) Hence, $\min \mathcal{G}_\sigma \leqslant \min \mathcal{H}_\sigma$.

For any $s_1,s_2,s_3 \in \mathbb{M}$ we have $Sl_\sigma(s_3;\{s_1,s_2\}) \geqslant Sl_\sigma(s_3;\text{Box}(s_1,s_2))$ since $\{s_1,s_2\} \subseteq \text{Box}(s_1,s_2)$. (See Lemma 2.4.) Therefore, $\min \mathcal{F}_\sigma \leqslant \min \mathcal{G}_\sigma$. $\square$

The two inequalities of Lemma 3.4 are only strict if the three points for which the minimum of $\mathcal{F}_\sigma$ or $\mathcal{G}_\sigma$ is attained are in a way different enough. For example, the next lemma implies that if the minimum of $\mathcal{F}_\sigma$ is attained for (s_1,s_2,s_3) but they are not all different, then both inequalities are equalities. For $\mathcal{G}_\sigma$ a stronger property holds. If s_1,s_2,s_3 are all on a line then the second inequality is an equality.

LEMMA 3.5. *If $\{s_1,s_2,s_3\}$ has cardinality* 1 *or* 2 *then $\mathcal{H}_\sigma(u_1,u_2) \leqslant \mathcal{F}_\sigma(s_1,s_2,s_3)$ for some $u_1,u_2 \in \{s_1,s_2,s_3\}$. (Used in proof of Lemmas* 3.13 *and* 3.15.*)*

Proof. If $Sl_\sigma(s_3;\{s_1,s_2\}) \leqslant 0$ then $\mathcal{H}_\sigma(s_1,s_2) \leqslant \mathcal{F}_\sigma(s_1,s_2,s_3)$. So assume the opposite:

$$Sl_\sigma(s_3;\{s_1,s_2\}) > 0. \tag{3.4}$$

We cannot have $s_1 = s_3$ or $s_2 = s_3$, since this contradicts (3.4). Hence, we must have $s_1 = s_2$, which implies $Sl_\sigma(s_2;s_1) = 0$.

$$\begin{aligned}
\mathcal{F}_\sigma(s_1,s_2,s_3) &= W_\sigma(s_1) - \frac{1}{2}Sl_\sigma(s_2;s_1) - \alpha Sl_\sigma(s_3;\{s_1,s_2\})\\
&= W_\sigma(s_1) - \alpha Sl_\sigma(s_3;\{s_1,s_2\})\\
&> W_\sigma(s_1) - \frac{1}{2}Sl_\sigma(s_3;\{s_1,s_2\})\\
&= W_\sigma(s_1) - \frac{1}{2}Sl_\sigma(s_3;s_1)\\
&= \mathcal{H}_\sigma(s_1,s_3).
\end{aligned}$$

For the inequality we used (3.4) and $\alpha < 1/2$. $\square$

Lemma 3.5 applies also to $\mathcal{G}_\sigma$ instead of $\mathcal{F}_\sigma$ but we shall not use this. In addition, $\mathcal{G}_\sigma$ has the following property.

LEMMA 3.6. *If $s_1, s_2, s_3 \in \mathbb{M}$ all have the same x-coordinate or the same y-coordinate, then $\mathcal{H}_\sigma(u_1, u_2) \leqslant \mathcal{G}_\sigma(s_1, s_2, s_3)$ for some $u_1, u_2 \in \{s_1, s_2, s_3\}$. (Follows from Lemmas* 2.4 and 2.6. *Used in proof of Lemma* 3.14.*)*

Proof. We shall prove something stronger than we need as this hardly changes the proof. If one of the three points is contained in $\text{BOX}(\cdot, \cdot)$ defined by the other two points then $\mathcal{H}_\sigma(u_1, u_2) \leqslant \mathcal{G}_\sigma(s_1, s_2, s_3)$ for some $u_1, u_2 \in \{s_1, s_2, s_3\}$. The lemma is a special case of this.

The proof is similar to that of Lemma 3.5. If $Sl_\sigma(s_3; \text{BOX}(s_1, s_2)) \leqslant 0$ then $\mathcal{H}_\sigma(s_1, s_2) \leqslant \mathcal{G}_\sigma(s_1, s_2, s_3)$ and we are done. So assume the opposite:

$$Sl_\sigma(s_3; \text{BOX}(s_1, s_2)) > 0. \tag{3.5}$$

Now assume $s_1 \in \text{BOX}(s_2, s_3)$ or $s_2 \in \text{BOX}(s_1, s_3)$ or $s_3 \in \text{BOX}(s_1, s_2)$. We cannot have the latter since that contradicts (3.5). Hence, either s_1 or s_2 is contained in $\text{BOX}(\cdot, \cdot)$ defined by the other two points. By symmetry of $\mathcal{H}$ and $\mathcal{G}$ in their first two arguments, we may assume the latter is true. Hence, $d(s_1, s_2) + d(s_2, s_3) = d(s_1, s_3)$. By Lemma 2.6,

$$Sl_\sigma(s_3; s_1) = Sl_\sigma(s_3; s_2) + Sl_\sigma(s_2; s_1). \tag{3.6}$$

For the first inequality below we use Lemma 2.4 and for the second we use $\alpha < 1/2$ and (3.5):

$$\begin{aligned}
\mathcal{G}_\sigma(s_1, s_2, s_3) &= W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) - \alpha Sl_\sigma(s_3; \text{BOX}(s_1, s_2)) \\
&\geqslant W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) - \alpha Sl_\sigma(s_3; s_2) \\
&> W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_2; s_1) - \frac{1}{2} Sl_\sigma(s_3; s_2) \\
&= W_\sigma(s_1) - \frac{1}{2} Sl_\sigma(s_3; s_1) \\
&= \mathcal{H}_\sigma(s_1, s_3). \qquad \square
\end{aligned}$$

Initially, the potential function is zero and in general it is upper bounded by the optimal value of the given sequence. This is stated in the next two lemmas. Let ϵ be the empty request sequence.

LEMMA 3.7. $\Phi_\epsilon = 0$. *(Follows from Lemma* 3.2. *Used in proof of Theorem* 3.1.*)*

Proof. Any point s is dominated by the origin $\mathcal{O}$ w.r.t. the empty sequence. By Lemma 3.2, we see that $\min \mathcal{F}_\epsilon = \mathcal{F}_\epsilon(\mathcal{O}, \mathcal{O}, \mathcal{O}) = 0$ and $\min \mathcal{G}_\epsilon = \mathcal{G}_\epsilon(\mathcal{O}, \mathcal{O}, \mathcal{O}) = 0$. $\square$

LEMMA 3.8. $\Phi_\rho \leqslant \text{OPT}_\rho$ *for any sequence ρ. (Used in proof of Theorem* 3.1.*)*

Proof. Let q be the endpoint of an optimal solution for ρ. Then $W_\rho(q) = \text{OPT}_\rho$ and $\mathcal{F}_\rho(q, q, q) = \mathcal{G}_\rho(q, q, q) = W_\rho(q)$. Hence, $\min \mathcal{F}_\rho \leqslant W_\rho(q) = \text{OPT}_\rho$ and $\min \mathcal{G}_\rho \leqslant W_\rho(q) = \text{OPT}_\rho$.

$$\Phi_\rho = (1 - \gamma) \min \mathcal{F}_\rho + \gamma \min \mathcal{G}_\rho \leqslant (1 - \gamma)\text{OPT}_\rho + \gamma \text{OPT}_\rho = \text{OPT}_\rho. \qquad \square$$

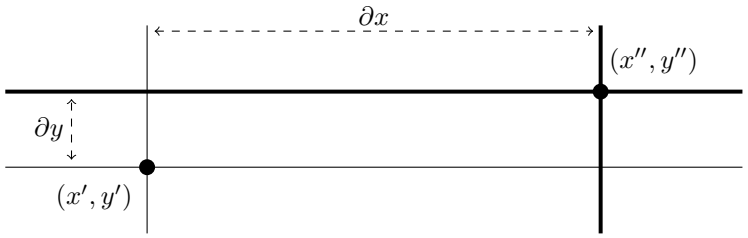

FIG. 3.2. *Two subsequent requests $r' = r(x', y')$ and $r'' = r(x'', y'')$. We assume $\partial x \geqslant \partial y$.*

3.3. Properties of the work function. Each metrical service system has its own specific properties of its work function. For example, Koutsoupias and Papadimitriou show a quasi-convexity property of the work function for the k-server problem [25]. A good understanding of the CNN work function is lacking but the two simple properties we show in this section are enough to prove constant competitiveness. These properties hold for the generalized 2-server problem as well.

Let σ' be an arbitrary request sequence for the CNN problem and let $r' = r(x', y')$ be the last request in σ'.

LEMMA 3.9. *Any $(x, y) \in \mathbb{M}$ is dominated w.r.t. σ' by (x', y) or by (x, y'). (Used in proof of Lemmas* 3.12 *and* 3.14.*)*

Proof. Any point is dominated by a point of the last request. Therefore, (x, y) is dominated by $(x', \hat{y})$ or by $(\hat{x}, y')$ for some $\hat{y} \in \mathbb{Y}$ or $\hat{x} \in \mathbb{X}$. In general, if s is dominated by t then s is dominated by any point on the shortest path between s and t. Now, note that (x', y) is on the shortest path between (x, y) and $(x', \hat{y})$, and that (x, y') is on the shortest path between (x, y) and $(\hat{x}, y')$. □

Let σ' be followed by request $r'' = r(x'', y'')$ and denote the extended sequence by $\sigma'' = \sigma' r''$. To simplify notation we denote $d_{\mathbb{X}}(x', x'') = |x' - x''|$ by ∂x and do the same for y. (See Figure 3.2.) From now on we assume without loss of generality that

$$\partial x \geqslant \partial y.$$

Remember the definition of extended cost. From (2.1) we know that

$$\nabla_{r''}(W_{\sigma'}) = \max_{s \in \mathbb{M}} Sl_{\sigma'}(s; r'').$$

Since this will be the only extended cost that we consider in this proof, we denote it simply by ∇. Further, let $\xi \in \mathbb{M}$ be a point where the maximum is attained, i.e.,

$$\nabla = \nabla_{r''}(W_{\sigma'}) = Sl_{\sigma'}(\xi; r''). \tag{3.7}$$

Point ξ will be used in Lemmas 3.13 and 3.14.

LEMMA 3.10. $\nabla \leqslant (1 + \lambda)\partial x$. *(Follows from Lemma* 2.8. *Used in proof of Lemma* 3.14.*)*

Proof. Any $s \in \mathbb{M}$ is dominated by a point in r' w.r.t. σ'. Hence, by Lemma 2.8(b), we may restrict ourselves to r', i.e., $\nabla = \max_{s \in \mathbb{M}} Sl_{\sigma'}(s; r'') = \max_{s \in r'} Sl_{\sigma'}(s; r'')$. For any point s in r', there is a point in r'' at distance at most ∂x, implying (using (2.2)) $Sl_{\sigma'}(s; r'') \leqslant (1 + \lambda)\partial x$ for any point s in r'. □

3.4. The potential applied to CNN. In this section, we apply our potential function to the CNN problem. Lemmas 3.13, 3.14, and 3.15 state how $\min \mathcal{F}$ and

$\min \mathcal{G}$ increase when a new request r'' arrives, i.e., when going from σ' to σ''. Then, Lemma 3.16 combines these results and gives a lower bound on the increase of the potential function in terms of the extended cost. The proof of Theorem 3.1 is then straightforward.

The following lemma is given without proof as it is easy to check by looking at the definitions.

LEMMA 3.11. *Consider request sequence σ' as fixed and consider the next request $r'' = r(x'', y'')$ as a variable. Then $\min \mathcal{G}_{\sigma' r''}$ and ∇ are Lipschitz continuous in both x'' and y''. (Used in proof of Lemma* 3.14.*)*

Lemma 3.11 is true as well for $\mathcal{F}$ and $\mathcal{H}$ but we do not need that. We shall use the following easy property several times:

$$W_{\sigma'}(s) = W_{\sigma''}(s) \text{ for any } s \in r''. \tag{3.8}$$

Any $s \in \mathbb{M}$ is dominated w.r.t. σ' by a point in r' and Lemma 3.9 gives two candidate points. The next lemma reduces this to one candidate in certain cases.

LEMMA 3.12. *Assume that $\mathcal{F}_{\sigma''}$ or $\mathcal{G}_{\sigma''}$ is minimized in (s_1, s_2, s_3). Then, the following is true for any $i \in \{1, 2, 3\}$.*

1. *If $s_i = (x'', y)$ for some $y \neq y'$ then (x', y) dominates s_i w.r.t. σ'.*
2. *If $s_i = (x, y'')$ for some $x \neq x'$ then (x, y') dominates s_i w.r.t. σ'.*

(Follows from Lemmas 3.2 and 3.9. *Used in proof of Lemmas* 3.13 *and* 3.14.*)*

Proof. We only prove the first, since the second follows by symmetry. By Lemma 3.9, point $s_i = (x'', y)$ is dominated w.r.t. σ' by (x'', y') or by (x', y). Suppose the first is true. Then, using (3.8), s_i is dominated by this point w.r.t. σ'' as well. In that case Lemma 3.2 implies that $\mathcal{F}_{\sigma''}$ and $\mathcal{G}_{\sigma''}$ are strictly reduced by replacing s_i by (x'', y'). This contradicts the assumption of minimality. Thus, s_i is dominated by (x', y) w.r.t. σ'. ∎

Equation (3.8) implies that if $s_1, s_2, s_3 \in r''$ then

$$\mathcal{H}_{\sigma'}(s_1, s_2) = \mathcal{H}_{\sigma''}(s_1, s_2) \text{ and } \mathcal{F}_{\sigma'}(s_1, s_2, s_3) = \mathcal{F}_{\sigma''}(s_1, s_2, s_3).$$

(This is not true for $\mathcal{G}$.) These two easy equalities will be used several times without reference in the following lemmas. From now, we let

$$\alpha \leqslant \frac{1 - \lambda}{12 + 4\lambda}. \tag{3.9}$$

We use this bound for Lemmas 3.13 and 3.14, although for Lemma 3.13 we could do with a weaker bound.

LEMMA 3.13. *Let $\mathcal{F}_{\sigma''}$ be minimized in (s_1, s_2, s_3). There are constants $c_1, c_2 > 0$ (depending on λ) such that,*

(Case A*) if the cardinality of $\{s_1, s_2, s_3\}$ is* 1 *or* 2 *then*

$$\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_1 \nabla, \text{ and}$$

(Case B*) if the cardinality of $\{s_1, s_2, s_3\}$ is* 3 *then*

$$\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_2 \partial y.$$

(Follows from Lemmas 3.2, 3.3, 3.5, and 3.12. *Used in proof of Lemma* 3.16.*)*

Proof. Lemma 3.3 tells us that $s_1, s_2, s_3 \in r''$. We use this in both cases.

Case A. By Lemma 3.5, there are points $u_1, u_2 \in \{s_1, s_2, s_3\}$ such that $\mathcal{H}_{\sigma''}(u_1, u_2) \leqslant \mathcal{F}_{\sigma''}(s_1, s_3, s_3)$. Remember the definition of ξ in (3.7):

$$\begin{aligned}
\min \mathcal{F}_{\sigma''} &= \mathcal{F}_{\sigma''}(s_1, s_2, s_3) \\
&\geqslant \mathcal{H}_{\sigma''}(u_1, u_2) \\
&= \mathcal{H}_{\sigma'}(u_1, u_2) \\
&= \mathcal{F}_{\sigma'}(u_1, u_2, \xi) + \alpha Sl_{\sigma'}(\xi; \{u_1, u_2\}) \\
&\geqslant \min \mathcal{F}_{\sigma'} + \alpha Sl_{\sigma'}(\xi; \{u_1, u_2\}) \\
&\geqslant \min \mathcal{F}_{\sigma'} + \alpha Sl_{\sigma'}(\xi; r'') \\
&= \min \mathcal{F}_{\sigma'} + \alpha \nabla.
\end{aligned}$$

Case B. Since the three points are different, at least one of the these points differs from both (x'', y') and (x', y''). By Lemma 3.12, this point is dominated w.r.t. σ' by a point at distance ∂x or ∂y. Now we use Lemma 3.2 with $\delta = \partial y$. Note that, by our bound on α, all right-hand sides in Lemma 3.2 are at least $\alpha(1 - \lambda)\partial y$:

$$\begin{aligned}
\min \mathcal{F}_{\sigma''} &= \mathcal{F}_{\sigma''}(s_1, s_2, s_3) \\
&= \mathcal{F}_{\sigma'}(s_1, s_2, s_3) \\
&\geqslant \min \mathcal{F}_{\sigma'} + \alpha(1 - \lambda)\partial y. \qquad \square
\end{aligned}$$

LEMMA 3.14. $\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant c_3 \nabla - c_4 \partial y$ *for some constants* $c_3, c_4 > 0$ *depending on* λ. *(Follows from Lemmas* 2.4, 3.2, 3.3, 3.6, 3.10, 3.11, and 3.12. *Used in proof of Lemma* 3.16.*)*

Proof. By Lemma 3.11, it is enough to prove that $\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant c_3 \nabla$ under the assumption that $y' = y''$. So we assume $y' = y''$ and denote both by y^*.

Let $\mathcal{G}_{\sigma''}$ be minimized for (s_1, s_2, s_3) and let $s_i = (x_i, y_i)$, for $i \in \{1, 2, 3\}$. We make the following partition of possible cases:

Case 1. $y_1 = y^*$ and $y_2 = y^*$ and $y_3 = y^*$;
Case 2. $y_1 = y^*$ and $y_2 = y^*$ and $y_3 \neq y^*$;
Case 3. $y_1 \neq y^*$ or $y_2 \neq y^*$.

By Lemma 3.3, we have $s_1, s_2, s_3 \in r''$. We shall use this property several times here. For example, if $y_i \neq y^*$ then $x_i = x''$.

Case 1. We apply Lemma 3.6: $\mathcal{H}_{\sigma''}(u_1, u_2) \leqslant \mathcal{G}_{\sigma''}(s_1, s_2, s_3)$ for some $u_1, u_2 \in \{s_1, s_2, s_3\}$.

$$\begin{aligned}
\min \mathcal{G}_{\sigma''} &= \mathcal{G}_{\sigma''}(s_1, s_2, s_3) \\
&\geqslant \mathcal{H}_{\sigma''}(u_1, u_2) \\
&= \mathcal{H}_{\sigma'}(u_1, u_2) \\
&= \mathcal{G}_{\sigma'}(u_1, u_2, \xi) + \alpha Sl_{\sigma'}(\xi; \textsc{Box}(u_1, u_2)) \\
&\geqslant \min \mathcal{G}_{\sigma'} + \alpha Sl_{\sigma'}(\xi; \textsc{Box}(u_1, u_2)) \\
&\geqslant \min \mathcal{G}_{\sigma'} + \alpha Sl_{\sigma'}(\xi; r'') \\
&= \min \mathcal{G}_{\sigma'} + \alpha \nabla.
\end{aligned}$$

The last inequality follows from $\textsc{Box}(u_1, u_2) \subset r''$ and Lemma 2.4.

Case 2. Since $\textsc{Box}(s_1, s_2) \subset r''$ and $s_1, s_2, s_3 \in r''$ we have

$$\mathcal{G}_{\sigma''}(s_1, s_2, s_3) = \mathcal{G}_{\sigma'}(s_1, s_2, s_3).$$

By Lemma 3.12, point $s_3 = (x'', y_3)$ is dominated by point $t = (x', y_3)$ w.r.t. σ'. Now

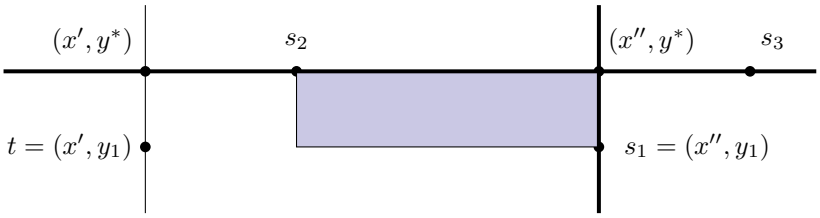

FIG. 3.3. *Case* 3 *of Lemma* 3.14: $y_1 \neq y^*$. *The shaded area is* $\textsc{Box}(s_1, s_2)$.

we apply Lemma 3.2(d):

$$\begin{aligned}\min \mathcal{G}_{\sigma''} &= \mathcal{G}_{\sigma''}(s_1, s_2, s_3)\\ &= \mathcal{G}_{\sigma'}(s_1, s_2, s_3)\\ &\geqslant \mathcal{G}_{\sigma'}(s_1, s_2, t) + \alpha(1-\lambda)\partial x\\ &\geqslant \min \mathcal{G}_{\sigma'} + \alpha(1-\lambda)\partial x\\ &\geqslant \min \mathcal{G}_{\sigma'} + \alpha(1-\lambda)\nabla/(1+\lambda).\end{aligned}$$

The last inequality is given by Lemma 3.10.

Case 3. Unlike the previous two cases, we may now have $\textsc{Box}(s_1, s_2) \not\subseteq r''$ which makes the proof slightly more complicated (see Figure 3.3). By symmetry, we may assume that $y_1 \neq y^*$. This implies $x_1 = x''$ and point $s_1 = (x'', y_1)$ is dominated by point $t = (x', y_1)$ with respect to σ'. We now apply Lemma 3.2(f):

$$\begin{aligned}\mathcal{G}_{\sigma'}(s_1, s_2, s_3) &\geqslant \mathcal{G}_{\sigma'}(t, s_2, s_3) + \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right)\partial x\\ &\geqslant \min \mathcal{G}_{\sigma'} + \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right)\partial x.\end{aligned} \tag{3.10}$$

It remains to bound $\min \mathcal{G}_{\sigma''} - \mathcal{G}_{\sigma'}(s_1, s_2, s_3)$. We have[3]

$$\begin{aligned}&\min \mathcal{G}_{\sigma''} - \mathcal{G}_{\sigma'}(s_1, s_2, s_3)\\ &\quad= \mathcal{G}_{\sigma''}(s_1, s_2, s_3) - \mathcal{G}_{\sigma'}(s_1, s_2, s_3)\\ &\quad= \alpha Sl_{\sigma'}(s_3; \textsc{Box}(s_1, s_2)) - \alpha Sl_{\sigma''}(s_3; \textsc{Box}(s_1, s_2))\\ &\quad\geqslant -2\alpha\partial x.\end{aligned} \tag{3.11}$$

The last inequality follows from $W_{\sigma'}(s_3) = W_{\sigma''}(s_3)$ and from $W_{\sigma''}(s) - W_{\sigma'}(s) \leqslant 2\partial x$ for any point $s \in \mathbb{M}$ (and $s \in \textsc{Box}(s_1, s_2)$ in particular). Below we use, subse-

[3]A more careful analysis gives a bound $-(1+\lambda)\alpha\partial x$ instead of $-2\alpha\partial x$.

quently, (3.11), (3.10), (3.9) and Lemma 3.10:

$$\begin{aligned}
&\min \mathcal{G}_{\sigma''} \\
&\quad = \mathcal{G}_{\sigma''}(s_1, s_2, s_3) \\
&\quad \geqslant \mathcal{G}_{\sigma'}(s_1, s_2, s_3) - 2\alpha\partial x \\
&\quad \geqslant \min \mathcal{G}_{\sigma'} + \left(\frac{1}{2}(1-\lambda) - \alpha(1+\lambda)\right)\partial x - 2\alpha\partial x \\
&\quad = \min \mathcal{G}_{\sigma'} + \left(\frac{1}{2}(1-\lambda) - \alpha(3+\lambda)\right)\partial x \\
&\quad \geqslant \min \mathcal{G}_{\sigma'} + \left(\frac{1}{2}(1-\lambda) - \frac{1}{4}(1-\lambda)\right)\partial x \\
&\quad = \min \mathcal{G}_{\sigma'} + \frac{1}{4}(1-\lambda)\partial x \\
&\quad \geqslant \min \mathcal{G}_{\sigma'} + \frac{1}{4}(1-\lambda)\nabla/(1+\lambda).
\end{aligned}$$

This completes the proof of the last case. ∎

In Lemma 3.16, we combine Lemmas 3.13 and 3.14 and distinguish the same two cases A and B as we did in Lemma 3.13. Lemma 3.14 will be used only for Case B, although it holds in general. For Case A, we need the following different bound.

LEMMA 3.15. *Let $\mathcal{F}_{\sigma''}$ be minimized in (s_1, s_2, s_3). If the cardinality of (s_1, s_2, s_3) is 1 or 2 then $\min \mathcal{G}_{\sigma''} \geqslant \min \mathcal{G}_{\sigma'}$.(Follows from Lemmas* 3.4 and 3.5. *Used in proof of Lemma* 3.16.*)*

Proof. By Lemma 3.5, there are points $u_1, u_2 \in \{s_1, s_2, s_3\}$ such that $\min \mathcal{H}_{\sigma''} \leqslant \mathcal{H}_{\sigma''}(u_1, u_2) \leqslant \mathcal{F}_{\sigma''}(s_1, s_2, s_3) = \min \mathcal{F}_{\sigma''}$. In Lemma 3.4, the inequalities are the other way around. Hence,

$$\min \mathcal{F}_{\sigma''} = \min \mathcal{G}_{\sigma''} = \min \mathcal{H}_{\sigma''}.$$

Further, note that $\min \mathcal{H}_{\sigma''} \geqslant \min \mathcal{H}_{\sigma'}$ (since, by defintion of $\mathcal{H}$, we have that $\mathcal{H}_{\sigma''}(t_1, t_2) \geqslant \mathcal{H}_{\sigma'}(t_1, t_2)$ for any pair of points t_1, t_2). We conclude that

$$\min \mathcal{G}_{\sigma''} = \min \mathcal{H}_{\sigma''} \geqslant \min \mathcal{H}_{\sigma'} \geqslant \min \mathcal{G}_{\sigma'},$$

where the last inequality follows again from Lemma 3.4. ∎

LEMMA 3.16. *$\Phi_{\sigma''} - \Phi_{\sigma'} \geqslant c_5\nabla$ for some constant $c_5 > 0$, depending on λ. (Follows from Lemmas* 3.13 and 3.14. *Used in proof of Theorem* 3.1.*)*

Proof.

$$\Phi_{\sigma''} - \Phi_{\sigma'} = (1-\gamma)\left(\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'}\right) + \gamma\left(\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'}\right).$$

Let $\mathcal{F}_{\sigma''}$ be minimized in (s_1, s_2, s_3). We distinguish between the same two cases as in Lemma 3.13.

Case A. The cardinality of $\{s_1, s_2, s_3\}$ is 1 or 2.

Case B. The cardinality of $\{s_1, s_2, s_3\}$ is 3.

Case A. By Lemma 3.13, $\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_1\nabla$ for some constant $c_1 > 0$ and by Lemma 3.15, $\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant 0$. Hence,

$$\Phi_{\sigma''} - \Phi_{\sigma'} \geqslant (1-\gamma)c_1\nabla.$$

Case B. By Lemma 3.13 and Lemma 3.14,

$$\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'} \geqslant c_2\, \partial y,$$
$$\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant c_3 \nabla - c_4 \partial y$$

for some constants $c_2, c_3, c_4 > 0$. Hence,

$$\begin{aligned}\Phi_{\sigma''} - \Phi_{\sigma'} &= (1-\gamma)\,(\min \mathcal{F}_{\sigma''} - \min \mathcal{F}_{\sigma'}) + \gamma\,(\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'}) \\ &\geqslant (1-\gamma) c_2\, \partial y + \gamma(c_3 \nabla - c_4 \partial y) \\ &= \gamma c_3 \nabla + ((1-\gamma)c_2 - \gamma\, c_4)\partial y.\end{aligned}$$

By choosing γ small enough, the constant before ∂y will be positive. We choose $(1-\gamma) = \gamma c_4 / c_2$, i.e., $\gamma = c_2/(c_2 + c_4)$. Hence,

$$\Phi_{\sigma''} - \Phi_{\sigma'} \geqslant \gamma c_3 \nabla. \tag{3.12}$$

Combining Case A and Case B we obtain

$$\Phi_{\sigma''} - \Phi_{\sigma'} \geqslant \min\{(1-\gamma)c_1, \gamma c_3\}\, \nabla = c_5 \nabla,$$

where

$$c_5 = \min\{(1-\gamma)c_1, \gamma c_3\} = \min\left\{\frac{c_1 c_4}{c_2 + c_4}, \frac{c_2 c_3}{c_2 + c_4}\right\}. \qquad \square$$

Proof of Theorem 3.1. Let ρ be any request sequence. Using Lemma 3.16 and taking the sum over all requests, we get

$$\Phi_\rho - \Phi_\epsilon \geqslant c_5 \nabla_\rho.$$

Lemma 3.7 states that $\Phi_\epsilon = 0$ and Lemma 3.8 states that $\Phi_\rho \leqslant \textsc{Opt}_\rho$. Hence,

$$\nabla_\rho \leqslant \frac{1}{c_5}\, \textsc{Opt}_\rho.$$

By Lemma 2.2, the competitive ratio is at most $(1/c_5 - 1)/\lambda$.

4. General metric spaces. In this section, we extend Theorem 3.1 to arbitrary symmetric metric spaces.

THEOREM 4.1. *The work function algorithm* WFA_λ *is constant competitive for the generalized* 2*-server problem for any constant* λ *with* $0 < \lambda < 1$.

On one hand, the generalization of the proof is easy since all lemmas stay exactly the same, apart from some constants. Moreover, the only proof that really changes is that of Lemma 3.6. However, to prove this lemma we make the potential function even more complex than it already is.

A small problem that appears in a discrete metric space is that the new potential function may no longer be a Lipschitz continuous function of the given request as we stated in Lemma 3.11. To overcome this, we extend the metric space into a metric space $\overline{\mathbb{M}} \supseteq \mathbb{M}$ where any two points are joint by a continuous path, i.e., for any pair $u_1, u_2 \in \overline{\mathbb{M}}$ and $\zeta \in [0,1]$ there is a point $u_3 \in \overline{\mathbb{M}}$ such that $d(u_1, u_3) = \zeta d(u_1, u_2)$ and $d(u_2, u_3) = (1-\zeta) d(u_1, u_2)$. This can easily be done and is a common assumption for online routing problems. See, for example, [12] for a discussion on this. We avoid using the notation $\overline{\mathbb{M}}$ and simply assume that $\mathbb{M}$ has this property. Note that this is done only for the analysis. The request sequence and the work function algorithm will only use points of the original metric space.

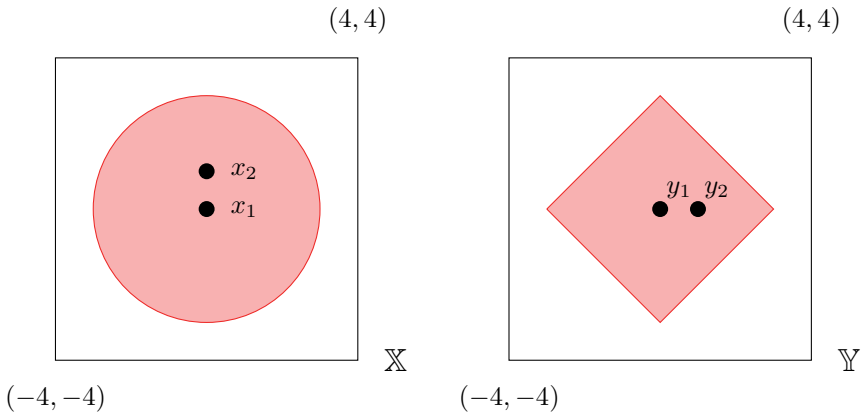

FIG. 4.1. *Example of* SPHERES. *Here,* $\mathbb{X}$ *is the Euclidean plane and* $\mathbb{Y}$ *is the plane with the* L_1 *norm. Further,* $s_1 = (x_1, y_1) = ((0,0),(0,0))$, *and* $s_2 = (x_2, y_2) = ((0,1),(1,0))$. *Hence,* $d^{\mathbb{X}}(x_1, x_2) = 1$ *and* $d^{\mathbb{Y}}(y_1, y_2) = 1$. *If constant* $\eta = 3$ *then* SPHERES(s_1, s_2) *is the Cartesian product of the shaded areas.*

4.1. Adjusting the potential. The first point in the CNN proof where we used the restriction to $\mathbb{R}^2$ is in the potential function: The set BOX(s_1, s_2) is defined only for $s_1, s_2 \in \mathbb{R}^2$. It was defined especially for Lemma 3.6 which says that if the points (s_1, s_2, s_3) have the same x- or y-coordinate then one of them is redundant. We applied this in Lemma 3.14 (Case 1) where we replaced the redundant point by point ξ. Lemma 3.6 still holds for a general metric space but its proof does not apply anymore because equality (3.6) is in general an inequality: For any three points (s_1, s_2, s_3) and sequence σ,

$$Sl_\sigma(s_3; s_1) \leq Sl_\sigma(s_3; s_2) + Sl_\sigma(s_2; s_1).$$

Unfortunately, we need $\geq$ here for the proof of Lemma 3.6 to hold. Looking ahead at (4.4) one sees an alternative inequality which takes the place of (3.6). The trick is quite simple. We make two changes to the potential function: We add the constraint that s_3 should be relatively far from s_1 and s_2 and we take two different measures for slack. (See Figure 4.2.) The intuition is that if a point b has a nonnegative slack with respect to a point a then by using a steeper slack function which has parameter $\mu > \lambda$, the slack of b with respect to a is at least $(\mu - \lambda)d(a, b)$. We make this precise below.

The following definition takes the place of BOX. (See Figure 4.1) Let $\eta \gg 1$.

$$\begin{aligned}\text{SPHERES}(s_1, s_2) = \{\ (x, y) \in \mathbb{X} \times \mathbb{Y} \mid\ & d^{\mathbb{X}}(x, x_1) \leqslant \eta \cdot d^{\mathbb{X}}(x_1, x_2) \\ & \text{and } d^{\mathbb{Y}}(y, y_1) \leqslant \eta \cdot d^{\mathbb{Y}}(y_1, y_2)\ \}.\end{aligned}$$

Note that SPHERES is in fact the Cartesian product of a sphere around x_1 and a sphere around y_1. Instead of (x_1, y_1), we could also take (x_2, y_2) or somehow a point in between. This makes no real difference if η is large. (One could think of BOX(s_1, s_2) as the Cartesian product of a 1-dimensional sphere of diameter $|x_2 - x_1|$ around point $(x_1 + x_2)/2$ and a 1-dimensional sphere of diameter $|y_2 - y_1|$ around point $(y_1 + y_2)/2$.)

The other change that we make in the potential function is adjusting the constants. The whole proof for Theorem 3.1 is still valid (up to a constant) if we replace

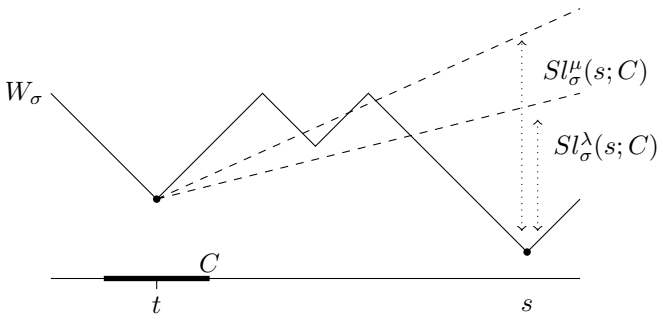

FIG. 4.2. *Two kinds of slack: one with parameter λ and one with parameter $\mu > \lambda$.*

the λ's that appear in the potential function by some other constant μ for which $\lambda \leqslant \mu < 1$, while keeping WFA_λ the same. (There is no need to verify this claim since we do not use it explicitly.) This freedom in the parameter leaves a way for fine tuning the potential function as we will do here. We fix such a μ with $\lambda < \mu < 1$ and define for $s \in \mathbb{M}$ and $C \subseteq \mathbb{M}$ the slack like we did before but now with μ instead of λ. In addition, we keep the old definition and add the parameter λ in the notation:

$$Sl^\mu_\sigma(s;C) = \min_{t\in C}\{W_\sigma(t) + \mu d(s,t)\} - W_\sigma(s),$$
$$Sl^\lambda_\sigma(s;C) = \min_{t\in C}\{W_\sigma(t) + \lambda d(s,t)\} - W_\sigma(s).$$

Next, we define the new $\mathcal{H}_\sigma, \mathcal{F}_\sigma$, and $\mathcal{G}_\sigma$. For simplicity, we keep the same names although they are now slightly different functions:

$$\mathcal{H}_\sigma(s_1,s_2) = W_\sigma(s_1) - \frac{1}{2}Sl^\mu_\sigma(s_2;s_1),$$
$$\mathcal{F}_\sigma(s_1,s_2,s_3) = \mathcal{H}_\sigma(s_1,s_2) - \beta Sl^\lambda_\sigma(s_3;\{s_1,s_2\}),$$
$$\mathcal{G}_\sigma(s_1,s_2,s_3) = \mathcal{H}_\sigma(s_1,s_2) - \beta Sl^\lambda_\sigma(s_3;\textsc{Spheres}(s_1,s_2)).$$

Note that μ is used for the slack of s_2 while λ is used for the slack of s_3. The potential function is

$$\Phi_\sigma = (1-\kappa)\min \mathcal{F}_\sigma + \kappa \min \mathcal{G}_\sigma,$$

where $0 < \kappa < 1$. To prove constant competitiveness, there is no need to specify precise values of the constants. We only need to choose the constants either large or small enough. The order in which we choose them and the domains are listed below. For example, given λ and the choice of μ, there is a number η_0 such that any choice $\eta \geqslant \eta_0$ is fine for our proof. We do not compute the values η_0, β_0, or κ_0 but it will be clear from the proof that such values exist:

λ :	given parameter;	
μ :	$\lambda < \mu < 1$;	
η :	$\eta \geq \eta_0 \gg 1$,	where η_0 depends on λ and μ;
β :	$0 < \beta \leqslant \beta_0 < 1/2$,	where β_0 depends on λ, μ, and η;
κ :	$0 < \kappa < \kappa_0 < 1$,	where κ_0 depends on λ, μ, η, and β.

4.2. Adjusting the proofs of the lemmas. All lemmas stay exactly the same apart from some constants. Moreover, all proofs stay basically the same. The only proof that is really different is that of Lemma 3.6. Let us go over all the lemmas one by one.

Nothing changes for section 2 since it comes before the potential function and holds for any metric space. The first lemma in section 3 is Lemma 3.2. The lemma holds with different constants. The bounds we get are

$$\begin{aligned}
(a),(d) &: \delta \cdot \beta(1-\lambda),\\
(b),(c) &: \delta \cdot \left(\frac{1}{2}(1-\mu) - \beta(1+\lambda)\right),\\
(e) &: \delta \cdot \left(\frac{1}{2}(1-\mu) - \eta\beta(1+\lambda)\right),\\
(f) &: \delta \cdot \left(\frac{1}{2}(1-\mu) - (\eta+1)\beta(1+\lambda)\right).
\end{aligned} \tag{4.1}$$

The proof for (a),(b),(c),(d) remains the same, only α becomes β and some of the λ's become μ. In (e), there is an additional factor η because a move of s_2 over some distance may cause the border of $\textsc{Spheres}(s_1, s_2)$ to move by η times this distance. For a move of s_1, this factor is $\eta+1$ since Spheres is defined around s_1. The precise bounds are not so important. We only need to see that we can choose β small enough to let all the right-hand sides be $\Omega(\delta)$.

Nothing changes for Lemma 3.3. In the proof of Lemma 3.4, only Box needs to be replaced by Spheres. In the proof of Lemma 3.5, we only need to update the definition of $\mathcal{F}$.

New proof of Lemma 3.6.

Proof. Let s_1, s_2, s_3 have the same y-coordinate. We may assume that

$$Sl^{\lambda}_{\sigma}(s_3; \textsc{Spheres}(s_1, s_2)) > 0, \tag{4.2}$$

since otherwise $\mathcal{H}_\sigma(s_1, s_2) \leqslant \mathcal{G}_\sigma(s_1, s_2, s_3)$ and we are done. By this assumption, we have $s_3 \notin \textsc{Spheres}(s_1, s_2)$. Hence, $d(s_1, s_3) > \eta d(s_1, s_2)$ (using $d^{\mathbb{X}}(s_i, s_j) = d(s_i, s_j)$ for $i, j \in \{1, 2, 3\}$). Then

$$\begin{aligned}
Sl^{\mu}_{\sigma}(s_3; s_1) &= W_\sigma(s_1) + \mu d(s_1, s_3) - W_\sigma(s_3)\\
&= W_\sigma(s_1) + \lambda d(s_1, s_3) - W_\sigma(s_3) + (\mu-\lambda) d(s_1, s_3)\\
&= Sl^{\lambda}_{\sigma}(s_3; s_1) + (\mu-\lambda) d(s_1, s_3)\\
&> Sl^{\lambda}_{\sigma}(s_3; s_1) + \eta(\mu-\lambda) d(s_1, s_2).
\end{aligned} \tag{4.3}$$

By choosing η large enough (given the values of λ and μ), we guarantee that $\eta(\mu-\lambda) \geqslant 1+\mu$. If we also use that $(1+\mu)d(s_1, s_2) \geqslant Sl^{\mu}_{\sigma}(s_2; s_1)$ (follows directly from (4.1)) then the analogue of (3.6) becomes

$$Sl^{\mu}_{\sigma}(s_3; s_1) > Sl^{\lambda}_{\sigma}(s_3; s_1) + Sl^{\mu}_{\sigma}(s_2; s_1). \tag{4.4}$$

The remainder of the proof is similar to the original proof. For the first two inequalities below we use, respectively, (4.4) and Lemma 2.4. For the last inequality we use (4.2)

and $\beta < 1/2$:

$$\begin{aligned}\mathcal{H}_\sigma(s_1, s_3) &= W_\sigma(s_1) - \frac{1}{2} Sl^\mu_\sigma(s_3; s_1) \\ &< W_\sigma(s_1) - \frac{1}{2} Sl^\mu_\sigma(s_2; s_1) - \frac{1}{2} Sl^\lambda_\sigma(s_3; s_1) \\ &\leqslant W_\sigma(s_1) - \frac{1}{2} Sl^\mu_\sigma(s_2; s_1) - \frac{1}{2} Sl^\lambda_\sigma(s_3; \textsc{Spheres}(s_1, s_2)) \\ &< W_\sigma(s_1) - \frac{1}{2} Sl^\mu_\sigma(s_2; s_1) - \beta Sl^\lambda_\sigma(s_3; \textsc{Spheres}(s_1, s_2)) \\ &= \mathcal{G}_\sigma(s_1, s_2, s_3). \qquad \square\end{aligned}$$

The proof of Lemma 3.7 stays the same. Also the proof of Lemma 3.8 stays the same apart from α becoming β and γ becoming κ. Lemmas 3.9 and 3.10 do not depend on the potential function, nor do they depend on the metric space. These lemmas and proofs stay exactly the same. Lemma 3.11 was given without proof and again it can easily be verified from the definitions. The proof of Lemma 3.12 does not change. For Lemma 3.13 there are a few small changes. In Case A, only α becomes β. In Case B, the last inequality is different since the inequalities of Lemma 3.2 are different. The new values were given in formula (4.1). All we need to notice is that by choosing β small enough (depending on λ, μ, and η), the right-hand sides are $\Omega(\delta)$.

Also in the proof of Lemma 3.14 there are a few small changes. Of course, α becomes β and Box becomes Spheres. The new function $\mathcal{G}$ is still Lipschitz continuous. Hence, we may assume $y' = y''$. We consider the same three cases and the proof for the first and second cases remains the same. For Case 3 we need to use the new bounds of Lemma 3.2. Then, (3.10) becomes

$$\mathcal{G}_{\sigma'}(s_1, s_2, s_3) \geqslant \min \mathcal{G}_{\sigma'} + \left(\frac{1}{2}(1-\mu) - (\eta+1)\beta(1+\lambda)\right) \partial x.$$

Combining this with (3.11) as we did, we get

$$\min \mathcal{G}_{\sigma''} - \min \mathcal{G}_{\sigma'} \geqslant \left(\frac{1}{2}(1-\mu) - (\eta+1)\beta(1+\lambda) - 2\beta\right) \partial x.$$

By choosing β small enough (depending on λ, μ, and η), the right-hand side is at least $c_3 \nabla$ for some constant c_3 (using Lemma 3.10). The proof of Lemma 3.15 remains the same. Finally, the only change in the proof of Lemma 3.16 is that γ becomes κ.

5. A decomposition approach for the generalized *k*-server problem. The generalized k-server problem appears a lot more complicated for dimensions $k \geqslant 3$. It is unclear if for any fixed $k \geqslant 3$ a constant competitive ratio $f(k)$ is possible at all. In any case, the ratio will be at least $k^{\Omega(k)}$ [17]. The question is important for its relation to sum problems discussed in the introduction. Interestingly, the proof for $k = 2$ does show a decomposition into subproblems which can be generalized to any k and which seems to be a real simplification of the problem. Although an answer to these subproblems is missing, it does give an example of decomposing a sum problem into (apparently) easier problems.

Suppose that we can find k functions $\mathcal{F}^{(i)}_\sigma : \mathbb{M}^{k+1} \to \mathbb{R}$ for $i = 1, 2, \ldots, k$ with the following two properties:

(i) $\min \mathcal{F}^{(i)}_\epsilon = 0$ and $\min \mathcal{F}^{(i)}_\sigma \leqslant \textsc{Opt}_\sigma$ (where ϵ is the empty sequence);

(ii) let $r' = r(x'_1, x'_2, \ldots, x'_k)$ and $r'' = r(x''_1, x''_2, \ldots, x''_k)$ be two subsequent requests and let $\pi_1, \pi_2, \ldots, \pi_k$ be a permutation of $1, 2, \ldots, k$ such that $\partial x_{\pi_1} \leqslant \partial x_{\pi_2} \leqslant \cdots \leqslant \partial x_{\pi_k}$, where $\partial_{\pi_i} = |x''_{\pi_i} - x'_{\pi_i}|$. Further, denote $\sigma' = \sigma r'$ and $\sigma'' = \sigma r' r''$ and denote the extended cost $\nabla_{r''}(W_{\sigma'})$ simply by ∇. Then for all i there are constants $a^{(i)}, b^{(i)}, c^{(i)}, d^{(i)} > 0$ (depending on k and λ) such that either (A) or (B) holds:

(A) $\min \mathcal{F}^{(i)}_{\sigma''} - \min \mathcal{F}^{(i)}_{\sigma'} \geqslant a^{(i)}\nabla - b^{(i)}\partial x_{\pi_{i-1}}$ and for all $j > i$
$\min \mathcal{F}^{(j)}_{\sigma''} - \min \mathcal{F}^{(j)}_{\sigma'} \geqslant 0$,

(B) $\min \mathcal{F}^{(i)}_{\sigma''} - \min \mathcal{F}^{(i)}_{\sigma'} \geqslant c^{(i)}\partial x_{\pi_i}$.

Then, the following potential function proves that WFA_λ is constant competitive for some constant depending on k and λ:

$$\Phi_\sigma = \sum_{i=1}^{k} \gamma^{(i)} \cdot \min_{s_1,\ldots,s_{k+1} \in \mathbb{M}} \mathcal{F}^{(i)}_\sigma(s_1, s_2, \ldots, s_{k+1}),$$

where $\gamma^{(1)} + \cdots + \gamma^{(k)} = 1$ and $\gamma^{(i)}/\gamma^{(i+1)} \geqslant b^{(i+1)}/c^{(i)}$ for $i = 1, 2, \ldots, k-1$.

First, let us see how this relates to our proof for $k = 2$. We denoted $\mathcal{F}^{(1)} = \mathcal{F}$ and $\mathcal{F}^{(2)} = \mathcal{G}$ and denoted x_1 and x_2 by x and y. We assumed $\partial y \leqslant \partial x$ which implies $\pi_1 = 2$ and $\pi_2 = 1$. Property (i) holds (and was used for Lemma 3.7 and Lemma 3.8). Now, it is easy to check that property (ii) corresponds to Lemmas 3.13, 3.14, and 3.15. (Define $\partial x_{\pi_0} := 0$ and note that $\nabla = O(\partial x_k)$.)

Next, we give a short sketch why this would give a proof of competitiveness and then we argue why this is an interesting decomposition. We need to show that the increase in the potential for the new request r'' is at least some constant times ∇. (Then, if additionally $\Phi_\epsilon = 0$ and $\Phi_\sigma \leqslant \textsc{Opt}_\sigma$, competitiveness follows from Lemma 2.2.) First consider $i = 1$. (Define $\partial x_{\pi_0} := 0$.) If case (A) applies then we are done. So assume from now that case (B) applies for $i = 1$. Consider $i = 2$. If case (A) applies for $i = 2$ then, by the choice of the $\gamma^{(i)}$, the increase in the potential function is at least

$$\gamma^{(1)} c^{(1)} \partial x_{\pi_1} + \gamma^{(2)}(a^{(2)}\nabla - b^{(2)}\partial x_{\pi_1}) \geqslant \gamma^{(2)} a^{(2)} \nabla.$$

So assume case (B) applies and consider $i = 3$. We can repeat the argument until finally we consider $i = k$. Then the proof follows from case B as well since $\partial x_{\pi_k} = \Omega(\nabla)$.

Now we will argue that the decomposition seemingly simplifies the analysis. Remember that the general idea is to find a potential function Φ_σ with the property that the increase for every new request is at least some constant (depending on k) times the extended cost ∇ of the new request. In the decomposition, this property is split into k weaker properties. Assume that the functions $\mathcal{F}^{(i)}$ are all Lipschitz continuous functions of the last request. By this we mean, if r'' is changed to some other request $\tilde{r}''$ while keeping the arguments $s_1, \ldots, s_{k+1}$ fixed then the value $\mathcal{F}^{(i)}_{\sigma''}(s_1, \ldots, s_{k+1})$ changes by at most some constant (depending on k and λ) times $||r'' - \tilde{r}''||$. Lipschitz continuity seems a natural property. Note that the extended cost $\nabla := \nabla_{r''}(W_{\sigma'})$ is always Lipschitz continuous in r''. If Lipschitz continuity holds, then to prove (ii), we may assume that $\partial x_{\pi_h} = 0$ for all $h < i$, as we did in the proof of Lemma 3.14. For example, for $\mathcal{F}^{(k)}_\sigma$ we only need to show an increase of $\Omega(\nabla)$ under the (strong) condition that $\partial x_j = 0$ for all $j \leqslant k-1$, i.e., under the condition that only one coordinate changes.

We will not speculate on a general decomposition theorem for sum problems and merely say that the outline appears a significant step towards a proof for $k \geqslant 3$ and

is an interesting contribution towards a general theory of competitiveness of metrical service systems.

6. Notes and open problems. The most prominent research direction is to enhance the theory of competitiveness of metrical service (or task) systems and in particular for the generalized work function algorithm. Our proof shows that only very limited information of the work function may be needed to show that WFA_λ performs well. In fact, we only used the obvious properties that apply to any work function, e.g. that any point $s \in \mathbb{M}$ is dominated by some point t on the last request and that the work function is Lipschitz continuous with constant 1. (As a comparison, Koutsoupias and Papadimitriou show for their k-server proof that the k-server work function has some nice quasi-convexity property.) So, if this is all we use, why does not this imply competitiveness for any metrical service system? The answer is that the potential function was designed for the typical requests of the generalized 2-server problem, i.e., the potential function exploits that the support of any work function is a subset of the last request. This kind of analysis, that is purely based on the geometry of a single request, is interesting for metrical service systems in general. For this purpose, our potential function has some valuable ingredients such as the use of convex sets like Box and Spheres and the use of slack functions with different parameters (λ and μ). These techniques are helpful for isolating extreme solutions, i.e., (a small number of) solutions which in a way represent all offline solutions.

An illustrative example is the problem of chasing lines. In this system, the metric space is $\mathbb{R}^d$ and the set $\mathcal{R}$ of requests contains all lines and line segments in $\mathbb{R}^d$. By taking our function $\mathcal{G}$ as the potential function (where $\mathrm{Box}(s_1, s_2)$ is now defined as the line segment between s_1 and s_2), it follows immediately that WFA_λ is constant competitive (independent of d) for any $\lambda \in (0, 1)$. All that we need to notice is the following alternative formulation of Lemma 3.6: If $s_1, s_2, s_3 \in \mathbb{R}^d$ are all on a straight line then $\mathcal{H}_\sigma(u_1, u_2) \leqslant \mathcal{G}_\sigma(s_1, s_2, s_3)$ for some $u_1, u_2 \in \{s_1, s_2, s_3\}$. Now assume that sequence σ is followed by a request r and that s_1, s_2, s_3 minimize $\mathcal{G}_{\sigma r}$. Then, all three points are on the last request r and hence all are on a straight line. The lemma says that one of the three points is redundant. Replacing one of the three points by a point $\xi \in \mathbb{M}$ with maximum extended cost ∇, we see that the increase for the potential function is $\Omega(\nabla)$ and competitiveness follows. The algorithm by Friedman and Linial [19] for line chasing is much less general and uses angles and coordinates in the Euclidean plane. Of course, how one can implement WFA_λ efficiently for the line chasing problem is a different story.

6.1. Open problems. There are some very intriguing open problems in online optimization. Examples are the *k-server conjecture* (deterministic and randomized) and the *dynamic optimality conjecture* [34] for binary search trees. (We refer to [16] for a survey of recent results.) The latter conjecture states that there exists a constant competitive online algorithm for binary search trees. Maybe not so well known is that the binary search tree problem (without insertions or deletions) can be transformed into a metrical service system with loss of a constant factor in the approximation. This can be done as follows. Let $1, 2, \ldots, n$ be the items of the tree. By a binary search tree, we mean a rooted tree with maximum degree three. Then, the metric space consists of all binary search trees with nodes $1, 2, \ldots, n$ and the distance between two trees is the minimum number of rotations needed to transform one tree into the other (or we may take any other distance functions that is within a constant factor). Now, for each item i we define a request r^i which is the set of all binary search trees with root i. The collection of possible requests is $\mathcal{R} = \{r^1, r^2, \ldots, r^n\}$. Let BST be the binary

search tree problem (as defined in [34]) and let BST* be the BST problem modeled as a metrical service system as described above . The next theorem states that the (online) approximation ratios of these two problems are within a constant factor. A similar result is given in [8] (Lemma 1.3) for the list update problem, which is the 1-dimensional equivalent of the BST problem. There, it is shown that any c-competitive algorithm remains c-competitive if the cost to serve i is $\text{depth}(i)-1$ instead of $\text{depth}(i)$. Below, we also show the other direction and model it as a metrical service system.

LEMMA 6.1. *The (online) approximation ratios of the* BST *problem and its metrical service system formulation* BST* *are within a constant factor.*

Proof. In [34], the cost for serving an item i is one plus $\text{depth}(i)$, the depth of item i in the current tree. This differs with our service system model in two ways. First, in our model there is the restriction that i has to be moved to the root in order to serve it. Note that this restriction only increases the cost by a small constant factor since i can be moved to the root and back at a cost $O(\text{depth}(i))$. The other difference is that in our model there is no additive cost of one to serve a request. In particular, that means that items at the root are served at a cost of one in the BST model while these are free in BST*. We call BST* the *zero cost model*. Next, we compare the competitive ratios for BST and BST* with the restriction that items can only be served at the root. Under this restriction, let OPT and OPT_0 denote the optimum in, respectively, the standard cost and the zero cost model. Then for any sequence σ, $\text{OPT}(\sigma) = \text{OPT}_0(\sigma) + |\sigma|$. Let ALG be any c-competitive algorithm for BST*. Then, it is c-competitive for BST as well (See also Lemma 1.3 in [8]):

$$\text{ALG}(\sigma) = \text{ALG}_0(\sigma) + |\sigma| \leqslant c\text{OPT}_0(\sigma) + |\sigma| = c\text{OPT}(\sigma) - (c-1)|\sigma|.$$

For the other direction, assume that some algorithm ALG is c-competitive for BST. We will show that this gives a $(2c-1)$-competitive algorithm for BST*. For any request sequence σ we define σ' as the sequence obtained by removing the repeated requests. For example, if item i is requested three times consecutively then we remove two of these. Now define algorithm ALG$'$ as follows. For any request sequence σ it gives the truncated sequence σ' to ALG and then behaves exactly like ALG. This means that when a requested item i is moved to the root, the search tree remains unchanged until the first moment that a different item is requested. This way, sequence σ is served using the online solution for σ'. By assumption, $\text{ALG}(\sigma') \leqslant c\text{OPT}(\sigma')$. Further, if we assume that the first request is not to the root then $|\sigma'| \leqslant \text{OPT}_0(\sigma')$.

$$\begin{aligned}\text{ALG}'_0(\sigma) &= \text{ALG}'_0(\sigma') = \text{ALG}(\sigma') - |\sigma'| \leqslant c\text{OPT}(\sigma') - |\sigma'| \\ &= c\text{OPT}_0(\sigma') + (c-1)|\sigma'| \leqslant (2c-1)\text{OPT}_0(\sigma') = (2c-1)\text{OPT}_0(\sigma). \qquad \square\end{aligned}$$

The BST problem is still not well understood. It is not known if the problem is NP-hard, nor is there a constant factor offline approximation algorithm known. Lower bounds on the optimal solution are hard to get. However, if constant competitiveness is possible then probably there is no need for this kind of bounds. In online optimization the analysis is usually based on some kind of extreme solutions that in a way represent all possible offline solutions. A simple (and highly relevant) example is the list update problem [8]. The move-to-front rule has optimal competitive ratio of $2 - 2/(n+1)$, where n is the size of the list. It is easy to see that it is 2-competitive since with loss of a factor 2 we may assume that each item can only be served at the front. But then, there is only one optimal solution and the move-to-front algorithm gets one step closer to the optimal solution with every rotation that it makes. The only information about the optimal offline solution that is used in this analysis is

that its current configuration serves the current request. Hence, for list-update it is enough to consider only one offline solution. Another example is the (optimal) double coverage algorithm for the k-server problem on trees [11] where the potential function is defined only by the current configuration of the online solution and that of the optimal solution. An example with two extreme solutions is the line chasing problem that we discussed at beginning of this section. We sketched a proof with a potential function which is defined by three solutions. We could show that one of these was redundant and the proof followed easily. Hence, for WFA_λ applied to line chasing there are only two extreme solutions. This analysis follows purely from the geometry of a single request. There is no need for lower bounds on a sequence of requests. More complicated examples are the $2k-1$ ratio for the k-server problem [25] with a potential based on $k+1$ configurations, and our potential which uses six configurations.

It is not hard to show that WFA_λ is in fact not constant competitive for binary search trees when we define the metric space as in Lemma 6.1. However, all kinds of variations are possible. Consider the following adjustment of the metric space. The cost of a single rotation remains one but the cost of a splaying operation on item i is only some small constant times $\text{depth}(i)$. This way, WFA_λ will behave much like the splay tree algorithm and it seems a good candidate for being constant competitive.

A question that pops up is whether such an approach with an adjusted metric has potential at all since we just noted that WFA_λ is not competitive for the natural distance function. Is WFA_λ robust in the sense that small changes in the metric give small changes in the competitive ratio of WFA_λ? In that case our suggested approach is doomed to fail. Fortunately, the answer is negative and follows from the next example.

Example 1. Consider a metrical service system on a star graph with k leaves. Let c be the center and let $U = \{u_1, u_2, \ldots, u_k\}$ be the set of leaves. The distances are $d(c, u_1) = 1 - \epsilon$ and $d(c, u_i) = 1$, $i = 2, \ldots, k$. The set of requests is $\mathcal{R} = \{\{c\}, \{U \setminus u_2\}, \{U \setminus u_3\}, \ldots, \{U \setminus u_k\}\}$. The optimal online algorithm moves to c whenever the request is $\{c\}$ and moves to u_1 otherwise. The competitive ratio of this algorithm is 1. The work function algorithm WFA_λ behaves the same for any $\lambda \in (0, 1)$ and therefore has ratio 1 as well. If we now change $d(c, u_1)$ from $1 - \epsilon$ to $1 + \epsilon$ then the optimal online algorithm stays the same and now has competitive ratio $1 + \epsilon$. However, WFA_λ can be forced to visit all u_i between two requests for c and has the ratio $(k + \epsilon)/(1 + \epsilon)$.

An obvious drawback of the work function approach for the BST problem is that it is computationally expensive. In fact, no polynomial time constant factor approximation is known. Nevertheless, at the moment it is very interesting to see if a constant competitive algorithm is possible at all, no matter how high the running time.

Below, we list some interesting open problems related to this paper, starting with the BST problem discussed above.

- ◇ Give a constant competitive work function based algorithm for binary search trees (without insertions or deletions). Although the algorithm would be inefficient it would clearly be a big step towards proving competitiveness for more efficient algorithms like splaying.
- ◇ Prove or disprove that the generalized k-server problem or weighted k-server problem has an $f(k)$-competitive algorithm for some function $f(k)$. Same for the randomized problem. An outline for a possible proof is given in section 5.
- ◇ What is the competitive ratio of the k-point request problem, introduced in [30]? Fiat et al. [18] gave an upper bound of $O(9^k)$ which was improved by Burley [9] who showed that the generalized work function algorithm is $O(k2^k)$-

competitive. The best known lower bound is $\Omega(2^k)$ [18]. Just as Burley we conjecture that $O(2^k)$ is possible. A good candidate seems a dynamic work function algorithm: one that adjusts the parameter λ online. Such a dynamic work function algorithm would be more powerful than the generalized work function algorithm. Randomization reduces the ratio drastically as shown by Ramesh [31] who gave an upper bound of $\Omega(k^{13})$ against a lower bound of $k/2$ [18].

- ◇ What is the competitive ratio of the continuous CNN problem? A lower bound of 3 and upper bound of 6.46 is given in [1].
- ◇ Give other examples of natural metrical service systems that have a constant competitive ratio. For example, Friedman and Linial [19] give a competitive algorithm if the requests are a convex subset of $\mathbb{R}^2$. They conjecture that the same applies to $\mathbb{R}^d$ for any fixed d and show that it is enough to prove this for affine half-spaces. In the beginning of this section we sketched a proof that WFA_λ is constant competitive for lines in $\mathbb{R}^d$ and it would be interesting to extend this to half-spaces.
- ◇ The k-server problem has some simple special cases for which $2k-1$ is still the best known ratio, for example the 3-server problem and the k-server problem on a cycle: Find an algorithm with a smaller ratio. The ratio for trees is k but it is unknown if the work function algorithm achieves this ratio. See [23] for more background on this.
- ◇ What is the competitive ratio of the weighted k-point request problem, discussed in [15]? This problem is a special case of the generalized k-server problem and a generalization of the k-point request problem.
- ◇ Extend the theory of sum problems. For example, by analyzing the sum problem of another elementary metrical task system.
- ◇ Prove (or disprove) that the generalized work function algorithm WFA_λ is $O(\log n)$-competitive for the online matching problem on a line. A lower bound of $\Omega(\log n)$ and an upper bound of $O(n)$ were given in [24].

Acknowledgment. I thank the reviewers sincerely for carefully reading this long proof with its numerous details. Their comments have been very useful.

REFERENCES

[1] J. Augustine and N. Gravin, *On the continuous CNN problem*, in Proceedings of the 21st International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 6507, Springer, New York, 2010, pp. 254–265.

[2] E.J. Anderson, K. Hildrumb, A.R. Karlina, A. Rasalac, and M. Saks, *On list update and work function algorithms*, Theoret. Comput. Sci., 287 (2002), pp. 393–418.

[3] G. Ausiello, V. Bonifaci, and L. Laura, *Lazy On-Line Algorithms for Metrical Service Systems*, Technical report DIS 07-04, Department of Computer and System Sciences, Università di Roma La Sapienza, Roma, Italy, 2004.

[4] G. Ausiello, V. Bonifaci, and L. Laura, *On explorers, chasers and cameramen*, in Proceedings of the 3rd International Conference on Fun with Algorithms (FUN2004), Isola d'Elba, Italy, 2004.

[5] Y. Bartal and E. Koutsoupias, *On the competitive ratio of the work function algorithm for the k-server problem*, Theoret. Comput. Sci., 324 (2004), pp. 337–345.

[6] A. Blum and C. Burch, *On-line learning and the metrical task system problem*, Mach. Learn., 39 (2000), pp. 35–58.

[7] A. Borodin, N. Linial, and M. Saks, *An optimal online algorithm for metrical task system*, J. ACM, 39 (1992), pp. 745–763.

[8] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, 1998.

[9] W. Burley, *Traversing layered graphs using the work function algorithm*, J. Algorithms, 20 (1996), pp. 479–511.
[10] M. Chrobak, *Column on online algorithms*, SIGACT News, 34 (2003), pp. 68–77.
[11] M. Chrobak and L. L. Larmore, *An optimal on-line algorithm for k servers on trees*, SIAM J. Comput., 20 (1991), pp. 144–148.
[12] M. Chrobak and L. Larmore, *The server problem and on-line games*, in On-line Algorithms: Proceedings of a DIMACS Workshop, Dimacs Ser. Discrete Math. Theoret. Comput. Sci. 7, 1992, pp. 11–64.
[13] M. Chrobak and L. Larmore, *Metrical Service Systems: Deterministic Strategies*, Technical report UCR-CS-93-1, Department of Computer Science, University of California at Riverside, Riverside, CA, 1993.
[14] M. Chrobak and 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. Woeginger, eds., Lecture Notes in Comput. Sci. 1442, Springer, New York, 1998, pp. 74–96.
[15] M. Chrobak and J. Sgall, *The weighted 2-server problem*, Theoret. Comput. Sci., 324 (2004), pp. 289–312.
[16] E.D. Demaine, D. Harmon, J. Iacono, D. Kane, and M. Pătraşcu, *The geometry of binary search trees*, in Proceedings of the 20th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA09), SIAM, Philadelphia, 2009, pp. 496–505.
[17] A. Fiat and M. Ricklin, *Competitive algorithms for the weighted server problem*, Theoret. Comput. Sci., 130 (1994), pp. 85–99.
[18] A. Fiat, D.P. Foster, H. Karloff, Y. Rabani, Y. Ravid, and S. Vishwanathan, *Competitive algorithms for layered graph traversal*, SIAM J. Comput., 28 (1998), pp. 447–462.
[19] J. Friedman and N. Linial, *On convex body chasing*, Discrete Comput. Geom., 9 (1993), pp. 293–321.
[20] K. Iwama and K. Yonezawa, *Axis-Bound CNN Problem*, Technical report, National Institute of Informatics, Tokyo, Japan, 2001.
[21] K. Iwama and K. Yonezawa, *Axis-bound CNN problem*, IEICE TRANS, E87-A (2002), pp. 1235–1242.
[22] K. Iwama and K. Yonezawa, *The orthogonal CNN problem*, Inform. Process. Lett., 94 (2004), pp. 115–120.
[23] E. Koutsoupias, *The k-server problem*, Comput. Sci. Rev., 3 (2009), pp. 105–118.
[24] E. Koutsoupias and A. Nanavati, *The online matching problem on a line*, in Proceedings of the 1st Workshop on Approximation and Online Algorithms, Springer, Berlin, 2003, pp. 191–197.
[25] E. Koutsoupias and C. Papadimitriou, *On the k-server conjecture*, J. ACM, 42 (1995), pp. 971–983.
[26] E. Koutsoupias and D. Taylor, *The cnn problem and other k-server variants*, in Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1770, Springer, New York, 2000, pp. 581–592.
[27] E. Koutsoupias and D.S. Taylor, *The CNN problem and other k-server variants*, Theoret. Comput. Sci., 324 (2004), pp. 347–359.
[28] M. Manasse, L. McGeoch, and D. Sleator, *Competitive algorithms for on-line problems*, in Proceedings of the 20th Symposium on Theory of Computing, ACM, New York, 1988, pp. 322-333.
[29] M. Manasse, L. McGeoch, and D. Sleator, *Competitive algorithms for server problems*, J. Algorithms, 11 (1990), pp. 208–230.
[30] C.H. Papadimitriou and M. Yannakakis, *Shortest paths without a map*, Theoret. Comput. Sci., 84 (1991), pp. 127–150.
[31] H. Ramesh, *On traversing layered graphs on-line*, J. Algorithms 18 (1995), pp. 480–512.
[32] R. Sitters and L. Stougie, *The generalized two-server problem*, J. ACM, 53 (2006), pp. 437–458.
[33] R. Sitters, L. Stougie, and W. de Paepe, *A competitive algorithm for the general 2-server problem*, in Proceedings of the 30th International Colloquium on Automata, Languages, and Programming, J. Baeten, J. Lenstra, J. Parrow, and G. Woeginger, eds., Lecture Notes in Comput. Sci. 2719, Springer, New York, 2003, pp. 624–636.
[34] D. Sleator and R. Tarjan, *Self-adjusting binary search trees*, J. ACM, 32 (1985), pp. 652–686.
[35] Y.F. Verhoeven, *A lower bound on the competitivity of memoryless algorithms for a generalization of the CNN problem*, Theoret. Comput. Sci., 359 (2006), pp. 58–68.