# Optimal Bounds for Online Page Migration with Generalized Migration Costs

Johannes Schneider[1], Stefan Schmid[2]
[1] IBM Research, Switzerland, [2] TU Berlin & T-Labs, Germany

## Abstract

This paper attends to a generalized version of the classic page migration problem where migration costs are not necessarily given by the migration distance only, but may depend on prior migrations, or on the available bandwidth along the migration path. Interestingly, this problem cannot be viewed from a Metrical Task System (MTS) perspective, despite the generality of MTS: The corresponding MTS has an unbounded state space and, thus, an unbounded competitive ratio. Nevertheless, we are able to present an optimal online algorithm for a wide range of problem variants, improving the best upper bounds known so far for more specific problems. For example, we present a tight bound of $\Theta(\log n/\log\log n)$ for the competitive ratio of the virtual server migration problem introduced recently.

## I. Introduction

The classic Page Migration problem asks for a strategy to dynamically position a page (or a server) in the network such that the access costs to this page (measured in the distance between host issuing the request and the host storing the page) as well as the migration costs of the page are minimized. Concretely, in the realm of online algorithms and competitive analysis, it is assumed that the sequence of requests accessing the page from different locations in the network is arbitrary and cannot be foreseen. In the traditional problem version introduced in the 80ies, the migration costs are proportional to the distance the server is migrated, as well as to the page size. Subsequently, the problem has been generalized to the well-known Metrical Task System (MTS) problem, which is also a generalization of many other online problems on graphs (e.g., the $k$-Server Problem). In a metrical task system, costs for satisfying a request (or more precisely: performing a task) in a certain state (e.g., for a server being at a certain node) are given as well as a metric space for distances between states.

Traditionally, these problems are motivated by shared memory management in networks. A page can be moved (or *migrated*[1]) to different computers in the network, sharing the same address space. The problem is also encountered as the *file migration* problem in a distributed network. Instead of pages, files are migrated. With the rise of the world wide web, where requests are supposed to be fulfilled in real time, the problem has received additional attention. Another (recent) application is the *virtual server migration* (in [1]) in the context of cloud computing [15] and the novel network virtualization paradigms. Opposed to simple pages, entire services, i.e. programs with data, are moved across the network. Moreover, these migration services may be offered to service providers (or directly to users) by infrastructure owners (or resource brokers and resellers).

[1]We will treat the verbs *move* and *migrate* as synonyms.

These recent networking trends introduce more general migration costs that cannot be modeled with the existing problem formulations. For instance, consider a service provider offering a low-latency SAP application which is flexibly moved closer to the (potentially mobile) users. Without support of live migration technology, each migration may come at a service interruption cost which depends on the migration time. This cost is unlikely to depend on the distance the server is moved but rather on the available bandwidth along the migration path.

Interestingly, it turns out that obtainable competitive ratios critically depend on the migration cost model. We, in this paper, present a novel online algorithm for migration cost functions and we show its (asymptotic) optimality for the classic problem variants, i.e., for a scenario where migration costs depend on distances and also for the scenario where costs are determined by the number of migrations. In the latter case, we improve on the asymptotic competitive ratio of prior work. We also conjecture that our algorithm is (asymptotically) optimal for any other reasonable cost function (see Section III). In particular, our generalized migration model allows the costs of a migration to be dependent on an aggregate measure of prior migrations, i.e., the number of all migrations or the total distance the server has been migrated. For instance, a cloud or virtual network provider may provide discounts to good customers using much resources and migrating frequently. Thus, the state of an algorithm is characterized not only by its current server location but also by its prior movements, resulting in a state space that increases exponentially with the number of requests (this holds despite the fact that some convergence criteria apply due to the definition of the cost function; see Section III). Note that metrical task systems cannot deal with such a scenario due to an unbounded state space. Therefore, our problem is not merely an instance of a MTS but rather pushes for an extension of MTS.

Our analysis differs from prior work in that it does not rely on potential functions. Thus, we believe that the techniques employed in our analysis are of independent interest.

## II. Related Work

The problem of page migration has been investigated already in the 70ies in the context of placing a database in a computer network, e.g. [12]. Whereas initial research focused more on a mathematical formulation and solution of the problem, e.g. through integer linear programs, the first algorithmic treatment for larger instances was undertaken in the 80ies by [13]. Typically, some kind of usage pattern was assumed and treatment was mainly based on heuristics. The algorithmic study abandoning any assumption on usage patterns employing competitive analysis was established in [8]. More precisely, the page migration problem together with the page replication problem has been introduced by [8]. For the replication problem pages are not migrated but *copied*; a copied page cannot be deleted. Efficient algorithms were derived for trees and complete graphs. After a sequence of improvements, constant-competitive solutions have emerged, i.e. [2] gives a 7-competitive deterministic solution. This was later improved by [5] which attains a competitive ratio of 4.086. On the other hand, [11] shows a lower bound of roughly 3.148. The concept of copying the behavior of an optimal algorithm is common for many online algorithms. The seemingly natural choice of moving to a center of gravity, which minimizes the access costs for a set of requests, has already been introduced in [2] with the so called *Move-To-Min* algorithm and has been used also in other work [6]. However, the decision when to move the server is different for most algorithms, in particular for ours. The analysis in [2] and [5] heavily rely on the linearity of the migration cost function $g$ in the distance migrated, i.e. $g(d) = D \cdot d$ for some fixed $D \geq 1$. Both papers rely on a commonly encountered technique for analysis of online algorithms, namely potential functions. Our analysis is different. It refrains from using a potential function argument and rather minimizes sums of interdependent variables, e.g., see Lemma V.9.

However, if costs are dependent on the number of migrations only, i.e. in the so-called *virtual server migration* [6] problem then the same paper [6] shows a non-constant lower bound of $\Omega(\log n / \log\log n)$ where $n$ is the number of nodes in the network, which follows from the related problem of *online function tracking* [7], [16]. We achieve asymptotic optimality for both cost models. In particular, we improve upon [6], which gives a randomized $O(\log n)$-competitive algorithm, and [1] which also describes a deterministic algorithm yielding a competitive ratio of $O(\log n)$. In [1] each node $v$ maintains a counter $c(v)$ that is incremented with every request $r$ by the distance $d(v, r)$ between node $v$ and request location $r$. A node is called active as long as $c(u) < \beta/40$ for some threshold $\beta$ describing the migration cost (and being at least the diameter of the graph). Once the counter $c(u)$ of $u$ reaches the threshold $\beta$ the server is migrated to an active node $u$, such that node $u$ is a center of gravity for all active nodes $\{w \in V | c(w) < \beta/40\}$. Once there are no active nodes left, all counters are reset to 0. The analysis shows that with every movement of the server a constant fraction of nodes become inactive, i.e. after $O(\log n)$ movements each costing $\beta$ all nodes must have become inactive. It is also shown that the optimal offline algorithm must move its server at least once or incurs access costs at least $\Omega(\beta)$, yielding a competitive ratio of $O(\log n)$. The virtual server migration problem differs from the page migration problem in the sense that the cost of migration depends on the available bandwidth along the migration path, rather than on the path length.

The page migration problem is related to many other problems. The *facility location problem* [14] asks for optimal facility locations such that facility building costs and access costs are minimized; the facilities cannot be migrated. In the *k-server problems* (see, e.g., [9] and in particular [3] for a construction achieving a polylogarithmic competitive ratio), an online algorithm must control the migration of a set of $k$ servers, represented as points in a metric space, and handle requests that come also in the form of points in the space. As each request arrives, the algorithm must determine which server to move to the requested point. The goal of the algorithm is to reduce the total distance that all servers traverse.

The virtual server migration problem, the facility location problem, the $k$-server problem and the page migration problem are all instances of *metrical task systems (MTS)* (e.g., [4], [10]). For metrical task systems, there exist (relatively complex) asymptotically optimal deterministic $\Theta(n)$-competitive algorithms, where $n$ is the state (or "configuration") space, and randomized $O(\log^2 n \log\log n)$-competitive algorithms given that the state space fulfills the triangle inequality. Assuming that $n$ is given by the number of nodes, i.e. independent of prior migrations, then a general deterministic solution is at least exponentially worse than our competitive algorithm ranging from being $\Theta(1)$ to being $\Theta(\log n / \log\log n)$ competitive. However, if the algorithm's state is determined not only by its current server location but also by decisions made in the past, i.e. prior migrations, the state space grows with every possibility of a migration, i.e. with every request. Thus, in this case the competitive ratio becomes unbounded for state-of-the-art MTS algorithms.

## III. Model and Definitions

Consider an undirected graph $G = (V, E)$ with a distance function $d : V \times V \to \Re$, e.g. a physical network or a substrate network on which virtual services are offered. For two nodes $u, v \in V$ $d(u, v)$ gives the length of the shortest path on $G$. We assume that $d$ is a metric, in particular that the triangle inequality holds, i.e. $d(u, v) \leq d(u, w) + d(w, v)$ for any nodes $u, v, w \in V$. There is a single server positioned at some node $u \in V$. The server must serve a sequence of (potentially infinitely many) requests by nodes $\sigma := (r_0, r_1, ...,)$ with $r_i \in V$. For convenience, we use *request* $r_i$ to denote node $r_i \in V$ issuing the $(i+1)^{st}$ request. A subsequence $\sigma(l)$ consists of the first $l$ requests of $\sigma$. The cost of a request $r \in \sigma$ is given by the distance $d(u, r)$. Let the *access costs* $C$ for a sequence $R$ of requests for a server positioned at node $u$ be $C(u, R) := \sum_{r \in R} d(u, r)$. The server can be migrated from any node $u$ to any other node $v \neq u$ before a request is issued. The objective is to strike a balance between access costs and migration costs. The function $g(x|y)$ defines the cost of moving the server, where $x$ and $y$ can have two different meanings. Costs can be based on the distance $x$ the server is moved given that it has already been moved for distance $y$. Costs can also be determined by the number of movements. In this case $g(1|y)$ denotes the costs to move the server (to an arbitrary node) given $y$ movements have been performed so far. We assume that the function $g(x|y)$ behaves like real world costs in a typical consumer-producer relationship, and may e.g., describe *discounts* for frequent customers. This means that migrating more does never increase marginal costs, but it might *decrease* it. More precisely, $g(x|y)$ is monotonically decreasing with $x$ and $y$, i.e.,

$$\forall z + w \le x + y : g(x|y)/x \le g(z|w)/z \qquad (1)$$

We assume equality in (1) for $z + w = x + y$. This means that given a fixed number of migrations have been performed, the costs per distance of a migration (or per migration) are the same as long as the overall distance migrated (or overall number of migrations) is the same. We also require that it is always at least as expensive to migrate the server for distance $x$ as it is to answer a request being at distance $x$ from a server, i.e. $\forall y : g(x|y) \ge x$. In particular, for migration costs based on the number of movements this implies that the most expensive migration between two nodes of maximal distance, i.e. $\max_{a,b \in V} d(a,b)$ (= diameter of the graph), is a lower bound on the cost of a migration, i.e. $g(1|y) \ge \max_{a,b \in V} d(a,b)$. If only $g(x)$ is stated without conditioning on another variable $y$ then the costs depend only on $x$.

We assume a conservative perspective, i.e. nothing is known about requests in $\sigma$ occurring in the future. An online algorithm ON has to decide whether or not to migrate the server only based on already served requests. We are in the realm on online algorithms and competitive analysis. Our goal is to minimize the (strict) competitive ratio $\rho$, i.e., the overall cost of the online algorithm ON divided by the overall cost of an optimal offline algorithm OFF (knowing $\sigma$ in advance).

The total costs of our algorithm ON are the sum of its total access costs $C_A^{\text{ON}}(\sigma)$ and its total server migration cost $C_M^{\text{ON}}(\sigma)$, i.e. $C_A^{\text{ON}}(\sigma) + C_M^{\text{ON}}(\sigma)$. Analogously for OFF the total costs are $C_A^{\text{OFF}}(\sigma) + C_M^{\text{OFF}}(\sigma)$.

$$\rho := \max_\sigma \frac{C(\text{ON})}{C(\text{OFF})} = \max_\sigma \frac{C_A^{\text{ON}}(\sigma) + C_M^{\text{ON}}(\sigma)}{C_A^{\text{OFF}}(\sigma) + C_M^{\text{OFF}}(\sigma)}$$

Initially, ON's server and OFF's server are located at the same node, i.e. $f_0$. Let $(f_0, f_1, ..., f_{s_{\text{ON}}})$ be the initial server position $f_0$ concatenated with the sequence of $s_{\text{ON}}$ nodes (in chronological order) where an online algorithm ON places the server during the execution such that it replies to at least one request for each node $f_i$ with $i \in [0, f_{s_{\text{ON}}} - 1]$ . Let $f_{s_{\text{ON}}}$ be its last position after the last migration once all requests $\sigma$ have been handled. Define $d_{avg}(v,R) := \sum_{r \in R} d(v,r)/|R|$ for a subsequence of requests $R \subseteq \sigma$. The *center(s) of gravity* $CG(R) \subseteq V$ are defined as the closest nodes $U \subseteq V$ for which $C(u,R) := \sum_{r \in R} d(u,r) = |R| \cdot d_{avg}(u,R)$ is minimal among all nodes $V$. Frequently, there is just a single minimum, i.e. $u := CG(R)$ for a node $u \in V$.

We abbreviate the cost based on the number of migrations as *co.nb.m.* and the costs based on the distance as *co.di.*.

Let $F_i$ be all requests handled while the server was at position $f_i$. Let $o_0^i$ be OFF's server position for handling the request immediately preceding the first request in $F_i$ (for $i > 0$) and OFF's initial server position for $i = 0$. Assume OFF migrated from $o_0^i$ to $o_1^i$ and so on until $o_m^i$, where it handled the last request in $F_i$. Let $O_j^i \subseteq F_i$ be the sequence of requests that OFF handled at node $o_j^i$. By definition $\cup_{i \in [1,m]} O_j^i = F_i$. Let $o_c^i$ be a closest server position of OFF to $f_{i+1}$ and let $o_f^i$ be a furthest server position of OFF to $f_{i+1}$, i.e. $o_c^i := \arg\min_{k \in [0,m]} d(o_k^i, f_{i+1})$ and $o_f^i := \arg\max_{k \in [0,m]} d(o_k^i, f_{i+1})$. Define $d_o^i := \sum_{k \in [1,m]} d(o_{k-1}^i, o_k^i)$.

## IV. ALGORITHM

In our online algorithm *Follower* each request $r$ is added to set $F_i$ for a server being at $f_i$. To compute the next server location $f_{i+1}$ the algorithm only takes into account requests $F_i$ served from its (server's) current position $f_i$. The node $f_{i+1}$ is a closest node which lies at a center of gravity $CG$ of $F_i$. The server is migrated to $f_{i+1}$ as soon as access costs $C(f_i, F_i)$ match at least the costs for migration to $f_{i+1}$. If the current server position $f_i$ is a center of gravity $f_i \in CG(F_i)$ then all requests $F_i$ are "forgotten" by incrementing $i' := i + 1$ and adding upcoming requests to the next (empty) $F_{i'}$. In this case $f_{i'} = f_i$.

In fact, to ensure a balance between access and movement costs the server might perform an intermediate migration to a node where no requests are handled. More precisely, instead of moving the server directly to $f_{i+1}$ it may be moved from $f_i$ to $w_i \ne f_{i+1}$ and then only from there to $f_{i+1}$. The node $w_i$ is chosen such that the movement costs including the migration to node $w_i$ and then to $f_{i+1}$ match at least the access costs and exceed them as little as possible.[2] In the distance-based cost model, let $x_i^{\text{ON}}$ be the distance the server was moved by ON in the migration from $f_i$ to $f_{i+1}$ not including a potential detour to $w_i$, i.e., we have $x_i^{\text{ON}} := d(f_i, f_{i+1})$ with $i \in [0, s_{\text{ON}} - 1]$ (for convenience, $x_{s_{\text{ON}}}^{\text{ON}} := 0$). For co.nb.m. let $x_i^{\text{ON}} = 1 \quad \forall i$. For ON denote $k_i^{\text{ON}}$ as the aggregate distance of prior migrations and the number of prior migrations, respectively, before replying to any of the requests $F_i$. For co. di. we have $k_{i+1}^{\text{ON}} := \sum_{j=0}^{i} (d(f_j, w_j) + d(w_j, f_{j+1}))$ and $k_0 := 0$. If *Follower* performs a movement from node $f_i$ to $w_i$ and then a second movement onto $f_{i+1}$ then the migration costs for co.di. (and analogously for co.nb.m) are given by $g(d(f_i, w_i)|k_i^{\text{ON}})$ for the first movement and by $g(d(w_i, f_{i+1})|k_i^{\text{ON}} + d(f_i, w_i))$ for the second movement.

## V. ANALYSIS

The analysis is split into three parts: A general part that is valid for co.nb.m. and co.di., followed by a separate part for each type of cost function $g$.

[2]The motivation for this behavior is that otherwise OFF might move frequently avoiding high access costs and reap the benefits of discounts by moving a lot. *Follower* on the contrary would move very little but incurs still very high access costs. For simplicity of the algorithm, we assume that such a node $w_i$ always exists, otherwise we might just move several times.

**Algorithm *Follower***

$i := 0$; $k_0 := 0$ $\forall j$: $F_j = \{\}$ {The server starts at an arbitrary node $f_0$}
Upon a new request $r$ do:
Serve request $r$ with server at $f_i$
$F_i := F_i \cup r$
$f' :=$ arbitrary $u \in CG(F_i)$
$x' := d(f_i, f')$ {for `co.di.`, and $x' := 1$ for `co.nb.m.`}
**if** $C(f_i, F_i) \geq g(x'|k_i)$ **then**
  $f_{i+1} := f'$; $x_i := x'$
  $y(w) := d(f_i, w) + d(w, f_{i+1})$ {for `co.di.`, and for `co.nb.m.` $y(w) := 2$ for $w \neq f_{i+1}$ and $y(w) := 1$ otherwise }
  $slack(w \in V) := g(y(w)|k_i) - C(f_i, F_i)$
  $w_i$:= Node $w$ with minimum $slack(w)$ such that $slack(w) \geq 0$
  Move server to $w_i$ and if $w_i \neq f_{i+1}$ onto $f_{i+1}$
  $k_{i+1} := k_i + y(w_i)$
  $i := i + 1$
**end if**

### A. General

We prove that OFF does not gain from moving more than *Follower* for any cost function $g$. This is due to the fact that access and movement costs of *Follower* are tightly coupled.

We call *movement costs* $C_M^{\text{ON}}(F_i)$ for *Follower* the costs due to server migration(s) from $f_i$ (to $w_i$) onto $f_{i+1}$, i.e. $g(y(w_i)|k_i)$ in Algorithm *Follower* with $y(w_i) = d(f_i, w_i) + d(w_i, f_{i+1})$ for `co.di.` and for `co.nb.m.` $y(w_i) = 1$ for $w_i = f_{i+1}$ and $y(w_i) = 2$ otherwise. The total movement costs are $C_M^{\text{ON}}(\sigma) := \sum_{i \in [0, s_{\text{ON}}]} C_M^{\text{ON}}(F_i)$.

First, we prove an upper bound on the movement costs $C_M^{\text{ON}}(F_i)$ based on the access costs $C(f_i, F_i)$. As a byproduct we get a bound on the total costs $C_M^{\text{ON}}(\sigma)$.

**Lemma V.1.** *We have* $C_M^{\text{ON}}(F_i) \leq 3C(f_i, F_i)$*, and after the last movement of* Follower $C_A^{\text{ON}}(\sigma) \leq C_M^{\text{ON}}(\sigma) \leq 3C_A^{\text{ON}}(\sigma)$.

*Proof:* Consider the migration(s) from $f_i$ (to $w_i$) to $f_{i+1}$. When *Follower* moves we have that $C(f_i, F_i) \geq g(x_i^{ON}|k_i^{ON})$ by definition of the algorithm. After the movement from $f_i$ to $w_i$ onto $f_{i+1}$ we have (also by definition) that the access costs are less than the movement costs, i.e. $C(f_i, F_i) \leq g(y(w_i)|k_i^{ON}) = C_M^{\text{ON}}(F_i)$. *Follower* chooses $w_i$ s.t. $slack(w) := C_M^{\text{ON}}(F_i) - C(f_i, F_i) \geq 0$ is minimized for $w = w_i$ (but still non-negative). The slack can be chosen such that $slack(w_i) \leq g(2|k_i^{ON})$, since moving from $f_i$ to $w_i$ onto $f_{i+1}$ incurs a granularity of at most 2 by altering the distance of $w_i$ by 1 for `co.di.` By performing two movements of distance $d(f_i, w_i) + d(w_i, f_{i+1})$ this yields a total granularity of at most 2 for `co.di.` The granularity is two for `co.nb.m.` since we perform one additional movement by migrating to $w_i$. In other words, the movement costs $C_M^{\text{ON}}(F_i)$ are at most $C_M^{\text{ON}}(F_i) \leq C(f_i, F_i) + g(2|k_i^{ON}) \leq C(f_i, F_i) + g(1|k_i^{ON}) + g(1|k_i^{ON}+1) \leq C(f_i, F_i) + g(1|k_i^{ON}) + g(1|k_i^{ON})$ due to Eq. 1. The access costs $C(f_i, F_i)$ must be at least the cost for moving for distance one for `co.di.` and for one movement for `co.nb.m.`, i.e. $C(f_i, F_i) \geq g(1|k_i^{ON})$. This yields the upper bound of $C_M^{\text{ON}}(F_i) \leq 3C(f_i, F_i)$. For the total movement costs we have $C_M^{\text{ON}}(\sigma) = \sum_{i \in [0, s_{\text{ON}}]} C_M^{\text{ON}}(F_i) \leq \sum_{i \in [0, s_{\text{ON}}]} 3C(f_i, F_i) = 3C_A^{\text{ON}}(\sigma)$. In the same way the lower bound follows using $C(f_i, F_i) \leq C_M^{\text{ON}}(F_i)$. ■

**Lemma V.2.** *To maximize the competitive ratio any sequence of requests is chosen such that after the last request* Follower *moves its server to* OFF*'s server location.*

*Proof:* Assume OFF's server is at $f'$ and *Follower*'s at $f''$ after the last request with $f' \neq f''$. Then, we could add requests $R$ from the server position $f'$ of OFF. These requests $r = f' \in R$ incur no costs for OFF since $d(r, f') = 0$ but increase access costs for *Follower*, thus yielding a larger competitive ratio. Once both servers coincide, any requests issued yield the same (access) costs for both, i.e. drive the competitive ratio towards one. Therefore the competitive ratio is maximized if the last request $r$ of a sequence of requests results in *Follower* to migrate its server to OFF's server location. ■

Since OFF knows requests ahead of time it might, for example, always move to the location of the next request, and reap the benefit of discounts for migrations. We prove that overall OFF does not benefit (up to a constant factor) from discounts, since essentially *Follower* always spends roughly the same costs that it incurs for access also on movements. In other words, OFF might as well not migrate its server at all to maximize the competitive ratio asymptotically. First, we bound the competitive ratio for the case that OFF moved its server more than *Follower*.

**Lemma V.3.** *Assume* Follower *moved its server to* $f_{i+1}$ *after* $l$ *requests* $\sigma(l)$*. If* OFF *has moved its server more than* Follower *(in terms of distance or number of migrations), i.e.* $C_M^{\text{OFF}}(\sigma(l)) \geq C_M^{\text{ON}}(\sigma(l))$*, then for the competitive ratio holds* $\rho \leq 4$*.*

*Proof: Follower* moved to position $f_{i+1}$ due to requests $F_i \subseteq \sigma(l)$. Assume the distance moved (or the number of migrations, respectively) by OFF is larger than that of *Follower* for $\sigma(l)$, i.e. $C_M^{\text{OFF}}(\sigma(l)) \geq C_M^{\text{ON}}(\sigma(l))$. By Lemma V.1 we have $C_A^{\text{ON}}(\sigma(l)) \leq C_M^{\text{ON}}(\sigma(l)) \leq 3C_A^{\text{ON}}(\sigma(l))$. Thus $C_A^{\text{ON}}(\sigma(l)) + C_M^{\text{ON}}(\sigma(l)) \leq 4C_A^{\text{ON}}(\sigma(l)) \leq 4C_M^{\text{ON}}(\sigma(l))$, implying $C_M^{\text{OFF}}(\sigma(l)) > 1/4 \cdot 4C_M^{\text{ON}}(\sigma(l)) \geq 1/4(C_A^{\text{ON}}(\sigma(l)) + C_M^{\text{ON}}(\sigma(l)))$. Therefore, four times the movement costs of OFF is larger than the total costs of *Follower* for the sequence $\sigma(l)$. ■

### B. Costs Based on Number of Migrations

For `co.nb.m.` we prove that it suffices to analyze the situation where OFF does not move its server to bound the asymptotic competitive ratio. In particular, we can consider the scenario that *Follower* moves away from OFF and then moves back until OFF's and *Follower*'s server coincide. These statements together help us to establish a bound on the competitive ratio.

Let $\rho_S$ be the competitive ratio assuming that OFF's server stays at the initial node, i.e. OFF never moves its server. We show how to construct a set of requests such that the competitive ratio only changes asymptotically and OFF does not move.

**Lemma V.4.** *co.nb.m. achieves a competitive ratio* $\rho \leq \max(4, \rho_S)$.

*Proof:* Assume that the costs for OFF's movements are smaller than *Follower*'s, i.e. $C_M^{\text{OFF}}(\sigma(l)) < C_M^{\text{ON}}(\sigma(l))$. If OFF performs a single movement, i.e. $s_{\text{OFF}} > 0$, OFF already incurs at least the same migration costs as *Follower* occurs for migrating once since it does not benefit from higher discounts because OFF migrated less overall. Mathematically speaking, $C_M^{\text{OFF}}(\sigma(l)) < C_M^{\text{ON}}(\sigma(l))$ implies $k_i^{ON} \geq k_i^{OFF}$ and thus by assumption (1) about $g$: $g(1|k_i^{ON}) \leq g(1|k_i^{OFF})$. *Follower* performs at most two movements (from $f_i$ to $w_i$ to $f_{i+1}$): The last request $r' \in \sigma(l)$ results in at most the migration costs to any other node since $g(1|y) \geq \max_{a,b \in V} d(a,b)$, i.e. $d(f_i, r') \leq g(1|y)$. Before the last request $r'$ the access costs must be less than the migration costs (otherwise *Follower* had already migrated), i.e. $C(f_i, F_i \setminus r') \leq g(1|k_i^{ON})$. After the last request: $C(f_i, F_i) \leq g(1|k_i^{ON}) + d(f_i, r') \leq 2g(1|k_i^{ON})$. Therefore *Follower*'s migration costs are at most double the one's of OFF for $s_{\text{OFF}} > 0$. Furthermore, its access costs $C(f_i, F_i)$ are by definition less than the migration costs (from $f_i$ to $w_i$ to $f_{i+1}$). Putting the pieces together we have that *Follower*'s total costs are at most $4g(1|k_i^{ON})$ and OFF's costs are at least $g(1|k_i^{OFF}) \geq g(1|k_i^{ON})$. Thus, the competitive ratio in case $C_M^{\text{OFF}}(\sigma(l)) < C_M^{\text{ON}}(\sigma(l))$ and OFF moves at least once, i.e. $s_{\text{OFF}} > 0$, is at most 4. For $C_M^{\text{OFF}}(\sigma(l)) \geq C_M^{\text{ON}}(\sigma(l))$ using Lemma V.3 yields an overall ratio of 4. ∎

To maximize the competitive ratio given fixed access costs $C(f_i, F_i)$ for *Follower* the requests $F_i$ should be chosen such that the costs are minimum for OFF. The next lemma restates this criterion based on the average distance of a request $F_i$ to OFF's and *Follower*'s server location.

**Lemma V.5.** *If* OFF*'s server stays at* $f_0$ *then to bound* $\rho$ *the average distance for a request* $r \in F_i$ *to* $f_{i+1}$ *and* $f_0$ *is the same, i.e.* $d_{avg}(f_{i+1}, F_i) = d_{avg}(f_0, F_i) \geq d(f_0, f_{i+1})/2$.

*Proof:* Say *Follower* moves from $f_i$ to $f_{i+1}$. Since $f_{i+1}$ is a center of gravity having minimal access costs for requests $F_i$, it only remains to minimize access costs for OFF being at $f_0$. Access costs for OFF are minimized, if OFF's server is at a center of gravity $CG(F_i)$, i.e. $CG(F_i) \supseteq \{f_{i+1}, f_0\}$. Thus we have $C(F_i, f_0) = C(F_i, f_{i+1})$. Therefore, $C(F_i, f_0) = \sum_{r \in F_i} d(r, f_0) = |F_i| d_{avg}(f_0, F_i)$. Analogously, $C(F_i, f_{i+1}) = |F_i| d_{avg}(f_{i+1}, F_i)$. Therefore, $d_{avg}(f_0, F_i) = d_{avg}(f_{i+1}, F_i)$. Furthermore, for any request $r \in F_i$ we have using the triangle inequality $d(f_0, r) + d(f_{i+1}, r) \geq d(f_0, f_{i+1})$. Therefore, also $d_{avg}(f_0, F_i) + d_{avg}(f_{i+1}, F_i) = 2d_{avg}(f_{i+1}, F_i) \geq d(f_0, f_{i+1})$ and $d_{avg}(f_0, F_i) = d_{avg}(f_{i+1}, F_i) \geq d(f_0, f_{i+1})/2$. ∎

**Lemma V.6.** *The competitive ratio of co.nb.m. is*

$$\rho \leq \max_{\sigma} 128 \frac{\sum_{i \in [0, s_{\text{ON}}]} g(x_i^{\text{ON}}|k_i^{\text{ON}})}{\sum_{i \in [0, s_{\text{ON}}]} g(x_i^{\text{ON}}|k_i^{\text{ON}}) \cdot \frac{d(f_0, f_{i+1})}{d(f_0, f_i) + d(f_i, f_{i+1})}}$$

*and* $d(f_i, f_{i+1}) = d(f_i, f_0) + d(f_0, f_{i+1})$ *as well as* $g(x_i^{\text{ON}}|k_i^{\text{ON}}) = C(f_i, F_i)$.

*Proof:* Using the triangle inequality, i.e. $d(f_i, r) \leq d(f_i, f_{i+1}) + d(f_{i+1}, r)$, we have for *Follower* $C(f_i, F_i) \leq |F_i| \cdot d(f_i, f_{i+1}) + C(f_{i+1}, F_i) \leq |F_i| \cdot d(f_i, f_{i+1}) + C(f_0, F_i)$. The last inequality follows, since $f_{i+1}$ is a center of gravity for $F_i$ and thus any node $f'$ (in particular $f_0$) must have at least the same access costs, i.e. $C(f', F_i) \geq C(f_{i+1}, F_i)$. For the total costs for handling requests $F_i$ and moving from $f_i$ (to $w_i$) onto $f_{i+1}$ we have due to Lemma V.1: $C(f_i, F_i) + C_M^{\text{ON}}(F_i) \leq 4C(f_i, F_i)$. We use these facts to maximize the competitive ratio $\rho$. We assume OFF does not migrate and account for this by incoporating a factor of 4 due to Lemma V.4 in the first inequality.

$$\begin{aligned}
\rho &= \max_{\sigma} \frac{C_A^{\text{ON}}(\sigma) + C_M^{\text{ON}}(\sigma)}{C_A^{\text{OFF}}(\sigma) + C_M^{\text{OFF}}(\sigma)} \leq \max_{\sigma} 4 \frac{C_A^{\text{ON}}(\sigma) + C_M^{\text{ON}}(\sigma)}{C_A^{\text{OFF}}(\sigma)} \\
&\leq \max_{\sigma} 16 \frac{\sum_{i \in [0, s_{\text{ON}}]} C(f_i, F_i)}{\sum_{i \in [0, s_{\text{ON}}]} C(f_0, F_i)} \\
&\leq \max_{\sigma} 16 \frac{\sum_{i \in [0, s_{\text{ON}}]} (|F_i| \cdot d(f_i, f_{i+1}) + C(f_0, F_i))}{\sum_{i \in [0, s_{\text{ON}}]} C(f_0, F_i)}
\end{aligned}$$

To maximize $\rho$ we can minimize $C(f_0, F_i) = |F_i| d_{avg}(f_0, F_i) \geq |F_i| d(f_0, f_{i+1})/2$ due to Lemma V.5. Next we derive a lower bound on $|F_i|$ using the triangle inequality $d(f_i, f_{i+1}) \leq d(f_i, f_0) + d(f_0, f_{i+1})$ and the condition that *Follower* moves only if $g(x_i^{\text{ON}}|k_i^{\text{ON}}) \leq C(f_i, F_i)$:

$$\begin{aligned}
g(x_i^{\text{ON}}|k_i^{\text{ON}}) &\leq C(f_i, F_i) \leq |F_i| \cdot d(f_i, f_{i+1}) + C(f_0, F_i) \\
&\leq |F_i| \cdot (d(f_i, f_{i+1}) + d(f_{i+1}, f_0)/2) \\
&\leq |F_i| \cdot (d(f_i, f_0) + d(f_0, f_{i+1}) + d(f_{i+1}, f_0)/2) \\
&= |F_i| \cdot (d(f_i, f_0) + 3d(f_{i+1}, f_0)/2)
\end{aligned}$$

Therefore $|F_i| \geq \frac{g(x_i^{\text{ON}}|k_i^{\text{ON}})}{d(f_i, f_0) + 3d(f_{i+1}, f_0)/2}$. To maximize $\rho$ we minimize $|F_i|$, i.e. assume equality, which also implies equality for $g(x_i^{\text{ON}}|k_i^{\text{ON}})$ and $C(f_i, F_i)$. To minimize $F_i$, i.e. $C(f_0, F_i)$ and to maximize the nominator we can assume $d(f_i, f_{i+1}) = d(f_i, f_0) + d(f_0, f_{i+1})$. We use the term for $|F_i|$ in the denominator and later in the nominator of $\rho$:

$$\begin{aligned}
C(f_0, F_i) &\geq |F_i| d(f_{i+1}, f_0)/2 = \frac{g(x_i^{\text{ON}}|k_i^{\text{ON}}) d(f_{i+1}, f_0)/2}{d(f_i, f_0) + 3d(f_{i+1}, f_0)/2} \\
&\geq \frac{g(x_i^{\text{ON}}|k_i^{\text{ON}})/3 \cdot d(f_{i+1}, f_0)}{d(f_i, f_0) + d(f_{i+1}, f_0)}
\end{aligned}$$

Next, to maximize the competitive ratio we use $|F_i| = \frac{g(x_i^{\text{ON}}|k_i^{\text{ON}})}{d(f_i, f_0) + 3d(f_{i+1}, f_0)/2} \leq \frac{g(x_i^{\text{ON}}|k_i^{\text{ON}})}{d(f_i, f_{i+1})}$ in the nominator. We also substitute $C(f_0, F_i)$ by $\frac{g(x_i^{\text{ON}}|k_i^{\text{ON}})/3 \cdot d(f_{i+1}, f_0)}{d(f_i, f_0) + d(f_{i+1}, f_0)}$ in the denominator and by $g(x_i^{\text{ON}}|k_i^{\text{ON}})/3$ in the nominator, yielding (leaving aside $\max$ and all $\sum_{i \in [0, s_{\text{ON}}]}$ for readability):

$$
\begin{aligned}
& 16(|F_i| \cdot d(f_i, f_{i+1}) + C(f_0, F_i))/C(f_0, F_i) \\
\leq\ & 16 \frac{\frac{g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}})}{d(f_i,f_{i+1})} \cdot d(f_i, f_{i+1}) + g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}})/3}{g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}})/3 \cdot \frac{d(f_{i+1},f_0)}{d(f_i,f_0)+d(f_{i+1},f_0)}} \\
\leq\ & 128 \frac{g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}})}{g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}}) \cdot \frac{d(f_{i+1},f_0)}{d(f_i,f_0)+d(f_{i+1},f_0)}}
\end{aligned}
$$

■

We define a *lazy phase* consisting of three consecutive parts: First, *Follower* is at the same node $f_0$ as OFF (or just moved there without having answered any request yet). Second, *Follower* moves once to another node $u \neq f_0$. Third, with every migration to a node $f_i$ where requests are handled *Follower* gets closer to OFF's server until both coincide again at $f_0$. Thus, by definition a lazy phase consists of at least two migrations, i.e. $s^{\mathrm{ON}} \geq 2$.

**Lemma V.7.** *If* OFF*'s server stays at $f_0$ then for the competitve ratio $\rho'$ for a sequence of lazy phases holds: $\rho \leq 2\rho'$*

*Proof:* In the beginning OFF's and *Follower*'s server are at the same node $f_0$ by assumption. If none of them moves the competitive ratio would be 1, since both incur the same (access) costs. Thus, eventually one of them moves. By assumption OFF never moves. Due to Lemma V.2 *Follower*'s last server position $f'$ is the same as OFF's server position at $f_0$, i.e. $f' = f_0$. Thus, *Follower* performs at least two movements.

Assume that *Follower*'s and OFF's server do not coincide before *Follower* increases the distance to OFF's server (again). Say *Follower*'s server is at $f_i$ and is moved to $w_i$ onto $f_{i+1}$ due to requests $F_i$ while OFF remains at $f_0$ such that $0 < d(f_0, f_i) < d(f_0, f_{i+1})$.

Using Lemma V.6 we can assume that $d(f_i, f_{i+1}) = d(f_i, f_0) + d(f_0, f_{i+1})$ and $C(f_i, F_i) = g(x_i^{ON}|k_i^{ON})$, i.e. the access and migration costs for *Follower* for handling requests $F_i$ amount to $2g(x_i^{ON}|k_i^{ON})$. The equality between movement and access costs, i.e. $C(f_i, F_i) = g(x_i^{ON}|k_i^{ON})$, implies that *Follower* moves directly from $f_i$ to $f_{i+1}$ without a detour to $w_i$. The total costs when moving first to $f_0$ due to requests $F_i'$ and then onto $f_{i+1}$ due to requests $F_0'$ amount to $2g(1|k_i^{ON}) + 2g(1|k_i^{ON} + 1)$.

For OFF the access costs are $C(f_0, F_i) = |F_i| d_{avg}(f_0, F_i)$. Using Lemma V.5 we can assume $d_{avg}(f_0, F_i) = d(f_0, f_{i+1})/2$ to minimize $C(f_0, F_i)$. Thus, $|F_i| = g(x_i^{ON}|k_i^{ON})/(d(f_i, f_0) + d(f_0, f_{i+1})/2)$ and OFF's costs are $|F_i| \cdot d(f_0, f_{i+1})/2$. In case *Follower* migrates first to $f_0$, OFF incurs no access costs for $r \in F_i'$, since each request $r$ can be issued from $f_0$, i.e. $r = f_0$, to make *Follower* move to $f_0$. To make *Follower* move to $f_{i+1}$ we can issue all requests $r \in F_0'$ from $f_{i+1}$, i.e. $r = f_{i+1}$. OFF incurs access costs $|F_0'| \cdot d(f_0, f_{i+1})$. Next, we use $g(x|y)/x = g(z|w)/z$ for some values $z + w = x + y$:

$$
\begin{aligned}
|F_i| &= g(d(f_i, f_0) + d(f_0, f_{i+1})|k_i^{ON})/(d(f_i, f_0) + d(f_0, f_{i+1})/2) \\
&\geq g(d(f_i, f_0) + d(f_0, f_{i+1})|k_i^{ON})/(d(f_i, f_0) + d(f_0, f_{i+1})) \\
&= g(d(f_0, f_{i+1})|k_i^{ON} + d(f_i, f_0))/d(f_0, f_{i+1}) = |F_0'|
\end{aligned}
$$

Thus, if *Follower* migrates to $f_0$ and then onto $f_{i+1}$ then the access costs $|F_0'|d(f_0, f_{i+1})$ for OFF are at most twice the access costs $|F_i| \cdot d(f_0, f_{i+1})/2$ when *Follower* moves directly to $f_{i+1}$.

■

**Lemma V.8.** *If no discounts are given, i.e., $g(x|y) = \beta \cdot x$, the competitive ratio can be bounded by considering any lazy phase:*

$$
\rho \leq \max_\sigma \frac{128 \sum_{j \in [0, s_{\mathrm{ON}}-1]} x_i^{\mathrm{ON}}}{\sum_{i \in [0, s_{\mathrm{ON}}-2]} x_i^{\mathrm{ON}} \cdot \frac{d(f_{i+1},f_0)}{d(f_0,f_i)+d(f_0,f_{i+1})}}
$$

*Proof:* Due to Lemma V.7 we can consider a sequence of lazy phases (increasing the competitive ratio by at most 2) with $d(f_i, f_{i+1}) = d(f_i, f_0) + d(f_0, f_{i+1})$. Lazy phases are independent, since $g(x|y) = \beta \cdot x$, i.e. there is no dependence on prior migrations and a lazy phase starts and ends with the server being at the same node. Due to the independence of lazy phases the maximum competitive ratio is the same for any phase. Thus going through $t > 0$ phases, where each phase has the same competitive ratio of say $c$, also yields an overall competitive ratio of $c$. Plugging $g(x|y) = \beta \cdot x$ into the term of Lemma V.6 and accounting for a factor of 2 due to using lazy phases (Lemma V.7) yields:

$$
\begin{aligned}
\rho &\leq \max_\sigma \frac{128 \sum_{j \in [0, s_{\mathrm{ON}}-1]} g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}})}{\sum_{i \in [0, s_{\mathrm{ON}}-2]} g(x_i^{\mathrm{ON}}|k_i^{\mathrm{ON}}) \cdot \frac{d(f_{i+1},f_0)}{d(f_0,f_i)+d(f_0,f_{i+1})}} \\
&= \max_\sigma \frac{128 \sum_{j \in [0, s_{\mathrm{ON}}-1]} x_i^{\mathrm{ON}}}{\sum_{i \in [0, s_{\mathrm{ON}}-2]} x_i^{\mathrm{ON}} \cdot \frac{d(f_{i+1},f_0)}{d(f_0,f_i)+d(f_0,f_{i+1})}}
\end{aligned}
$$

■

The following two theorems state that if no discounts based on the number of prior migrations are given, i.e., $g(1|y) = \beta \geq n$ then Algorithm *Follower* is $O(\log n / \log \log n)$ competitive. We start the analysis with a theorem stating the minimum of a sum involving expressions of numbers $x_i \in [1, n]$ with $i \in [1, s]$ for parameters $s$ and $n$.

**Lemma V.9.** *The term $\sum_{i \in [1,s]} x_{i+1}/(x_i + x_{i+1})$ for a given $x_i \in [1, n]$ depending only on $i, n, s$ and $s \geq 2$ is minimized for $x_i = n^{1-i/s}$ and is at least $s/(2n^{1/s})$.*

*Proof:* The sum $\sum_{i \in [1,s]} x_{i+1}/(x_i + x_{i+1})$ contains variables $x_i$ for $i \in [1, s]$ which potentially depend on $i, n$ and $s$, i.e., $x_i$ is fully characterized by these three values. There is no dependence of $x_i$ on $x_j$ for $i \neq j$. Therefore, we can compute the partial derivatives $\frac{\partial}{\partial x_i} \sum_{i \in [1,s]} \frac{x_{i+1}}{x_i + x_{i+1}} = (((x_i)^2 x_{i+1} + (x_{i-1})^2 x_{i+1} - x_{i-1}(x_{i+1})^2 - x_{i-1}(x_i)^2)) / (((x_{i-1} + x_i)^2 (x_i + x_{i+1})^2))$. (Note, for $i = 1$ we can remove all terms $x_{i-1}$.) We are looking for values of variables $x_i$ such that the nominator is 0, i.e.:

$$0 = ((x_i)^2 x_{i+1} + (x_{i-1})^2 x_{i+1} - x_{i-1}(x_{i+1})^2 - x_{i-1}(x_i)^2)$$
$$\Leftrightarrow 0 = (x_i)^2(x_{i+1} - x_{i-1}) + (x_{i-1})^2 x_{i+1} - x_{i-1}(x_{i+1})^2$$
$$\Leftrightarrow (x_i)^2 = -\frac{(x_{i-1})^2 x_{i+1} - x_{i-1}(x_{i+1})^2}{x_{i+1} - x_{i-1}}$$
$$\Leftrightarrow x_i = \pm\sqrt{-\frac{(x_{i-1})^2 x_{i+1} - x_{i-1}(x_{i+1})^2}{x_{i+1} - x_{i-1}}}$$

Since we have that $x_i \in [1, n]$ we do not have to consider the negative solution. Substituting $x_i = n^{1-i/s}$ yields that the partial derivatives are 0:

$$n^{1-i/s} = \pm\sqrt{-\frac{n^{3(1-i/s)}(n^{1/s} - n^{-1/s})}{n^{1-i/s}(n^{-1/s} - n^{1/s})}}$$
$$\Leftrightarrow \quad n^{1-i/s} = \pm n^{1-i/s} = n^{1-i/s} \text{ (since we require } x_i > 0)$$

To show that this is indeed a minimum, we require the second derivatives $\frac{\partial^2}{\partial(x_i)^2}\sum_{i\in[1,s]}\frac{x_{i+1}}{x_i + x_{i+1}}$ to be larger than zero at $x_i = n^{1-i/s}$. For the nominator of the second derivative we get: $2(-x_{i-1}x_i^3 - 3x_{i-1}x_i x_{i+1}^2 - x_{i-1}x_{i+1}^3 + x_i^3 x_{i+1} + x_{i-1}^3 x_{i+1} + 3x_{i-1}^2 x_i x_{i+1})$.

The denominator equals $(x_{i-1} + x_i)^3(x_i + x_{i+1})^3$. The denominator contains only positive terms. Therefore it is positive. Substituting $x_i = n^{1-i/s}$ in the nominator for each term and removing factor $n^{4-4i/s}$ from all terms (i.e. putting it in front) yields: $2n^{4-4i/s}(-n^{1/s} - 3n^{-1/s} - n^{-2/s} + n^{-1/s} + n^{2/s} + 3n^{1/s})$. Since $2n^{4-4i/s} > 0$ we can further simplify: $2(n^{1/s} - n^{-1/s}) - n^{-2/s} + n^{2/s}$. Suppose the worst case graph was of diameter one, i.e. $n = 1$ then $x_i \le 1$ for any $i$ and $\sum_{i\in[1,s]} x_{i+1}/(x_i + x_{i+1}) \le s/2 = s/(2n^{1/s})$. Assume that the graph is of diameter at least two, i.e. $x_i \in [1, n]$ with $n > 1$. For any $s > 0$ we have $n^{1/s} > 1$ (and $n^{-1/s} < 1$). Therefore, the term $2(n^{1/s} - n^{-1/s}) - n^{-2/s} + n^{2/s}$ is also larger or equal to 0 and thus the second derivative is positive at $x_i = n^{1-i/s}$, yielding a minimum at this point. Therefore:

$$\sum_{i\in[1,s]} x_{i+1}/(x_i + x_{i+1}) \le \sum_{i\in[1,s]}\frac{n^{1-(i+1)/s}}{n^{1-i/s} + n^{1-(i+1)/s}} \le$$
$$\sum_{i\in[1,s]} n^{1-(i+1)/s}/(2n^{1-i/s}) \le \max_s \sum_{i\in[1,s]} 1/2n^{1/s} = s/(2n^{1/s})$$

■

Now we can prove a bound on the competitive ratio, if there is no discount for performing several migrations, i.e., $g(1|y)$ is independent of $y$.

**Theorem V.10.** *For $g(x|y) := \beta \ge n$ Algorithm* Follower *is $O(\log n/\log\log n)$ competitive.*

*Proof:* Using $x_i^{\text{ON}} = 1\ \forall i$ and due to Lemma V.8 we can maximize the following term to get the competitive ratio:

$$\rho \le \max_\sigma 128\frac{\sum_{j\in[0,s_{\text{ON}}-1]} x_i^{\text{ON}}}{\sum_{i\in[0,s_{\text{ON}}-2]} x_i^{\text{ON}}\cdot\frac{d(f_{i+1},f_0)}{d(f_0,f_i)+d(f_0,f_{i+1})}}$$
$$= \max_\sigma 128\frac{s_{\text{ON}}}{\sum_{i\in[0,s_{\text{ON}}-2]}\frac{d(f_{i+1},f_0)}{d(f_0,f_i)+d(f_0,f_{i+1})}}$$

To maximize the overall expression, we have to minimize the following sum (depending on $s_{\text{ON}}$) $\sum_{i\in[0,s_{\text{ON}}-2]} d(f_{i+1}, f_0)/(d(f_0, f_i) + d(f_0, f_{i+1}))$. By definition we have that for two migrations $f_i \ne f_{i+1}$. In particular due to Lemma V.8, we need to consider only one lazy phase. Thus, we can assume that only the first and last position of *Follower*'s server coincide with OFF's server. We have that the distances $d(f_i, f_0)$ for $i \in [1, s_{\text{ON}} - 1]$ are at least 1 and at most the diameter of the graph, i.e., $d(f_i, f_0) \in [1, n]$. We can use Lemma V.9 to bound the sum $\sum_{i\in[1,s_{\text{ON}}-2]} d(f_{i+1}, f_0)/(d(f_0, f_i) + d(f_0, f_{i+1}))$ by $s_{\text{ON}}/(2n^{1/s_{\text{ON}}})$. Lemma V.9 requires independence of the terms $d(f_i, f_0)$, which is the case since the distances $d(f_0, f_i)$ for any $i$ can be chosen independent of $d(f_0, f_j)$ with $j \ne i$. The first term of the sum $d(f_1, f_0)/(d(f_0, f_0) + d(f_0, f_1)) = 1$. Thus, we get:

$$\rho \le \max_s \frac{s}{1 + s/(2n^{1/s})} \le 1/2\max_s \frac{s\cdot n^{1/s}}{n^{1/s} + s}$$

For a term $a \cdot b/(a + b)$ holds $a \cdot b/(a + b) \le \min(a, b)$. Thus, we bound the term $\frac{s\cdot n^{1/s}}{n^{1/s}+s}$ by $\min(s, n^{1/s})$, which is maximized for $s = n^{1/s}$. The last equality is not solvable in closed form, i.e., it yields $s = \log n/\mathit{LambertW}(\log n)$, where $\mathit{LambertW}$ denotes the *Lambert* function. However, using $s = c\log n/\log\log n$ there exists a $c \in [1, 3]$ such that: $n^{1/s} = s$ with $n^{1/s} = 2^{\log n\cdot\log\log n/(c\log n)} = 2^{\log\log n/c} = (\log n)^{1/c}$. Thus, we want to find $c$ such that $(\log n)^{1/c} = c\log n/\log\log n$. For $c \le 1$ we have $(\log n)^{1/c} > c\log n/\log\log n$. For $c \ge 3$ we get $(\log n)^{1/c} < c\log n/\log\log n$. Thus, there exists $c \in [1, 3]$ such that $n^{1/s} = s$. We get:

$$\rho \le 1/2\max_s \frac{s\cdot n^{1/s}}{n^{1/s}+s} \le 1/2\min(s, n^{1/s}) \le c/2\cdot\log n/\log\log n$$

Therefore, $s \in O(\log n/\log\log n)$. ■

### C. Costs Based on Distance Migrated

In the first lemma we relate the average distance $d_{avg}(f_i, F_i)$ of a request in $F_i$ from the current server position $f_i$ to a center of gravity. Then we prove that if OFF moved at least a constant fraction of the average distance $d_{avg}(f_i, F_i)$, it also incurs a constant fraction of *Follower*'s total costs to handle requests $F_i$. This implies the next Lemma V.13 saying that in case the average distance to $f_{i+1}$ is (somewhat) larger than the one to $f_i$, OFF incurs a constant fraction of *Follower*'s costs to handle requests $F_i$. This is used in Lemma V.14 to show that either OFF incurs a constant fraction of *Follower*'s costs for $F_i$ or (eventually) the distance between *Follower*'s and OFF's server must shrink fast. This in turn allows to bound the competitive ratio.

**Lemma V.11.** *For `co.di.` and some subsequence $F' = (r_0, ...) \subseteq F_i = (r_0, r_1, ...)$ (starting from the first request $r_0$ in $F_i$) holds for any $u \in CG(F')\backslash f_i$: $d(f_i, u) \le 2d_{avg}(f_i, F')$.*

*Proof:* Assume $d(f_i, u) > 2d_{avg}(f_i, F')$. Using the reverse triangle inequality we have $C(u, F') = \sum_{r \in F'} d(r, u) \geq \sum_{r \in F'} |d(f_i, u) - d(f_i, r)| \geq \sum_{r \in F'} d(f_i, u) - d(f_i, r) = \sum_{r \in F'} d(f_i, u) - \sum_{r \in F'} d(f_i, r) > |F'| 2d_{avg}(f_i, F') - |F'| d_{avg}(f_i, F') = |F'| d_{avg}(f_i, F') = C(f_i, F')$. Thus the access costs at $u$ are larger than at $f_i$ and therefore $u \notin CG(F')$. ■

**Lemma V.12.** *For* `co.di.` *and the sequence $\sigma(l)$, i.e. the first $l$ requests of $\sigma$, assuming $C_M^{\text{OFF}}(\sigma(l)) \leq C_M^{\text{ON}}(\sigma(l))$ with $F_i \subseteq \sigma(l)$ and* OFF $d_o^i \geq d_{avg}(f_i, F_i)/32$, OFF *incurs costs at least $C(f_i, F_i)/1024$.*

*Proof:* Consider the set $F' := F_i \setminus r_{|F_i|-1}$, i.e. the set $F_i$ without the last request $r_{|F_i|-1} \in F_i$. Assume $4d_{avg}(f_i, F') \geq d_{avg}(f_i, F_i)$. Due to $|F'| = |F_i| - 1$, we have $2|F'| \geq |F_i|$ and:

$$C(f_i, F_i) = |F_i| d_{avg}(f_i, F_i) \leq 2|F'| 4d_{avg}(f_i, F') = 8C(f_i, F') \quad (2)$$

Since *Follower* has not migrated after requests $F'$ we have for any $u \in CG(F')$:

$$C(f_i, F') < g(d(f_i, u)|k_i^{ON}) \quad (3)$$

Since $F' \subset F_i$ and $d(r, f_i) \geq 0$ for any $r \in F_i$, we have $C(f_i, F') \leq C(f_i, F_i)$. Together with $2|F'| \geq |F_i|$ we get: $2d_{avg}(f_i, F_i) = 2C(f_i, F_i)/|F_i| \geq C(f_i, F')/|F'| = d_{avg}(f_i, F')$. This together with Lemma V.11 yields $d(f_i, u) \leq 2d_{avg}(f_i, F') \leq 4d_{avg}(f_i, F_i)$. Due to the fact that OFF migrates at most the same distance as *Follower*, i.e. $k_i^{OFF} \leq k_i^{ON}$, and by assumption (1) about $g$:

$$\forall x > 0 : g(x|k_i^{OFF}) \geq g(x|k_i^{ON}) \quad (4)$$

Furthermore:

$$\begin{aligned} C(f_i, F') &\leq g(d(f_i, u)|k_i^{ON}) \text{ (due to Eq. 3)} \\ &\leq g(4d_{avg}(f_i, F_i)|k_i^{ON}) \leq 4g(d_{avg}(f_i, F_i)|k_i^{ON}) \end{aligned}$$

Thus, if OFF moved $d_{avg}(f_i, F_i)/32$, it incurs costs:

$$\begin{aligned} g(d_{avg}(f_i, F_i)/32|k_i^{OFF}) &\geq \text{(due to Ass. 1)} \\ g(d_{avg}(f_i, F_i)/32|k_i^{ON}) &\geq \text{(due to Eq. 4)} \\ 1/32g(d_{avg}(f_i, F_i)|k_i^{ON}) &\geq \text{(due to Eq. 1)} \\ C(f_i, F')/128 &\geq \text{(due to Eq. 5)} \\ C(f_i, F_i)/1024 & \quad \text{(due to Eq. 2)} \end{aligned}$$

Assume $4d_{avg}(f_i, F') < d_{avg}(f_i, F_i)$. The access costs can be bounded by two terms. The first term is the costs due to requests $F' = F_i \setminus r_{|F_i|-1}$, i.e. $|F'| d_{avg}(f_i, F') \leq |F_i| d_{avg}(f_i, F')$, and the second term $d(r_{|F_i|-1}, f_i)$ is the costs due to the last request $r_{|F_i|-1}$. To bound the second term we use: $C(f_i, F_i) = |F_i| d_{avg}(f_i, F_i) = |F_i| d_{avg}(f_i, F') + |F_i|(d_{avg}(f_i, F_i) - d_{avg}(f_i, F'))$. Therefore, $d(r_{|F_i|-1}, f_i) \geq |F_i|(d_{avg}(f_i, F_i) - d_{avg}(f_i, F'))$. Furthermore, due to $4d_{avg}(f_i, F') < d_{avg}(f_i, F_i)$ it holds $|F_i|(d_{avg}(f_i, F_i) - d_{avg}(f_i, F')) \geq 3/4|F_i| d_{avg}(f_i, F_i)$, i.e. the costs for the last request are at least $3/4C(f_i, F_i)$. Together with the assumption $4d_{avg}(f_i, F') < d_{avg}(f_i, F_i)$ this implies that $d(r_{|F_i|-1}, f_i) \geq 3/4C(f_i, F_i)$ and also $d(r_{|F_i|-1}, f_i) \geq d_{avg}(f_i, F_i)$. If OFF moved for distance $d_o^i \geq 0$, it incurs costs $d(r_{|F_i|-1}, f_i) - d_o^i + g(d_o^i|k_i^{ON})$, where $d(r_{|F_i|-1}, f_i) - d_o^i$ are the costs due to the last request. In case $d_o^i \geq d_{avg}(f_i, F_i)/64$ but $d_o^i \leq d(r_{|F_i|-1}, f_i)/2$ the costs are at least $d(r_{|F_i|-1}, f_i)/2 \geq 3/8C(f_i, F_i)$. In case $d_o^i > d(r_{|F_i|-1}, f_i)/2$ the costs are also at least $3/8C(f_i, F_i)$, since by assumption about $g$ for $x \geq 0$: $g(x|k_i^{ON}) \geq x$. ■

**Lemma V.13.** *If $d_{avg}(f_i, F_i) < c \cdot d_{avg}(f_{i+1}, F_i)$ for a constant $c \in [1, 3]$ then* OFF *incurs costs at least $C(f_i, F_i)/1024$.*

*Proof:* If OFF moved more than $d_{avg}(f_{i+1}, F_i)/2 \geq d_{avg}(f_i, F_i)/32$ we can use Lemma V.12 to lower bound the cost by $C(f_i, F_i)/1024$. Thus, assume from now OFF moved less.

Consider a set of maximal cardinality of distinct pairs $P$ from $F_i$, i.e. for any $(r_1, r_2) \in P$ with $r_1, r_2 \in F_i$ we have $r_1 \neq r_2$ and for any $(r_1, r_2), (r_3, r_4) \in P$ holds $\{r_1, r_2\} \cap \{r_3, r_4\} = \{\}$. Additionally, for each $(r_1, r_2) \in P$: $d(r_1, r_2) \geq d_{avg}(f_{i+1}, F_i)/4 > d_{avg}(f_i, F_i)/(4c)$ (by assumption).

Assume $|P| \geq |F_i|/8$. On average for a pair $(r_1, r_2) \in P$ OFF incurs costs at least the average distance $d_{avg}(f_{i+1}, F_i)/4$ between the pair, minus the distance $d_{avg}(f_i, F_i)/32$ OFF moved at most: $d_{avg}(f_{i+1}, F_i)/4 - d_{avg}(f_i, F_i)/32 \geq d_{avg}(f_i, F_i)/12 - d_{avg}\ (f_i, F_i)/32 \geq d_{avg}(f_i, F_i)/24$.

Thus, OFF incurs costs $2|P| d_{avg}(f_i, F_i)/24 \geq |F_i|/4 d_{avg}(f_i, F_i)/24 \geq C(f_{i+1}, F_i)/96$. Assume $|P| < |F_i|/8$. Since each pair $(r_1, r_2) \in P$ consists by definition of two requests with $d(r_1, r_2) \geq d_{avg}(f_{i+1}, F_i)/4$, there must be a set of requests $U \subseteq F_i$ with $|U| \geq 3|F_i|/4$ such that :

$$d(r_1, r_2) < d_{avg}(f_{i+1}, F_i)/4 \ \forall r_1, r_2 \in U, |U| \geq 3|F_i|/4 \quad (5)$$

Then the total access costs to handle all requests $U$ for a server being at a center of gravity $u \in CG(U)$ can be bounded as follows: $C(u, U) \leq C(\forall r \in F_i, U) \leq 3|F_i|/4 d_{avg}(f_{i+1}, F_i)/4$. Assume there is a node $w \in CG(F_i)$ such that for the furthest request $r' \in U$ holds $d(w, r') \geq d_{avg}(f_{i+1}, F_i)/2 + x$ for some $x > 0$. Due to Eq. (5) for any request $r \in U$: $d(w, r) \geq d_{avg}(f_{i+1}, F_i)/4 + x$. In this case $C(w, F_i) = C(w, U) + C(w, F_i \setminus U) = |U| d_{avg}(w, U) + |F_i \setminus U| d_{avg}(w, F_i \setminus U) \geq |U|(d_{avg}(w, r)/4 + x) + |F_i \setminus U| d_{avg}(w, F_i \setminus U)$. Therefore, assume we position the server at a request $r \in U$ with $d(r, r') = d_{avg}(f_{i+1}, F_i)/2$ instead of $w \in CG(F_i)$ with $d(w, r') \geq d_{avg}(f_{i+1}, F_i)/2 + x$ for some $x > 0$. This reduces overall costs by at least $|U|x - |F_i \setminus U|x \geq |F_i|x$, since by Equ. (5): $|U| \geq 3|F_i|/4$ and for any two requests $r_1, r_2 \in U$: $d(r_1, r_2) \leq d_{avg}(f_{i+1}, F_i)/4$. Thus, for any $w \in CG(F_i)$ for the furthest request $r' \in U$ holds $d(w, r') \leq d_{avg}(f_{i+1}, F_i)/2$.

The requests $F_i \setminus U$ induce access costs at least $C(f_{i+1}, F_i \setminus U) = C(f_{i+1}, F_i) - C(f_{i+1}, U) \geq$

$C(f_{i+1},F_i) - 3|F_i|/4d_{avg}(f_{i+1},F_i)/4 \geq C(f_{i+1},F_i)/2$. We have $d_{avg}(f_{i+1},F_i \setminus U) \geq d_{avg}(f_{i+1},F_i)$. If OFF moves less than $d_{avg}(f_{i+1},F_i)/32$ it incurs on average $d_{avg}(f_{i+1},F_i \setminus U) - d_{avg}(f_{i+1},F_i)/32 \geq d_{avg}(f_{i+1},F_i \setminus U)/2$ per request $r \in F_i \setminus U$ and overall at least $C(f_{i+1},F_i \setminus U)/2 \geq C(f_{i+1},F_i)/4$. Overall OFF incurs costs $C(f_{i+1},F_i)/4 = |F_i|d_{avg}(f_{i+1},F_i)/4 \geq d_{avg}(f_{i+1},F_i)/(4c)|F_i| \geq C(f_i,F_i)/12$. ∎

**Lemma V.14.** *To handle requests $F_i$ OFF either incurs costs $\Omega(C(f_i,F_i) + g(d_o^i|k_i^{OFF}))$ or the following two conditions hold: $d(o_c^i,f_{i+1}) < 3d(f_i,f_{i+1})/4$ and $C(f_i,F_i) < 4g(d(f_i,f_{i+1})|k_i^{ON})$.*

*Proof:* The case $|F_i = \{r\}| = 1$ is straight forward, since then $f_{i+1} = r$ and $C(f_i,F_i) = g(d(f_i,f_{i+1})|k_i^{ON})$.

Assume $0 \leq d_{avg}(f_{i+1},F_i) \leq d(f_i,f_{i+1})/2$. Using the triangle inequality and the previous assumption we have $d_{avg}(f_i,F_i) \leq d(f_i,f_{i+1}) + d_{avg}(f_{i+1},F_i) \leq 3/2d(f_i,f_{i+1})$. If $d(o_c^i,f_{i+1}) > 3d(f_i,f_{i+1})/4$ then OFF incurs on average costs at least $d(o_c^i,f_{i+1}) - d_{avg}(f_i,F_{i+1}) \geq 3d(f_i,f_{i+1})/4 - d(f_i,f_{i+1})/2 \geq d(f_i,f_{i+1})/4$, i.e. total costs of at least $|F_i|d(f_i,f_{i+1})/4$. From $d_{avg}(f_i,F_i) \leq 3/2d(f_i,f_{i+1})$ follows $C(f_i,F_i) \leq 3/2d(f_i,f_{i+1})|F_i|$. Thus, OFF incurs costs at least $1/6C(f_i,F_i)$.

Assume $d_{avg}(f_{i+1},F_i) > d(f_i,f_{i+1})/2$. Due to the triangle inequality we have $d_{avg}(f_i,F_i) \leq d(f_i,f_{i+1}) + d_{avg}(f_{i+1},F_i) < 3d_{avg}(f_{i+1},F_i)$. Due to Lemma V.13 with $c = 3$ we get a lower bound of the access costs for OFF of $C(f_i,F_i)/1024$.

Assume $C(f_i,F_i) \geq 8g(d(f_i,f_{i+1})|k_i^{ON})$. Using Lemma we have $d(f_i,f_{i+1}) \leq 2d_{avg}(f_i,F_i)$. In other words, in case $C(f_i,F_i) = d_{avg}(f_i,F_i)|F_i| < |F_i|d(f_i,f_{i+1})/2$ OFF does not migrate to $f_{i+1}$ (but stays at $f_i$). Thus, $g(d(f_i,f_{i+1})|k_i^{ON}) \geq |F_i|d(f_i,f_{i+1})/2$. From $C(f_i,F_i) \geq 4g(d(f_i,f_{i+1})|k_i^{ON}) \geq 2|F_i|d(f_i,f_{i+1})$ follows $d_{avg}(f_i,F_i) \geq 2d(f_i,f_{i+1})$. Using the triangle inequality $d_{avg}(f_i,F_i) \leq d(f_i,f_{i+1}) + d_{avg}(f_{i+1},F_i)$, implying $d_{avg}(f_i,F_i) - d(f_i,f_{i+1}) \leq d_{avg}(f_{i+1},F_i)$ and $d(f_i,f_{i+1}) \leq d_{avg}(f_{i+1},F_i)$. Therefore, if $C(f_i,F_i) \geq 4g(d(f_i,f_{i+1})|k_i^{ON})$ we have $C(f_i,F_i)/1024$ costs for OFF. ∎

**Theorem V.15.** *For the competitive ratio for `co.di.` it holds $\rho \in O(1)$.*

*Proof:* Due to Lemma V.14 in case the distance between OFF and *Follower* decreases (or is) below 3/4 of $d(f_i,f_{i+1})$ after handling requests $F_i$ and $C(f_i,F_i) < 4g(d(f_i,f_{i+1})|k_i^{ON})$, i.e. $d(o_c^i,f_{i+1}) \leq 3/4d(f_i,f_{i+1})$, OFF potentially incurs no costs for $F_i$ and *Follower* incurs access costs at most $C(f_i,F_i) < 4g(d(f_i,f_{i+1})|k_i^{ON})$. Assume the distance between $f_i$ and $f_{i+1}$ was $d' := d(f_i,f_{i+1})$. Then the maximal number $n_m$ of migrations of *Follower* until both servers coincide at the same node can be upper bounded as follows. For each migration the distance shrinks by 3/4, which gives the following inequality until the distance between *Follower*'s and OFF's server is less than 1: $(3/4)^{n_m} \cdot d' < 1$, i.e. $n_m = (\log d')/\log(4/3)$. This amounts to access costs for *Follower* of $\sum_{j=0}^{n_m} 4g((3/4)^j d'|k_{i+j}^{ON}) \geq 4g(O(d')|k_i^{ON})$. This in turn corresponds to total costs of at most $5g(O(d')|k_i^{ON})$. Thus, the costs until both servers coincide after being at distance $d'$ are at most larger by a constant factor than the costs to separate them for distance $d'$. In case the distance between OFF and *Follower* decreases after handling requests $F_i$ but access costs for *Follower* are larger, i.e. $C(f_i,F_i) \geq 4g(d(f_i,f_{i+1})|k_i^{ON})$ and $d(o_c^i,f_{i+1}) \leq 3/4d(f_i,f_{i+1})$, OFF incurs costs proportional to the access costs $C(f_i,F_i)$ of *Follower* and also movement costs $g(d_o^i|k_i^{OFF})/2$ due to Lemma V.14. In case the distance between OFF and *Follower* increases after handling requests $F_i$ OFF incurs costs proportional to the access costs due to Lemma V.14. Due to Lemma V.3 the competitive ratio is constant, if OFF moved more than *Follower*. Thus assume OFF moved less than *Follower*. The costs per unit moved are larger for OFF than for *Follower* using $g(x|y)/x \geq g(z|y)/z$ for $x \leq z$. Thus, if OFF moved a constant fraction of the distance *Follower* moved its server, OFF incurs also a constant fraction of *Follower*'s total costs, since by Lemma V.1 the total access costs are less than the total migration costs. ∎

## References


[1] D. Arora, M. Bienkowski, A. Feldmann, G. Schaffrath, and S. Schmid. Online strategies for intra and inter provider service migration in virtual networks. In *IPTComm*, 2011.

[2] B. Awerbuch, Y. Bartal, and A. Fiat. Competitive distributed file allocation. In *STOC*, 1993.

[3] N. Bansal, N. Buchbinder, A. Madry, and S. Naor. A polylogarithmic-competitive algorithm for the k-server problem. In *FOCS*, 2011.

[4] N. Bansal, N. Buchbinder, and J. Naor. Metrical task systems and the k-server problem on hsts. In *ICALP*, 2010.

[5] Y. Bartal, M. Charikar, and P. Indyk. On page migration and other relaxed task systems. In *SODA*, 1997.

[6] M. Bienkowski, A. Feldmann, D. Jurca, W. Kellerer, G. Schaffrath, S. Schmid, and J. Widmer. Competitive analysis for service migration in vnets. In *ACM SIGCOMM VISA*, 2010.

[7] M. Bienkowski and S. Schmid. Online function tracking with generalized penalties. In *SWAT*, 2010.

[8] D. Black and D. Sleator. Competitive algorithms for replicaiton and migration problems. In *Technical Report CMU-CS-89-201, Dep. of Computer Science Carnegie Mellon University*, 1989.

[9] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

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

[11] M. Chrobak, L. L. Larmore, N. Reingold, and J. Westbrook. Page migration algorithms using work functions. In *ISAAC*, 1993.

[12] J. Elam and J. Stutz. Some considerations and models for the distribution of a database. In *Rep. CCS 279, Uni. of Texas, Austin, Texas, May*, 1976.

[13] M. Fisher and D. Hochbaum. Database location in computer networks. In *Journal of the ACM, Volume 27, Issue 4*, 1980.

[14] D. Fotakis. On the competitive ratio for online facility location. In *ICALP*.

[15] N. Jain, I. Menache, J. Naor, and F. B. Shepherd. Topology-aware VM migration in bandwidth oversubscribed datacenter networks. In *ICALP*, 2012.

[16] K. Yi and Q. Zhang. Multi-dimensional online tracking. In *SODA*, 2009.